User Manual

English

eDrive Option
Perception

> ***Please note: when you are using Perception Version 8, some of the information in this manual may be outdated. For up-to-date information on the new ePower suite, load the ePower sheet in Perception (Sheets --> Manage sheets…) and then select "ePower Suite".***
> ***A first start of the ePower Suite will give you an introduction in form of a video.***

Document version 2.0 en

October 2021

For Perception 7.40

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

Table of Contents

1 Introduction

The Perception eDrive software option is a dedicated application solution for real time power calculations with simultaneous raw data acquisition on electric drive trains.
It covers the complete test setup from power source/sink to inverter output to machine shaft in a single, easy to operate software environment.

Setting up a measurement is done in a single page, where all information like the measurement method and sensor selection are present. The setup is supported by a graphical representation of the application including wiring diagrams to avoid operator errors.

The acquisition is controlled from a second page giving you both the real time readings of a high end power analyzer and a live waveform display including FFTs like a high end scope.
Beyond the standard 3-phase applications, multi-phase systems like 6- or 12-phase machines can be analyzed. Also, complex setups with multiple motors, multi-level inverters or up to six torque transducers can be analyzed in real time as well, without the need for multiple instruments to be daisy chained.

The eDrive option enhances the Perception formula database with advanced analysis like space vector- or dq0-transformation (aka Park transformation), both doable in real time or in post process.

The eDrive application user interface is highly expandable using existing Perception features. Thus the eDrive option allows powerful real time analysis for a variety of possible electrical or hybrid setups in an easy way.

The power results are stored as continuous traces, can be transferred straight into Excel for mappings or streamed via real time bus or software interface.

2 Preparations and requirements

There are a few requirements in order to use the eDrive application software. These are:

- **Perception Enterprise 64bit software**.
- **Perception eDrive software option.**
- A **GEN2tB**, **GEN3i, GEN3t, GEN7i, GEN7tA or GEN17t mainframe**
  (old mainframes GEN7t, GEN16t, GEN2i or GEN5i cannot be used with the eDrive option).
- At least **one card GN610B** (6 channels, 2 MS/s, 18 bit, +/- 1 kV), **one card GN611B** (6 channels, 200 kS/s, 18 bit, +/- 1 kV) or **one card GN1202B** (12 channels, 100 MS/s, 15 bit, +/- 100 V). This is the minimum requirement, where only a few of the eDrive “applications” can be addressed with this configuration.

  For most “applications”, two or even more **cards** are needed. This includes using analog out torque transducers in combination with 3ph inverters or testing motor – generator configurations. If not enough cards are available, these application modes cannot be used. See later chapter or eDrive datasheet for details.

  With other GEN DAQ input cards, the eDrive software cannot be used.

  Also, older GN61x variants of the 6ch, 1 kV card cannot be used with eDrive software.
  It needs to be newer GN61x**B** types, which can be identified on the front panel having a vertical, black label “FAST” just beside the release lever.

The cards(s) used for eDrive must be inserted in the mainframe starting from slot A upwards, with no gaps or other boards in between.

If you are using old GN61x cards, ask your local HBM representative for an upgrade offer.

If any of the above requirements are not fulfilled, eDrive cannot be used. Then the eDrive software option will refuse to start and displays a message about the failed requirement(s).

If a board GN611B is used in the "application", the maximum sample rate is set to 200 kS/s. Otherwise the maximum sample rate is 2 MS/s.

3 Starting the software

When Perception Enterprise 64bit with the eDrive option is started, there are two possibilities:

1, In Perception, the eDrive option is set to be loaded by default.

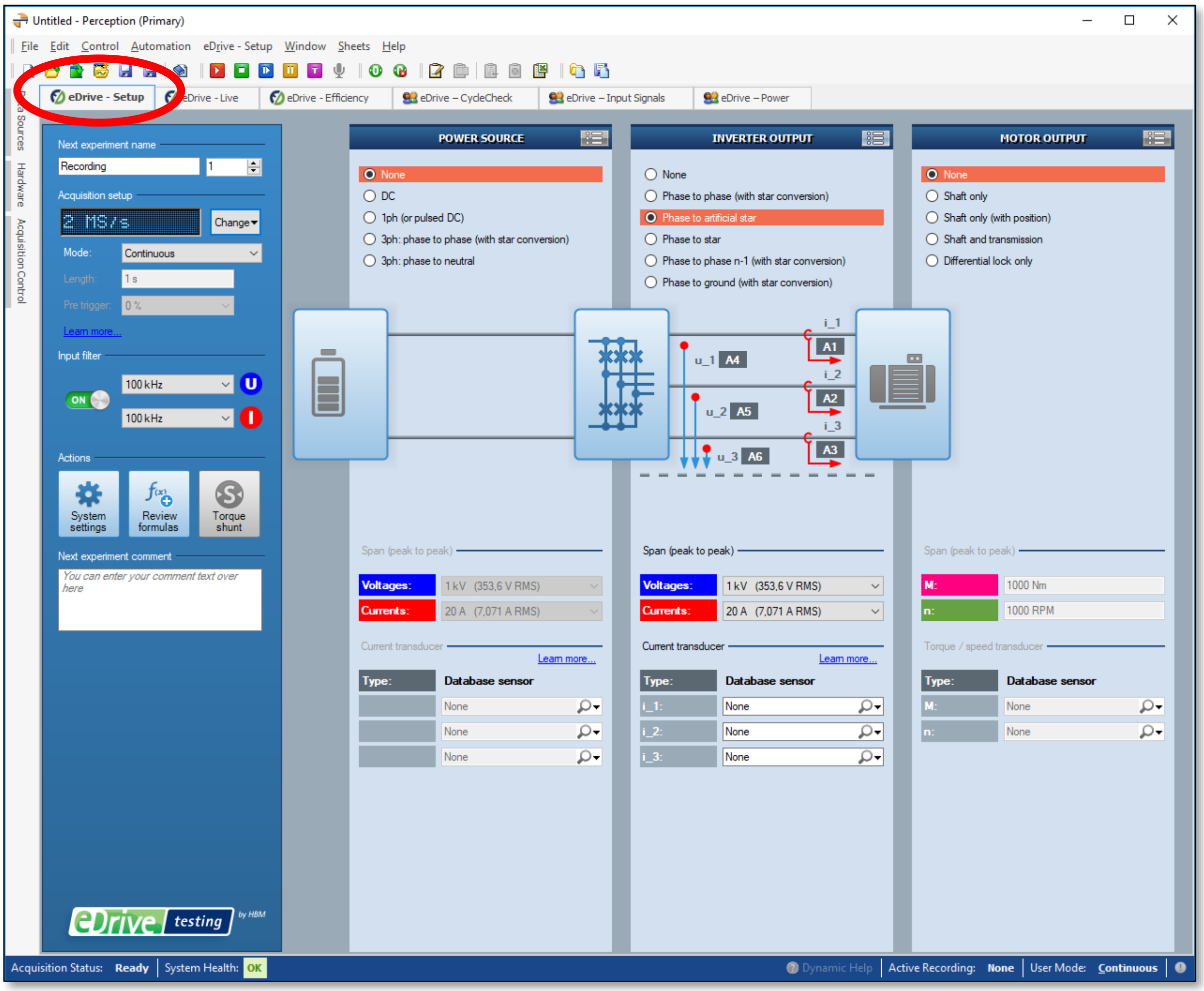
This is the “normal” behavior when a system with the eDrive software option is shipped.

Then Perception will boot up as shown below and show a total of 3 eDrive sheets:

- **eDrive – Setup** (marked in the picture below)
- **eDrive – Live**
- **eDrive – Efficiency**

When a eDrive capable mainframe is connected 3 more sheets are created:

- **eDrive – CycleCheck**
- **eDrive – Input Signals**
- **eDrive – Power**



2, In Perception, eDrive is not set to be loaded as default.

Then Perception will boot without the eDrive sheets.
To load these sheets use the **Sheets – Manage Sheets** menu.
After loading the eDrive option in this menu, the Perception user interface should look similar or close to the one shown above and you are ready to use the software.

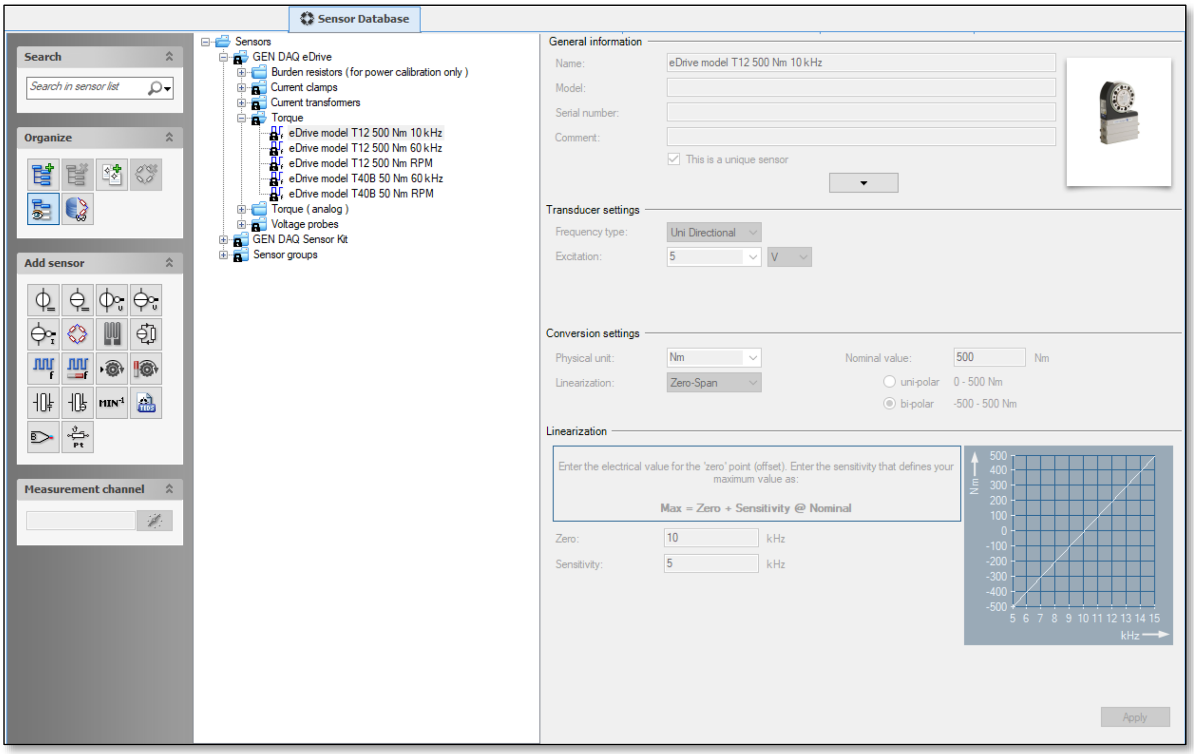
4 eDrive and the Perception sensor database

The eDrive option requires the sensors you are going to use for currents, torque and speed to be present in the sensor database of Perception. Without the proper sensors in the sensor database, the eDrive setup cannot be completed.

Before using the ***eDrive - Setup*** sheet, make sure the sensors used are available in the sensor database. Most HBM sensors and also some common current transducers and current clamps are already available in the sensor database.

If the sensors you want to use are not there, you need to enter these into the sensor database.

See below an example of a T12 torque sensor entered into the sensor database.



If you need assistance entering your sensors into the database, please refer to the **Perception Sensor Database User Manual** which describes the sensor database.
This manual is automatically copied to the PC as part of the Perception installation and can be found under START in **Program files \ HBM \ Perception \ Manuals \ English** or START **HBM \ English**.

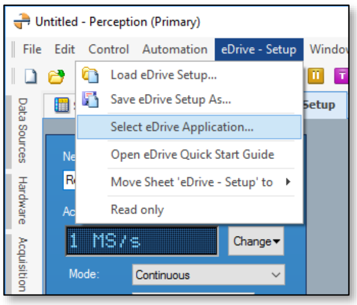
> ***The fastest way to enter a new sensor into the sensor database is to duplicate an existing one and to modify this new sensor to meet the requirements.***

5 Selecting the eDrive Application

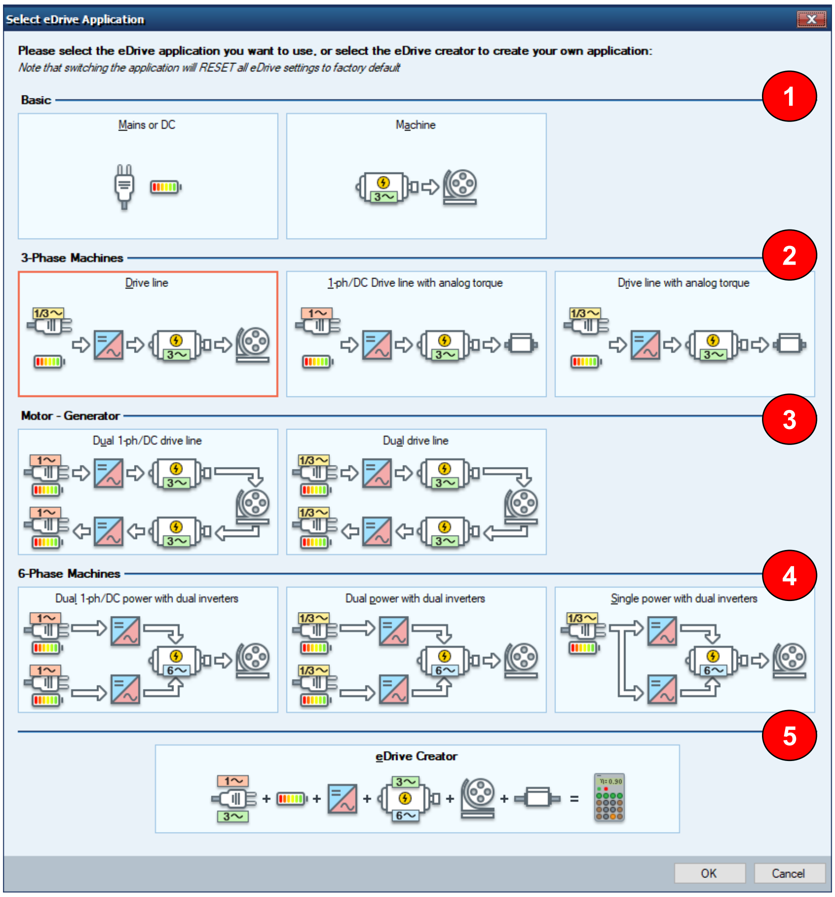
The eDrive option supports several predefined templates to acquire data and has the option to create a custom configuration. All of these templates are graphically represented, the real time formulas to compute the desired results are automatically created and the power values are displayed and stored. Depending on the selected application the main graphics as well as channel allocations and other settings are predefined.

> ***Selecting another application resets the eDrive software to defaults.***
> ***So the application should be selected first before using the eDrive software.***

The application selection is done under **eDrive – Setup**, then **Select eDrive Application….**



Access to the application selection menu



Picture A: Menu to select the desired eDrive application

Item	Description
A-1	Basic eDrive configuration templates. Used for testing a 50/60/400 Hz mains grid, a DC source or an electrical machine only.
A-2	3-Phase Machines eDrive templates. Used for testing a single drive line of power source/sink – inverter – electrical machine.
A-3	Motor – Generator eDrive templates Used for back to back testing of power source – inverter – motor – generator – inverter – power sink.
A-4	6-Phase Machines eDrive templates. Used for testing a single drive line of dual power source/sink – dual inverter – 6-phase electrical machine.
A-5	eDrive creator The eDrive creator allows to do drive efficiency measurement, inverter, motor or drive mappings and system analysis for any configuration, any complexity.

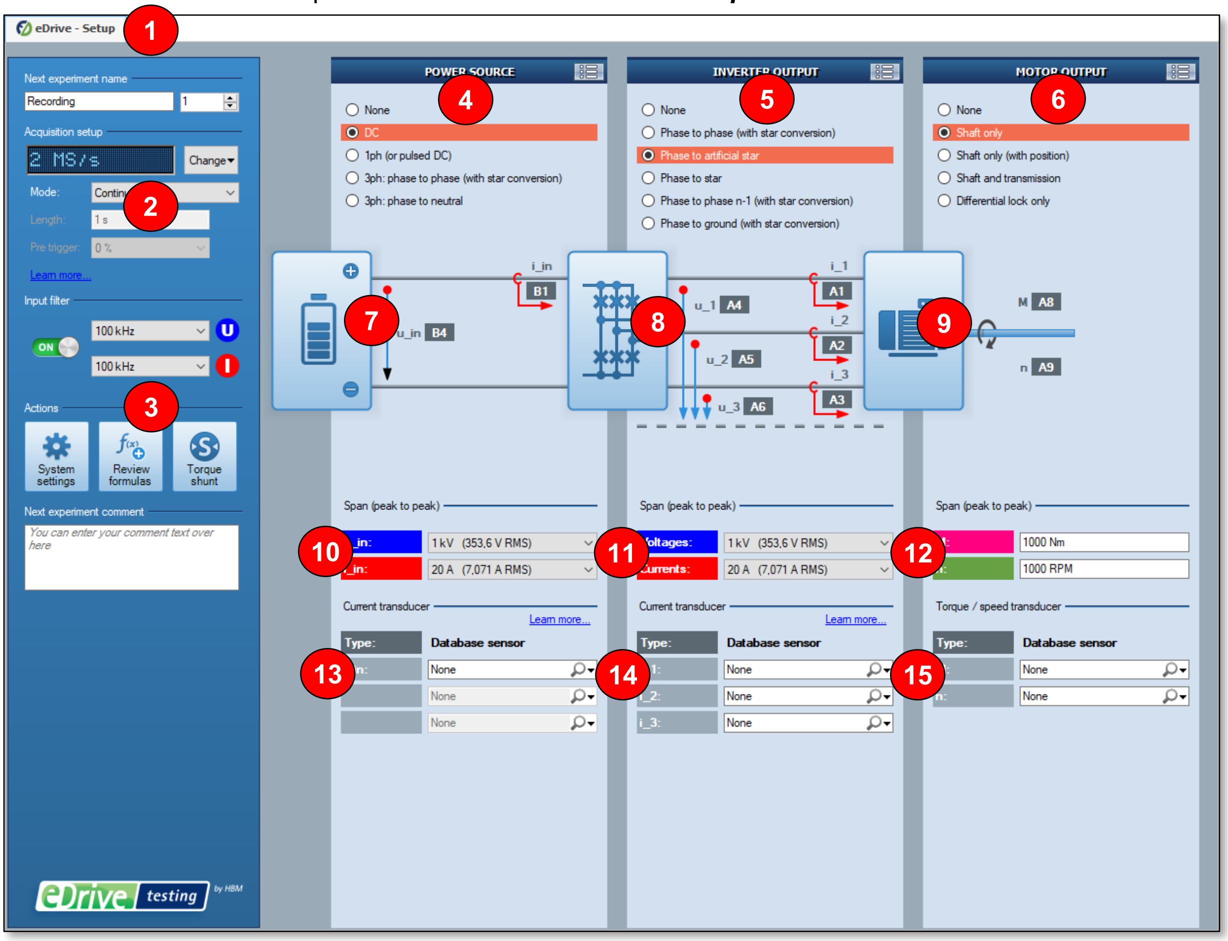
Depending on the number and type of available input boards some advanced applications might not be supported and are greyed out in the list

Extra calculations can be accomplished by manually entering your own formulas and create displays.

6 The eDrive – Setup sheet

6.1 Overview

An overview of the most important sections of the ***eDrive-Setup*** sheet is shown below.



Picture B: ***eDrive – Setup*** sheet

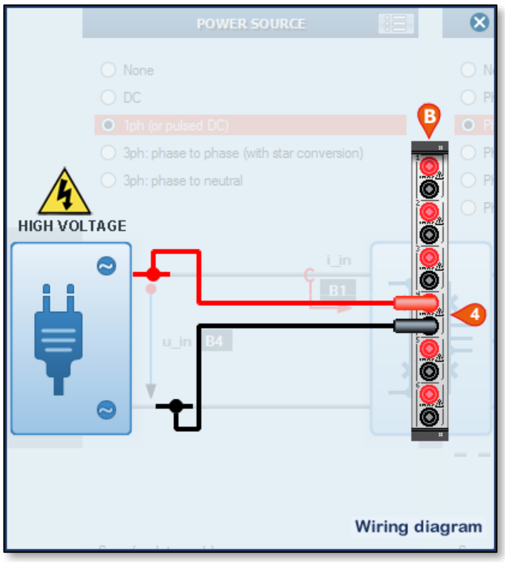
Item	Description	Item	Description	Item	Description
B-1	Selected Perception sheet **eDrive – Setup** is shown	B-4 to B-6	Select configuration for Power Source, Inverter Output and Motor Output	B-12	Select (display) spans for torque and speed
B-2	Main acquisition settings: Filename, Sample rate, Acquisition mode, Filter settings	B-7 to B-9	Area showing the selected configuration, signal names and channels used	B-13 to B-14	Select current sensors from sensor database
B-3	Buttons for other menus or functions	B-10 to B-11	Select spans for voltages and currents	B-15	Select torque and speed sensors from sensor database

6.2 How to use the eDrive – Setup sheet

Before using this sheet, make sure your sensors are available in sensor database.

1. Select the proper configuration for POWER SOURCE (B-4).
   The wiring diagram underneath (B-7) will show the selection, the signal names and channels used.

Click on the channel names (like B4) to show the required input wiring for this signal and channel. This works for all signals / channels in all three areas of the configuration selection.



*Example wiring diagram as shown in the **eDrive – Setup** sheet after clicking on channel label*

2. Select the proper configuration for INVERTER OUTPUT (B-5).
   The wiring diagram underneath (B-8) will show the selection, the signal names and channels used.
3. Select the proper configuration for MOTOR OUTPUT (B-6).
   The wiring diagram underneath (B-9) will show the selection, the signal names and channels used.
4. Select the current transducers (B-13) used for POWER SOURCE.

The fastest selection of sensors from the sensor database is by typing just a few letters of the sensor name into the entry field. Then a matching list will appear, from which the sensor can be selected. For example, if you have a clamp called "AYA M1V AC", just type "Ay" in the current sensor field, then select from the list that appears.

5. Select Span (B-10) for POWER SOURCE voltage and current measurement.

> ☞ ***The spans selected here are peak to peak, so bipolar.***
> ***So selecting a span of "1000 V" means "+/- 500 V".***
> ***Same is true for currents, torque and speed.***
>
> ☞ ***For easier usage of the spans, the RMS value is placed in brackets behind the peak to peak value.***
>
> ☞ ***Select the span for currents AFTER selecting a current sensor,***
> ***as the displayed selection list of spans depends on the sensor.***

6. Select the current transducers (B-14) used for INVERTER OUTPUT.
7. Select Span (B-11) for INVERTER OUTPUT voltage and current measurement.
8. Select the torque sensor (B-15) used for MOTOR OUTPUT.

> ☞ ***Note that speed sensors of "Frequency type: Uni Directional" are recognized in the eDrive software as torque sensor and shown in the selection field for "M".***

9. Select the speed sensor (B-15) used for MOTOR OUTPUT.

> ☞ ***Note that speed sensors of "Frequency type: Uni Directional, Bi Directional and Quadrature" are recognized in the eDrive software as speed sensor and shown in the selection field for "n".***

10. Select Spans (B-12) for MOTOR OUTPUT torque and speed measurement.

6.3 Other features and functions in the eDrive – Setup sheet

The area (B-2) in the upper left portion of the sheet is used to enter a name and “run number” for the next experiment to be recorded, to select the desired acquisition mode, the sample rate and the filter settings.

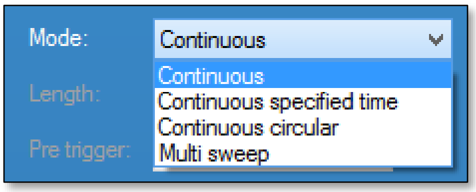
By default, the experiment file will be stored in the “c:\eDrive Recordings” folder of the PC. This storage location can be changed by the user, if desired.

The sample rate for all GEN DAQ input channels of an eDrive system is forced to be same, thus preventing phase shift between channels.

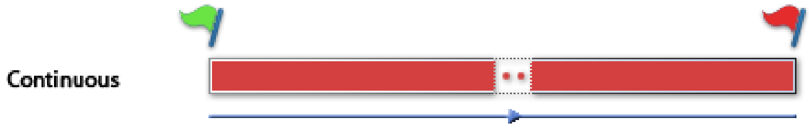
The filter settings for voltage and current channels should be the same if possible, to prevent phase shift. In some rare cases, in very noisy environments, it might be necessary to set the filters different. Be aware that this introduces a phase shift influencing the computed results.

6.4 Acquisition setup – MODE

This field is used to select one of three basic modes.

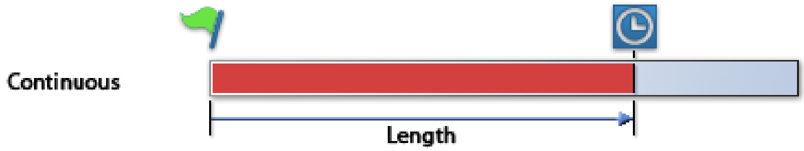


1 – **Continuous** is an acquisition mode that you Start and Stop with individual commands. These are either manually controlled or via remote control pins at the DIG I/O connector of the mainframe. Also Start and Stop can be controlled via software remote control calls (RPC).



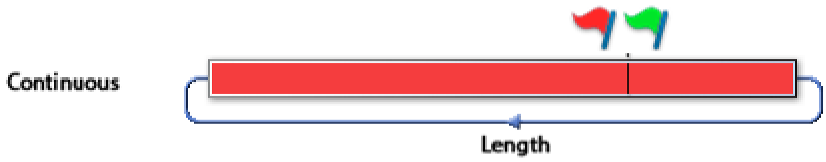
This mode is typically used for single, longer tests of a component, like a run up test or step response test. The result of a Continuous acquisition is a single experiment file containing an acquisition of typically a few seconds or minutes. Each Continuous acquisition will have a different length in time.

2 - **Continuous specified time** is very similar to the above, except that the acquisition now stops automatically after a preset time. So only the Start needs to be initiated. The shortest acquisition time is 1 s; the longest is only limited by free storage space.



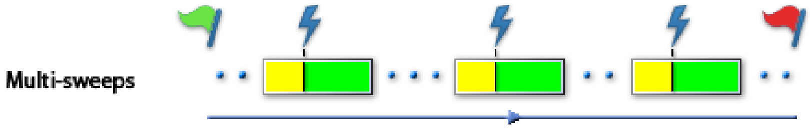
This mode is typically used for a small series of sequential tests.
Every individual acquisition is started only, and stopped automatically. This reduces the necessary user interaction. The result of multiple Continuous specified time acquisitions then is a series of experiments, each containing a single acquisition of defined length. This series of experiments can be processed later using Automated batch processing.

3 - **Continuous circular** is a continuous recording with a predefined maximum length of the recording. When the maximum recording time is reached, the new data is added to the recording and the oldest data gets discarded so that only the last specified length of data gets stored



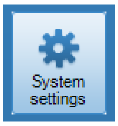
This mode it typically used for long running tests in which only the last recorded data is wanted. Every acquisition needs to be started and stopped manually. The result of a Continuous circular acquisition is a single experiment file containing an acquisition of typically the last few seconds or minutes.

3 - **Multi Sweeps** is a series of very short acquisitions. The acquisition is started, and then each trigger initiates the recording of a sweep of predefined length.
A trigger can be initiated using the keyboard, the Trigger Input at the DIG I/O of the mainframe or via software remote procedure call (RPC).



This mode is typically used for motor mapping, where hundreds or thousands of acquisitions – one per desired set point to be analyzed - should be made.
The result of a Multi-Sweep acquisition is a single experiment containing all the triggered sweeps (so a single file). The resulting file can be processed automatically during or immediately after acquisition using the Automated Recordings dialog, or later by using the Display Processing dialog.
The maximum sweep length in eDrive is limited to 1 s.

6.5 Other controls in the SETUP menu



The **System settings** button in area (B-3) is used to enter a submenu with some special features for the eDrive application.
See later chapter for details.

The **Review formulas** button in area (B-3) will create formulas to re-analyze acquired data using the post process formulas of Perception. User entered formulas are NOT recreated.
See later chapter for details.

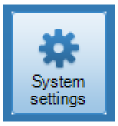
The **Torque shunt** button in area (B-3) is used to enter the submenu for torque shunt calibration and zeroing.
See later chapter for details.

Zeroing can be done with any torque transducer giving a frequency output signal proportional to torque. This is true for most modern, digital torque transducers, and especially for HBM torque transducers like the T12, T40B or T12HP.

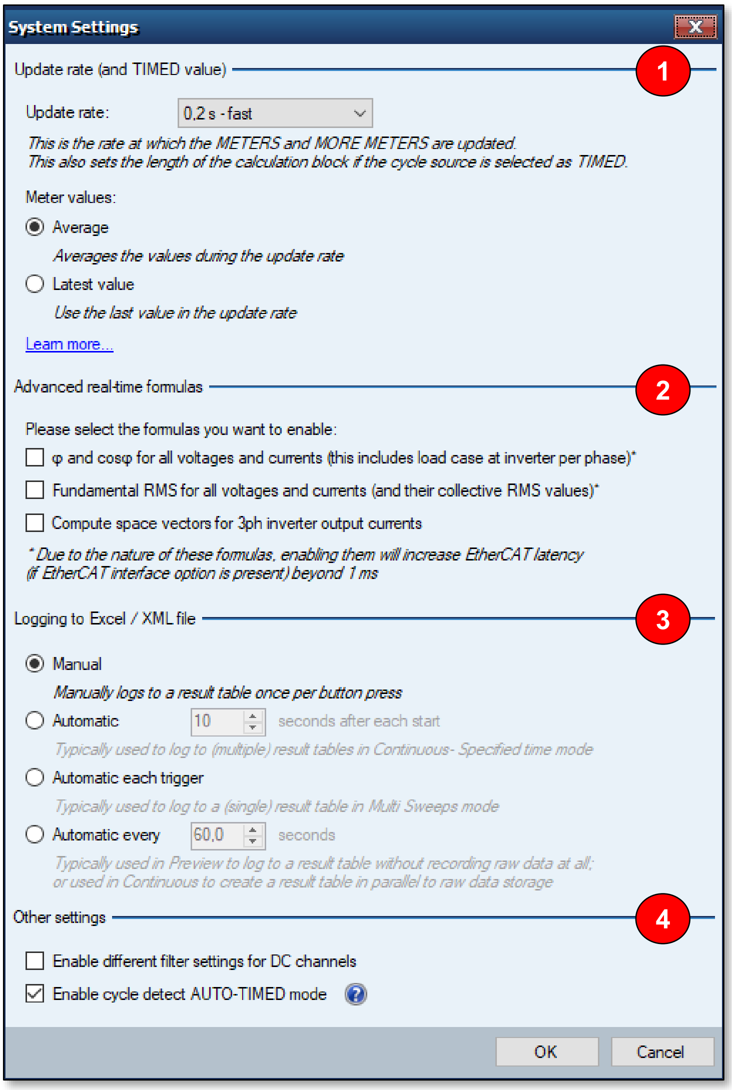
Shunt calibration can be done with all torque transducers with built-in shunt resistor, activated by a 5V signal, and properly connected. This is true for HBM torque transducers like the T12, T40B or T12HP.

6.6 The SYSTEM SETTINGS menu

For advanced system settings, this first level entry menu is used.



When pressed, the menu below appears:



Picture C: **SYSTEM SETTINGS** menu

Item	Description
C-1	Update rate (and TIMED block length) selection
C-2	Selection of advanced formulas to be executed
C-3	Setup for “Log to EXCEL / XML File”
C-4	Other settings

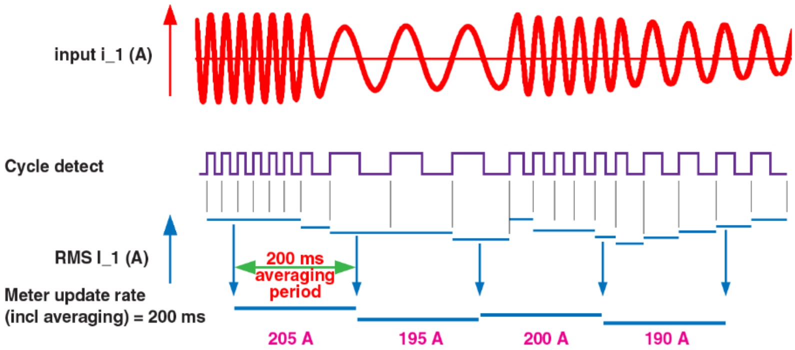
6.6.1 Update rate settings

The update rate can be selected as 200 ms – 500 ms – 1 s.
The update rate setting has different effects on the results, depending on how calculations are performed:

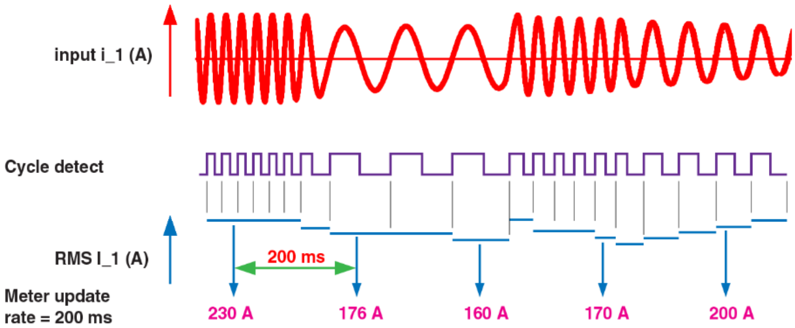
6.6.1.1 Calculations are done cycle based

Then the following applies, depending on the sub setting:

* **Average:** Cycle results are averaged over the selected “Update rate” for the METERS, MORE METERS, the log to EXCEL and eDrive efficiency mapping.

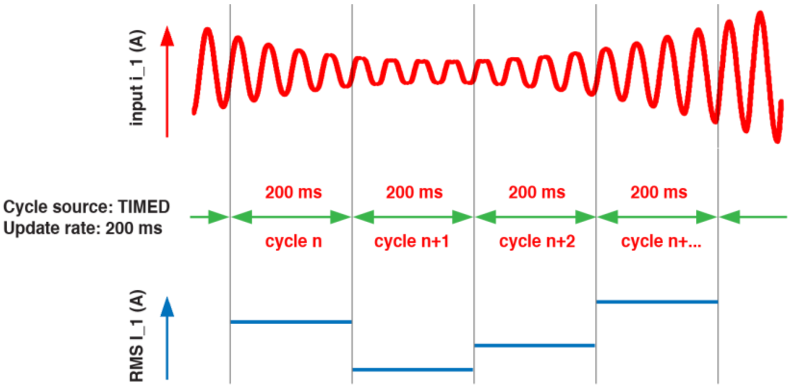


* **Latest value**: Cycle results are taken out of the (async) data stream at the selected “Update rate”. See graphics below for explanation.



Explanation: From the input signals the RMS and other power values are calculated per cycle. This is an asynchronous data stream with the (changing) frequency of the fundamental.
Out of this asynchronous data stream values are taken with the selected update rate and shown in the meters. Thus, each meter reading is a single cycle result without any averaging (unless the if calculation is set over multiple cycles).

6.6.1.2 aCalculations are done TIMED



Explanation: If the cycle source is set to TIMED in the eDrive – SETUP sheet, the time entered in the Update rate dialog is the interval for the TIMED cycle. All RMS and power results are then computed over the whole interval ignoring any signal cycles.
A longer interval leads to a more accurate result due to averaging and a more stable readout, but it also reduces dynamic behavior.

6.6.2 Advanced real-time formulas

These checkmarks are used to select if advanced results like φ, cosφ, fundamental RMS and space vectors should be computed or not.

Enabling these increases the load on the DSP's, while the computation still should be possible. The main impact of enabling these advanced calculations is on EtherCAT latency. Depending on the selections, the guaranteed latency of 1 ms may no longer be achievable.

> ***For users without EtherCAT, enabling these advanced computations has no negative effect.***

6.6.3 Logging to EXCEL / XML file

In this area the user specifies how the XML table should be created:

- **Manual**
  A single entry into the XML table is made on each press of the button "Manual Logging" in the eDrive – LIVE sheet. This is useful in an interactive way, for example when no raw data is stored at all.

- **Automatic "n" seconds after each start**
  A single entry into the XML table is made automatically at a defined time after starting an acquisition. Typically used in combination with mode: Continuous specified time.

- **Automatic each trigger**
  A single entry into the XML table is made automatically at each trigger of a multi sweep recording. Typically this setting is used for motor mapping in mode: Multi Sweeps.

- **Automatic every "n" seconds**
  Multiple entries into the XML table are made automatically at a predefined time grid, set in

seconds. Useful for long term recordings in mode: Continuous.

6.6.4 Other settings

6.6.4.1 Enable different filter settings for DC channels

If this is selected the eDrive – SETUP sheet layout will slightly change for the filter section and allow to select different filter frequencies for all DC channels from the input block (if this is in DC).

6.6.4.2 Enable cycle detect AUTO-TIMED mode

If auto-timed mode is enabled, values are calculated at least once a second.

This setting enables a timeout timer on the cycle detectors which guards if cycles are produced. If one (1) second passes since the last cycle is produced, a new cycle is generated and calculations are done over the past second, even if no cycles are detected in the signal.

6.7 Review Formulas

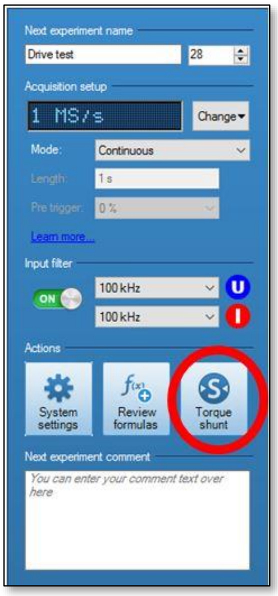
This button is used to create formulas for review, using the formula database of Perception.
See later chapter for an example analysis using these formulas.

User entered formulas are NOT recreated.

Formulas in the post-process formula database (FDB) use a different syntax than formulas in the real-time formula database (RT-FDB) which is used for live calculations.

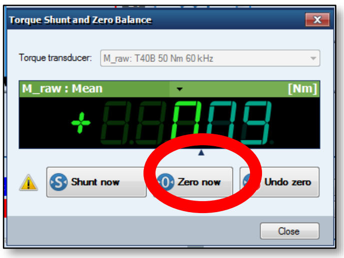
6.8 Zeroing and shunt calibration using a HBM T12(HP) or T40 torque transducer

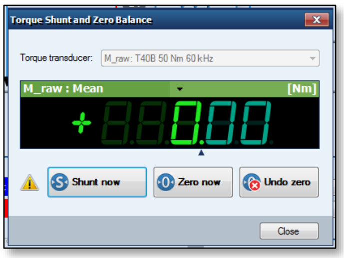
In order to check proper setup and cabling of the torque transducer as well as to zero the torque transducer, the Torque shunt menu is used.



Entering the torque shunt menu

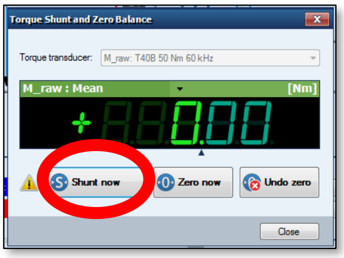
The system will go into Preview mode and a display will appear showing the actual torque reading. If there is an offset, use the button ***Zero now*** to zero the torque transducer.

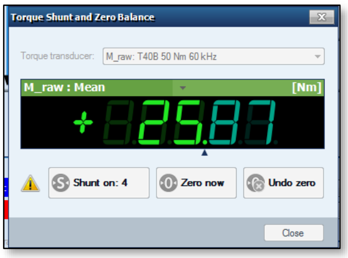




Torque reading before and after zeroing

*Then use **Shunt now** to enable shunting. The reading should be about half of the nominal value of your HBM torque transducer (example below shows 50 Nm transducer, so reading is ~25 Nm).*



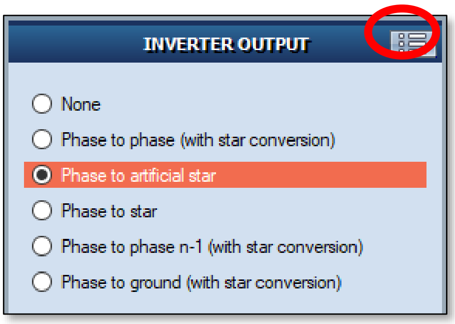


Torque reading when enabling the build-in shunt

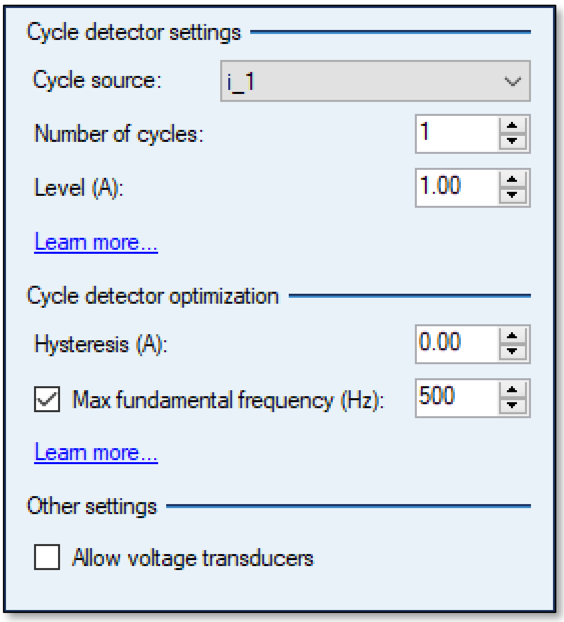
> ***Be careful using the "Shunt cal" function in a test bench. If the automation system controls the torque, using the shunt resistor might cause problems, as the automation system cannot compensate the "reading" now delivered by the torque transducer (as caused by a shunt).***

6.9 Cycle detector settings menu

This menu is located in the upper right corner of each block.



After clicking on the cycle detector settings menu button the cycle detector menu opens, as shown below:

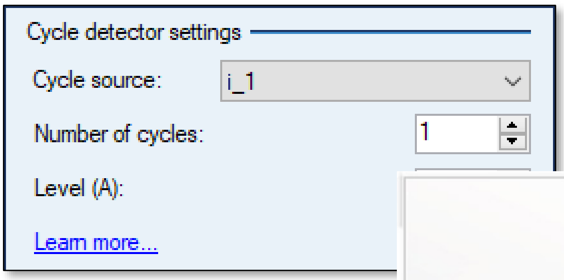


In this menu all cycle detection settings are made.

> ***On the eDrive - Live sheet the cycle detector settings menu is also present, but with a limited set of options.***

6.9.1 *Cycle detector settings*

These are the main settings for the cycle detection.



Item	Description
Cycle source	Channel used fo [illegible] i_1 or u_1; or can be set to [illegible] s or to get an approximate reading; or can be set to [illegible] n the reference pulse on channel Ev A7_01 (refere[illegible] eed signal) determines the cycle; in other words, the[illegible] hanical revolution
Number of cycles	Number of cycle[illegible] lculations are performed
Level	Cycle detection l[illegible] i_1 or u_1 are selected; typically 0 in all AC configurations

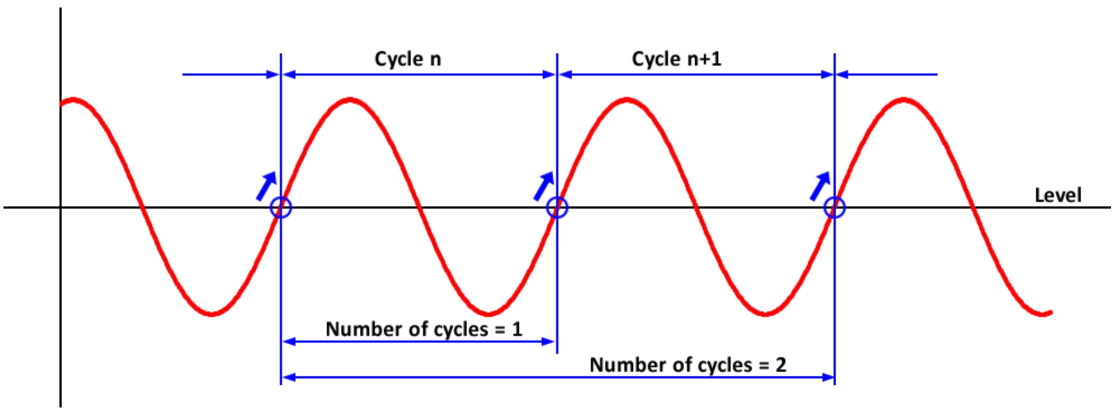
A correct power calculation requires that the math is performed over half-cycles or a multiple of half-cycles.

The **Cycle source selection** sets the channel that is used for cycle detection.

The **Number of cycles** selection sets the number of full cycles that are used for the live power calculations. Selecting more cycles improves accuracy (by averaging) while selecting fewer cycles improves response time, but might give a less stable reading.

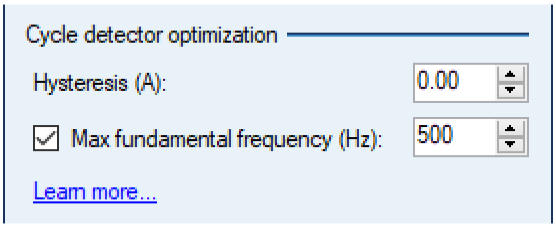
A cycle is defined as the time between two identical level crossings with respect to direction (up or down) and level.
The **Level** setting defines the level for crossing and is “0” by default.
The direction is set automatically and positive (up) by default.

The following diagram is an example of cycle detection.



6.9.2 *Cycle detector optimization*

These additional settings are used to make the cycle detection work in difficult or noisy environments.

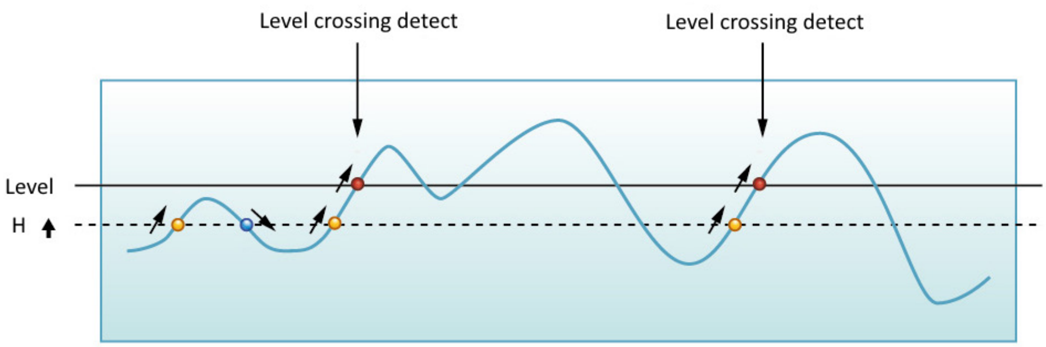


Item	Description
Hysteresis	Hysteresis for noise suppression in the channel used for cycle detection; in order to work properly, the hysteresis should be set to a value larger than the noise on the signal
Max. fundamental frequency	The setting here both applies a filter to the cycle source signal to reduce noise (for cycle detection only, not for calculation) and also introduces a cycle detect suppress time to prevent the cycle detection from misfiring before a cycle is completed

The signal used for cycle detection is typically not a smooth sinewave but a signal with noise and distortion.
To make cycle detection robust and less susceptible to noise and distortions there are two options:
- Introduce a **hysteresis** for noise suppression
- Enter a **maximum fundamental frequency** to set a cycle signal filter and cycle detector holdoff.

Hysteresis explained

Hysteresis is a digital noise suppression technique.
A level crossing is detected if the signal crosses the chosen hysteresis level first, and then crosses the chosen cycle detect level.
After this first level crossing detection the signal must cross again the chosen hysteresis level, and then the cycle detect level for the next level crossing detection, and so on.
For practical use this means that the hysteresis area needs to be larger than the noise, but smaller than the signal itself. This is depicted in the following diagram.



Maximum fundamental frequency explained

Enabling and entering the maximum fundamental frequency expected defines both a cycle signal filter as well as a cycle detector holdoff to improve correct cycle detection.

The maximum fundamental frequency defines the filter frequency that will be used to remove noise from the cycle source signal: a Bessel low-pass filter with a -3 dB cutoff at 2x the selected fundamental frequency.

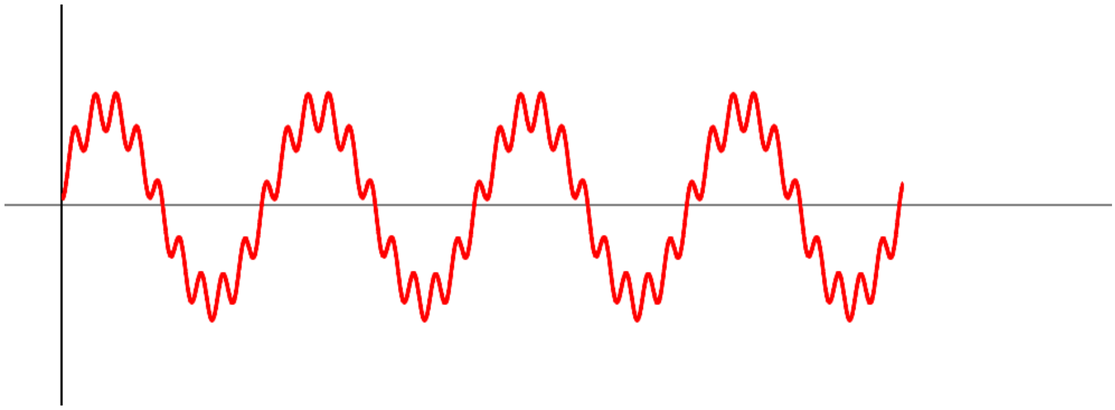
Example: Maximum fundamental frequency = 100 Hz -> Filter cutoff frequency = 200 Hz.

Note: This filter is only used for the cycle detection, not for computations.

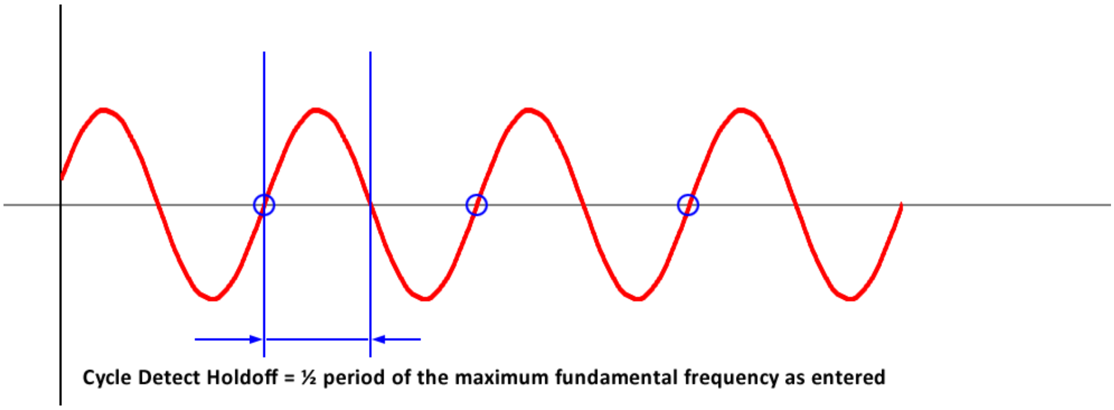
The maximum fundamental frequency also defines the minimum cycle time. Anything below that time is rejected by a hold-off period. By definition the hold-off period is set to half the cycle time of the maximum fundamental frequency.

Example: Maximum fundamental frequency = 100 Hz -> Cycle time T = 1/100 = 10 ms.
As a result a crossing within 5 ms (= 10 ms/2) after the first crossing will be disregarded.

Signal for cycle detection with noise:

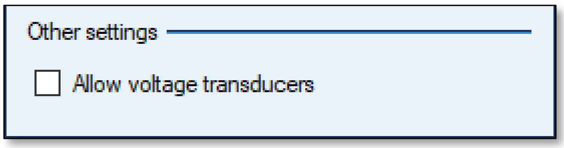


Filtered signal with additional cycle detect holdoff:



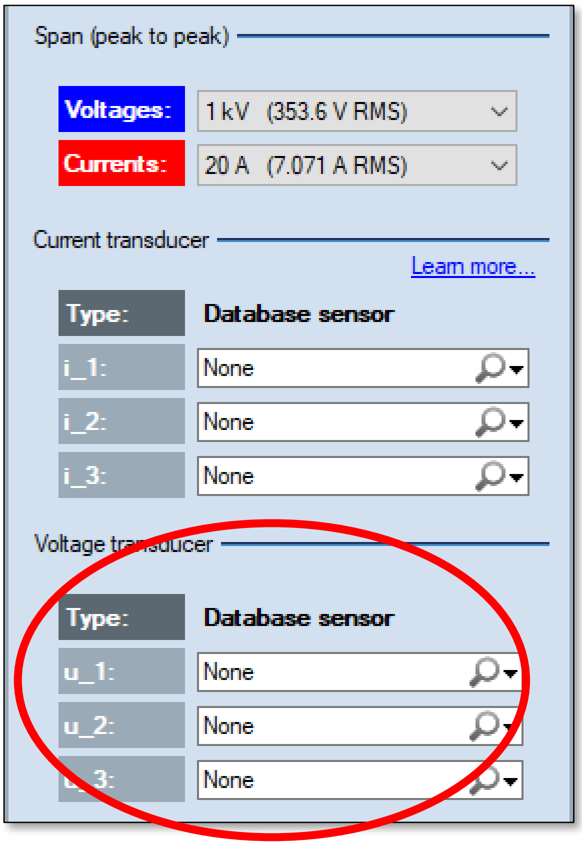
6.9.3 Other settings

This section is used to enable the usage of external voltage dividers.



Item	Description
Allow voltage transducers	Enables usage of voltage transducers for the voltage inputs; if checked, transducers can be applied (in the ***eDrive – Setup*** sheet) from the sensor database to each input channel

If enabled, another input section in the **eDrive – Setup** sheet becomes visible, where the voltage transducers can be selected:



In this user manual ***only the cycle detector settings for the INVERTER OUTPUT are explained.***
Note that there are similar menus available for POWER SOURCE and MOTOR OUPUT. Only difference there is that the Cycle Source for these can also be selected as LINKED TO INVERTER. Then the calculation interval for these blocks is synchronized with the INVERTER OUTPUT calculation interval.

7 The eDrive – Live sheet

7.1 Overview

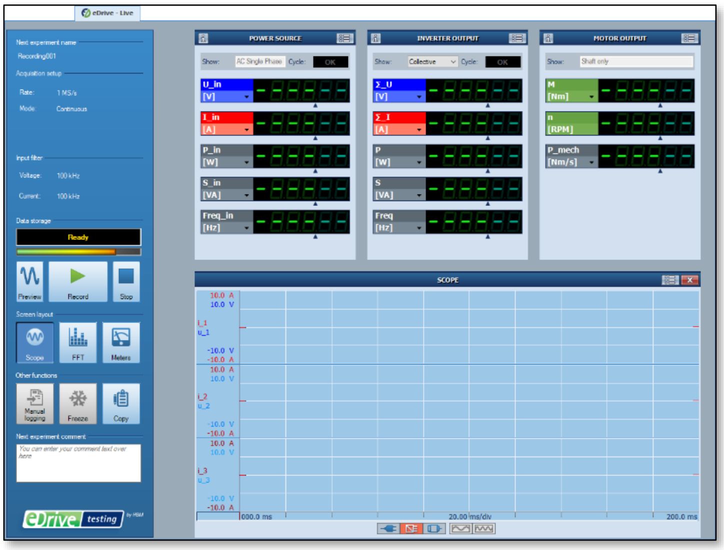
An overview of the most important sections of the ***eDrive - Live*** sheet is shown below:

Picture D: ***eDrive – Live*** sheet

Item	Description	Item	Description	Item	Description
D-1	Selected Perception sheet **eDrive – Live** is shown	D-4	Buttons for other menus or functions	D-7	Main meter area with Inverter Output results
D-2	Acquisition control area for Preview, Record and Stop	D-5	Cycle settings menu (for Inverter Output)	D-8	Main meter area with Motor Output results
D-3	ON / OFF buttons for display options Scope, FFT and Meters	D-6	Main meter area with Power Source results	D-9	Scope display, ON by default

7.2 How to use the eDrive – Live sheet

Select the ***eDrive – Live*** sheet. The default appearance is shown below. The system is still IDLE not showing any data.



The y-axis labels for the SCOPE still show default values of +/- 10 A and +/- 10 V. This is ok, as no data is displayed yet. As soon as you start a recording or go to Preview mode, the y-axis labels will show proper values.

Press ***Preview*** to get a LIVE display – still without storing data.

Above picture shows a default display of the ***eDrive – Live*** sheet with selections:
* Power Source = AC 1ph (or pulsed DC)
* Inverter Output = Phase to artificial star
* Motor Output = Shaft only

Typical signals from the inverter output are shown.

7.3 Main Meter settings

An overview of the most important sections of the METERS is shown below:

Picture E: ***eDrive – Live: Meters*** area

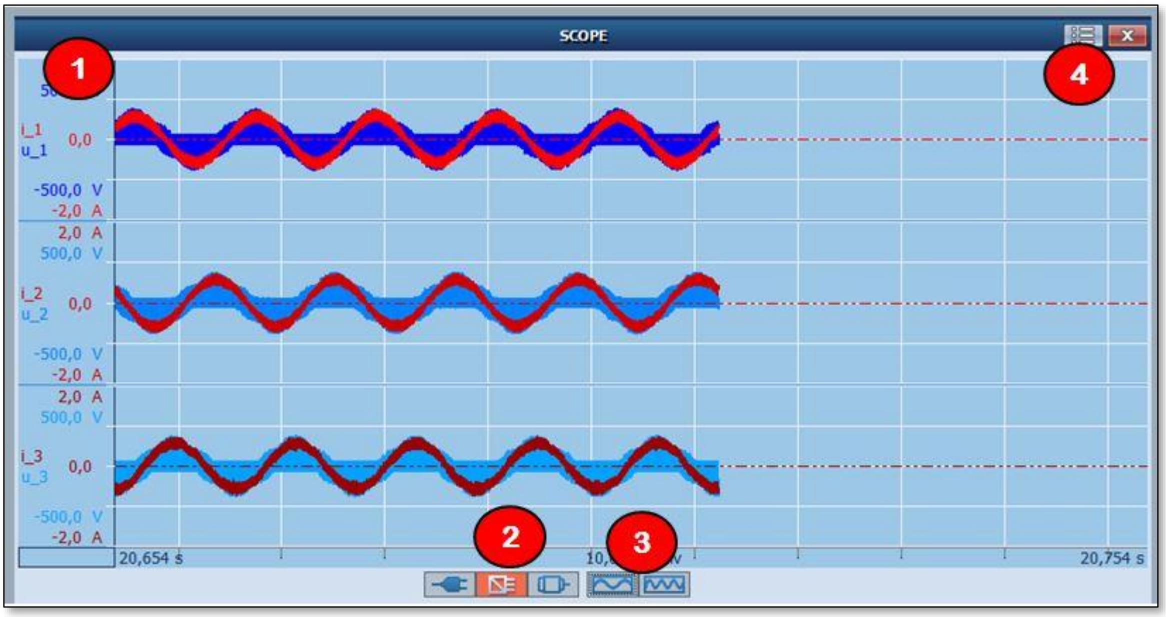
Item	Description	Item	Description	Item	Description
E-1	Meter block for POWER SOURCE results	E-5	Cycle settings menu	E-9	Decimal point position control: drag left/right to change decimal position
E-2	Meter block for INVERTER OUTPUT results	E-6	Selection of collective or phase values to be shown	E-10	"Energy flow" indicators[1]: ➲ Motor mode ⮌ Generator mode
E-3	Meter block for MOTOR OUTPUT results	E-7	Cycle detect status: **OK** – Cycle detect ok **NOK** – Cycle detect not ok **TIMED** – Time slice used **Ref_OK** – Cycle detect on reference pulse, and ok **Ref_NOK** - – Cycle detect on reference pulse, and not ok	E-11	Load indicator per phase[1]: L -> inductive load C -> capacitive load
E-4	Block setup information (will appear in tooltip)	E-8	Pulldown for unit prefix selection (none / k / M)		

[1] *Note: The "Energy flow" and "Load" indicators might not be shown due to other system settings.*

> ☞ ***In "Phase to phase" measurement, a phase to star conversion is done first, and then RMS and other power calculations are done. This means the "Phase to star" RMS voltages are displayed in the meters, even if the system is in "Phase to phase" measurement mode.***
> ***The "Phase to phase" RMS values can be displayed in the MORE METERS section, as explained later.***

7.4 Scope settings

An overview of the most important sections of the SCOPE is shown below:



Picture F: ***eDrive – Live: Scope*** area

Item	Description	Item	Description
F-1	Main scope display	F-3	x-axis zoom controls
F-2	Selection to show POWER SOURCE, INVERTER OUTPUT or MOTOR OUTPUT waveforms in the scope display	F-4	Display options menu: select display mode, grid on/off and more…

> ***The scope display shows the same traces as selected for the main meters block (E-4).***
> ***If the selection there is "Collective" for INVERTER OUPUT, the scope will show all traces.***
> ***If the selection is "Phase 1" (or 2, 3), the scope will also show phase 1 (or 2, 3) only.***

7.6 Cycle detect status display

The cycle detector settings are the most critical setup parameters of the system in order to deliver proper results.

If the cycle detection works, all results are correct. If the cycle detection fails, all or at least some results might be wrong.

In the Meters area (E-2), there is an indication of the cycle detect status (E-5).
This cycle status indication might have different values:

Status **OK** tells that the cycle detection did find cycles (and thus the power calculations are executed).

An OK in the cycle detector status tells there are cycles found, but not how many.
So there is still the possibility that there are TOO MANY cycles found due to too much noise. This also would show OK. In this case the fundamental frequency reading would not be stable but bouncing around significantly.
Then the cycle detector settings need to be adjusted to eliminate the noise.

If the "cycle detect AUTO-TIMED mode" is enabled, there will always be an OK. See: Enable cycle detect AUTO-TIMED mode for more info.

Status NOK tells that there were no cycles found in the selected cycle source channel with the selected parameters. In this case changing the cycle detector settings is required.

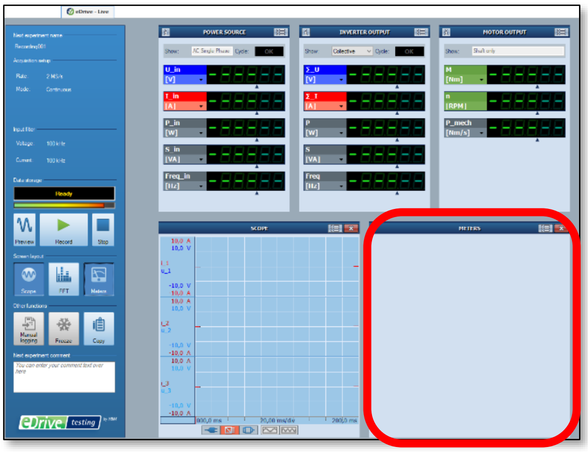
Status TIMED tells that the cycle detection is disabled and a time slice is used to do the power calculations. The length of the slice is set in the **System settings** menu / **Update rate**.

There are two other status messages potentially shown.
Ref_OK and Ref_NOK are used in case the reference pulse is used as cycle source. See below.

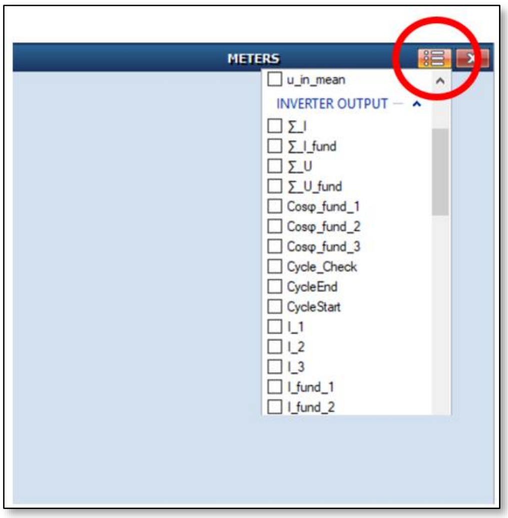
In order to change the cycle detector settings the cycle settings menu (E-9) is used.

7.7 Configuring More Meters

In the ***eDrive – Live*** sheet, click on the METERS button to enable the METERS display.



Open the context menu of the METERS display to get a list of all available parameters calculated.



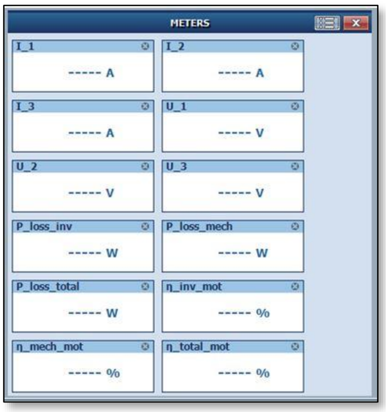
This list contains the following parameters:

- All asynchronous results from the RT-FDB
  - This includes both results from the formulas created automatically by the eDrive GUI as well as the results from user defined formulas
- All temperature channels present in the system
  - These are MEAN values
  - These values might come from a MX1609B, a MX809B, or a GN840B/1640B with some channels being in thermocouple or RTD mode
- All CAN bus messages recorded with an MX471B
  - These are MEAN values

Due to the asynchronuous nature of data coming from the QuantumX modules, the QuantumX based values shown in the METERS might have a sligthly different time stamps (up to ~200ms) compared to the values coming from the GEN DAQ input boards. This effects only the live meters, in review all data is fully synchronized.

Pending from the selection in the System Settings menu (see section 6.6.1) the METERS may contain averaged or the latest values.

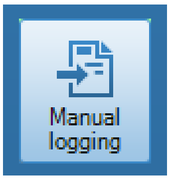
Select those you want to see, and close the context menu again. Now the METERS display should look like below. As the system is still in IDLE, there are no values in the meters.



If the system is in PREVIEW or in RECORD, the meters will display the requested values as shown below.

Note that the main and the more METERS on the LIVE screen are these values which are logged to excel. So make sure all the values you want to have in your logging file are present and displayed in the LIVE display.

7.8 Other features and functions in the eDrive – Live sheet



The **Manual logging** button in area (D-8) will store all meter values currently shown on the screen into an XML file.
When pressed, all meter values shown on the screen are stored (WYSIWYG – What You See Is What You Get).

The XML file name depends on the recording state the system is in when logging:

RECORD -> ***(recording name).XML***.

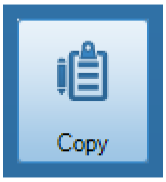
PAUSE or IDLE -> ***eDrive logfile (Perception start time in UTC format).XML***.

The logging files are stored in (drive:/)(Active Recording Folder)/Logfiles.

The default path is ***c:/eDrive Recordings/Logfiles*** unless changed by the user.

> ***If EXCEL is installed on the system, and logging is done in PAUSE or IDLE mode, the XML file will be opened automatically and the results are shown in EXCEL. With each press of MANUAL LOGGING the EXCEL table will grow by one row.***
> ***In RECORD mode the XML file will not be opened automatically, but only grow in the background.***
>
> ***Beyond the manual logging, it is also possible to do an automated logging, based on time or triggers. To do so use the SYSTEM SETTINGS menu.***

The **Freeze** button in area (D-8) freezes the display (only) for closer examination.
If there is a recording running in the background, this will continue.



The **Copy** button in area (D-8) is used to copy the current screen to the clipboard of Windows. From there it can be inserted into other applications like Word.

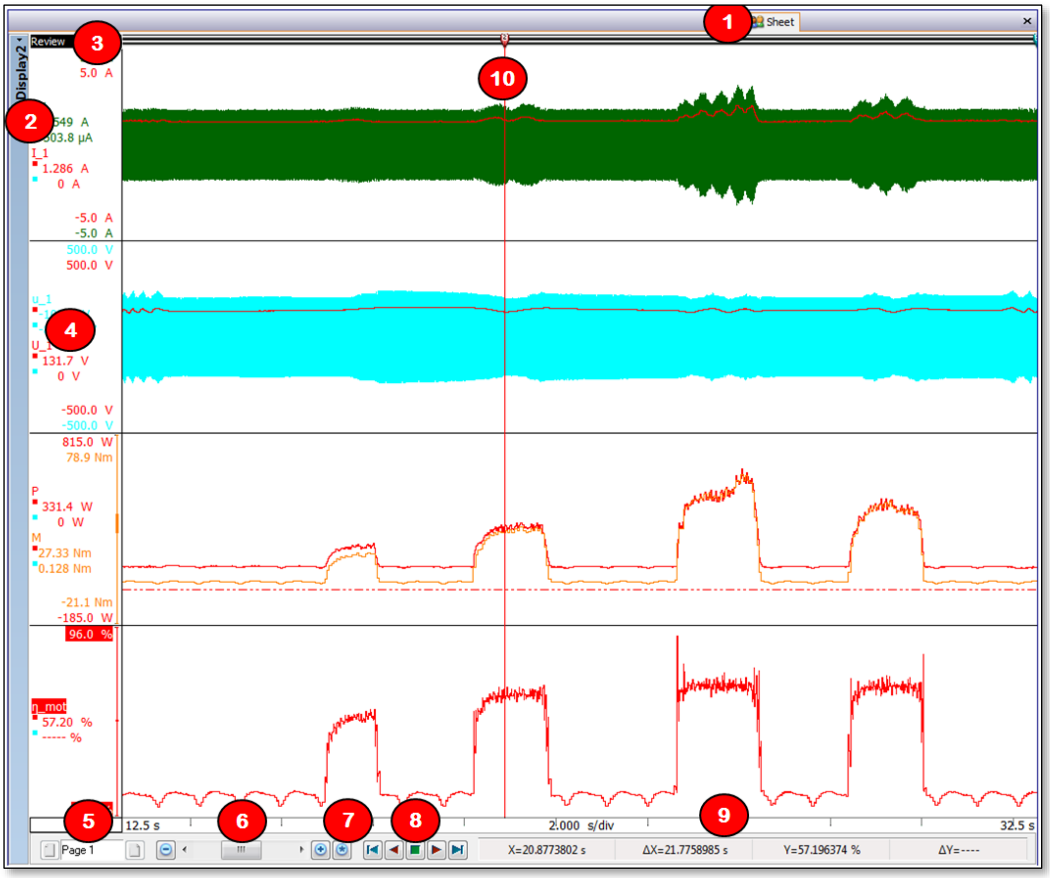
8 The eDrive – Display sheets

8.1 Overview

To show all relevant raw data and calculated results, eDrive automatically creates a total of three sheets, each with multiple pages:

- **eDrive – Input Signals** shows all the raw data
- **eDrive – Power** shows power results and efficiencies
- **eDrive – CycleCheck** to verify if the cycle detection is working properly

An overview of the most important sections of each display is shown below:



Picture G: ***eDrive – Live: Scope*** area

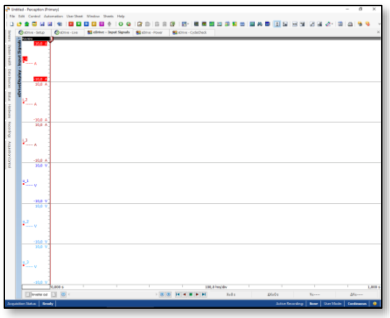
Item	Description	Item	Description	Item	Description
G-1	Sheet name, example "Sheet"	G-5	Page selector, example "Page 1"	G-9	x-axis scaling and cursor readings for "active" trace (reverse highlighted)
G-2	Display name, example "Display2"	G-6	x-axis scrollbar and zoom controls	G-10	Active (red) cursor and cursor handle
G-3	Display mode: REVIEW or LIVE	G-7	Auto scale x-axis button		
G-4	Channel names, cursor readings and y-axis labels	G-8	Scroll controls forward / backward		

8.2 General behavior of the displays in these sheets

The sheets and the pages in these sheets are created automatically. Their content is also changed dynamically when the measurement configuration is changed by the user. The sheets are set to READ ONLY, so the user cannot change these sheets unless they are unlocked manually.

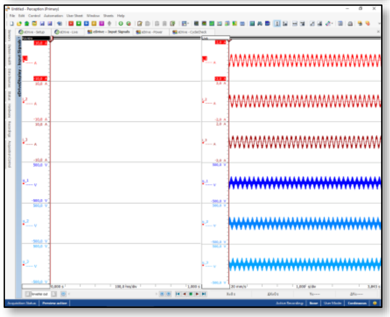
The appearance of the sheets depends on the acquisition state the eDrive system is in.

Acquisition states and resulting display content:

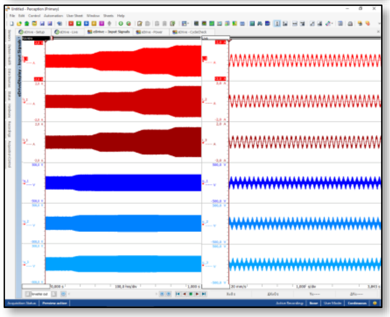


IDLE (stopped), after launching Pe
Display shows no data

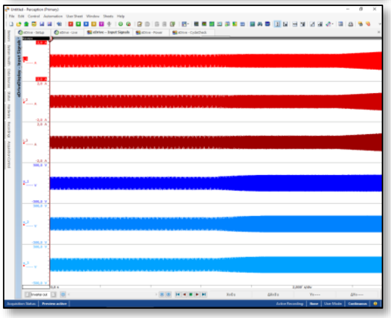
PAUSE
Display is split into two areas left a
Left REVIEW area shows no data
Right LIVE area shows the live tra
computed results



RECORD
Display is split into two areas left and right:
Left REVIEW area shows currently stored data and results.
Right LIVE area shows the live traces at the input and computed results



IDLE (stopped) after a previous recording was made:
Display shows last recorded data and computed results

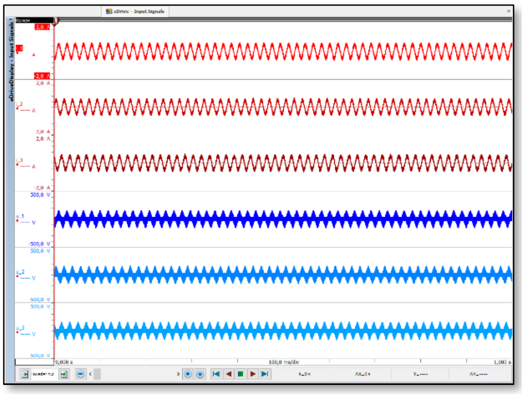


For the pictures above, the READ ONLY status of the sheets was removed and the background color of the displays was changed to white to improve printing.

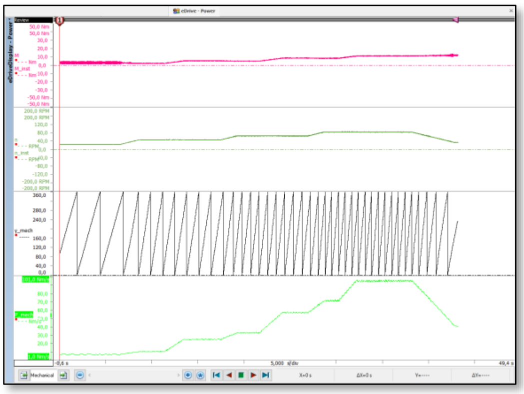
For more info on how to work with displays, please refer to the Perception software manual.

8.3 The eDrive – Display sheet with typical data after storage

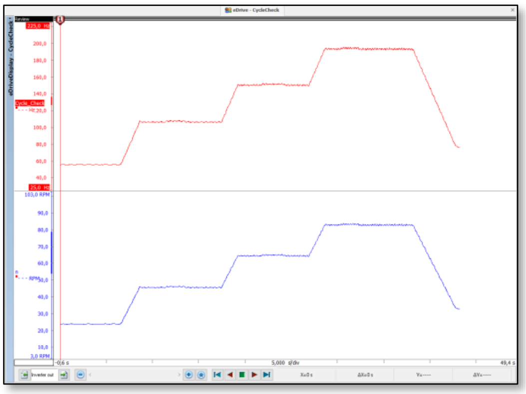
Once a recording is started, the three sheets and all their pages will be filled with the live raw data and the real time computed results. After stopping the displays then show the stored data and might look like the examples below.
These show typical data acquired during a run up test of a drive in several torque and speed steps.



Sheet **eDrive – Input signals**: Page INVERTER OUT shows i_1, i_2, i_3 and u_1, u_2, u_3



Sheet **eDrive – Power**: Page MECHANICAL shows M, n, γ_mech and P_mech



Sheet **eDrive – CycleCheck**: Page INVERTER OUT shows Cycle_Check and n

For the pictures above, the READ ONLY status of the sheets was removed and the background color of the displays was changed to white to improve printing.

8.4 Content of the created sheets and pages

Below is a complete listing of the three sheets created, the pages per sheet and the channels shown in these pages.

Sheet **eDrive – Input signals**

Page POWER SOURCE
- Instantaneous voltage and current channels

Page INVERTER OUT
- Instantaneous voltage and current channels

Page MECHANICAL
- Instantaneous values for torque and speed and mechanical angle (including event bit channels)

Sheet **eDrive – Power**

Page POWER SOURCE
- RMS values of all voltages and currents
- True power values per phase and total
- Apparent power values per phase and total
- Reactive power values per phase and total

Page INVERTER OUT
- RMS values of all voltages and currents
- True power values per phase and total
- Apparent power values per phase and total
- Reactive power values per phase and total

Page MECHANICAL
- Instantaneous values for torque and speed
- CycleMean values for torque and speed
- Mechanical angle
- Mechanical power

Page EFFICIENCIES
- Input (true) power, Output true power and mechanical power
- Inverter efficiencies for motor and generator mode
- Machine efficiencies for motor and generator mode
- For mode **Shaft and transmission**:
  Transmission efficiencies for motor and generator mode
- Total efficiencies for motor and generator mode

Sheet **eDrive – CycleCheck**

Page POWER SOURCE
- Cycle source and filtered cycle source signals used for the power source block
- Cycle master signal (= the result of the cycle detection) used for the power source block
- CycleCheck Signal (= fundamental frequency) used for the power source block

Page INVERTER OUT
- Cycle source and filtered cycle source signals used for the inverter output block
- Cycle master signal (= the result of the cycle detection) used for the inverter output block

- CycleCheck Signal (= fundamental frequency) used for the inverter output block

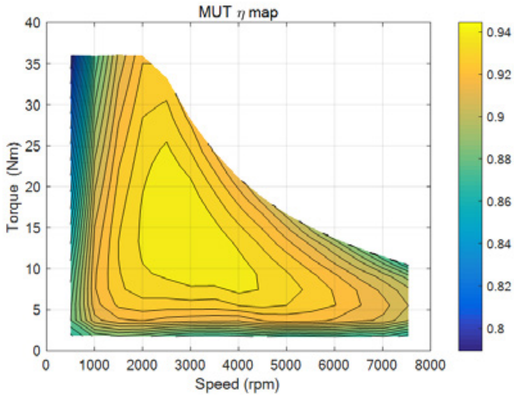
Page MECHANICAL

- Cycle source and filtered cycle source signals used for the mechanical block
- Cycle master signal (= the result of the cycle detection) used for the mechanical block
- CycleCheck Signal (= fundamental frequency) used for the mechanical block

9 Motor mapping

There are different diagrams to show the relation of three or more machine parameters. These diagrams are typically called motor maps, and the most common one is an efficiency map.

An efficiency map is a 3D diagram of efficiency η over torque M over speed n.
It shows the behavior of the e-machine at different torque / speed points and indicates where the machine should be used for maximum efficiency. Efficiency maps are used for optimization / validation and might be generated several times in the iterative optimization process of an electric drive.



Typical efficiency map created in MATLAB

This chapter gives an overview about the *process* to acquire the data and to compute the results needed for a motor map using the eDrive application, or – for more complex configurations – using the Perception software itself.

It explains how this process needs to be set up and followed to create motor maps effectively and fast. Using the tools of the eDrive option, it is very easy to gather the data for a motor map.

9.1 Different ways to create motor map data

There are three different ways to acquire the data for a motor map in eDrive or Perception:

- ***Real time*** using the **eDrive – Efficiency sheet** for real-time logging and map creation (and later re-analyse in MATLAB based on this data).
  This is the easiest and done in real time including the efficiency map, but is limited to results computed in the RT-FDB.
- ***Real time*** using the RT-FDB and the “Log to EXCEL / XML” feature of eDrive.
  This is easy and done in real time, but is limited to results computed in the RT-FDB. It also does not create the efficiency map.
- ***Post process*** after the recording using the FDB and the LOGFILE feature of Perception.
  This is more complex but offer nearly unlimited analysis capabilities for very advanced maps like iron or copper loss maps and much more.

> ***Please note that although they look very similar, the “Log to EXCEL” feature of eDrive and the “Logfile” feature of Perception are two different features.***

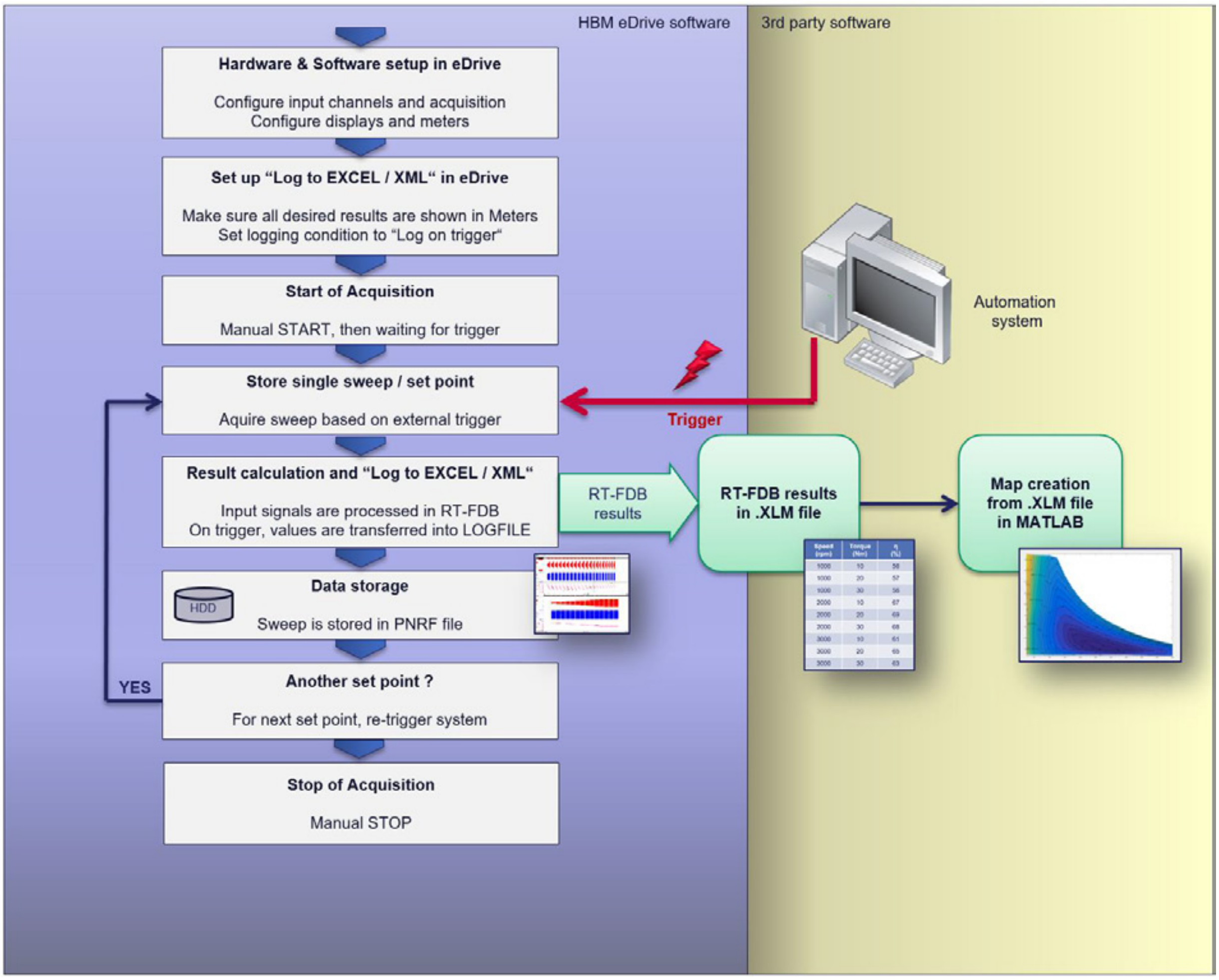
9.1.1 Different ways to create data for maps compared

Below is a high level comparison of the three different ways to create the input data for maps.

	eDrive Efficiency sheet	eDrive / Real time Log to XML	Perception / Post process Logfile
Setup	Very easy; all in eDrive	Very easy; all in eDrive	Needs special formulas, automation and logging setup
Analysis	RT-FDB	RT-FDB	FDB
Raw data storage	Optional	Optional	Required
Result creation	Real time (including MAP)	Real time	Post process
User formulas	✓	✓	✓
Complexity of calculations	Very limited	Very limited	Unlimited
Synchronous results	✓	✓	✓
Logging interval	Per sweep / trigger	Per sweep / trigger	Per sweep / trigger
Typical power calculations	✓	✓	✓
Special setups like 6ph motors	✓	✓	✓
Cross board synchronous math	✗	✗	✓
RPC result access	Synchronized	Synchronized	Not recommended as not synchronized
EtherCAT/CAN transfer	✓	✓	✗

9.2 Real time map creation in eDrive using "Log to EXCEL / XML"

The process to create maps in real time during acquisition is shown below.



Process flow using eDrive and the RT-FDB

BASIC STEPS:

Preparation

- Set the eDrive system into Multi-Sweep acquisition mode; select a sweep length long enough to capture at least one full cycle at lowest expected speed.
- Configure the eDrive system to measure all the relevant parameters and to compute all needed results. If needed add user formulas; note only asynchronous results can be logged.
- Adapt the cycle detector settings to work with all expected set points. If this is difficult, it might be better to use “reference pulse” as cycle source.
- Make sure all desired results are shown on the LIVE display of the eDrive system; if needed, add missing results into the MORE METERS.
- Set up **Log to EXCEL / XML** in the SYSTEM SETTINGS menu. Set it to **Automatic each trigger**.

Test process and result table creation (real time)

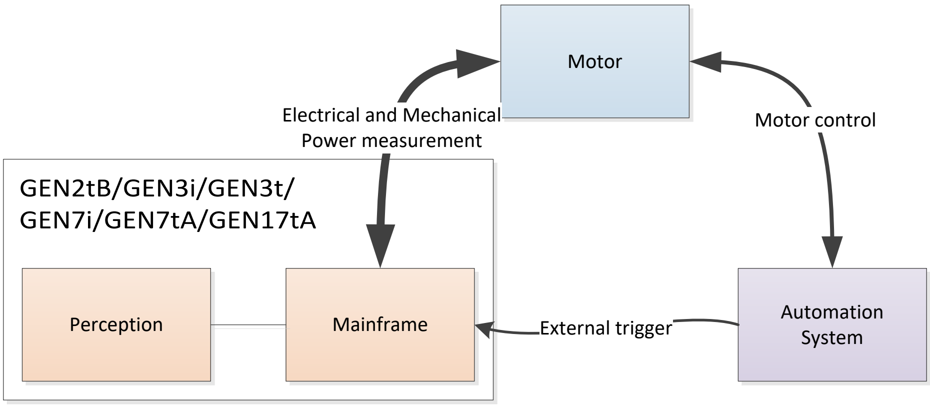
- Now drive the machine under test through the different set points.
  At each set point, send a trigger signal to the eDrive system.
- When finished with all set points, stop the acquisition.

Review results and map creation

- Review the created XML file.
- Now you can import this XML file into a graphical display software like MATLAB and show the resulting map.

9.3 Real time using the eDrive - Efficiency Sheet

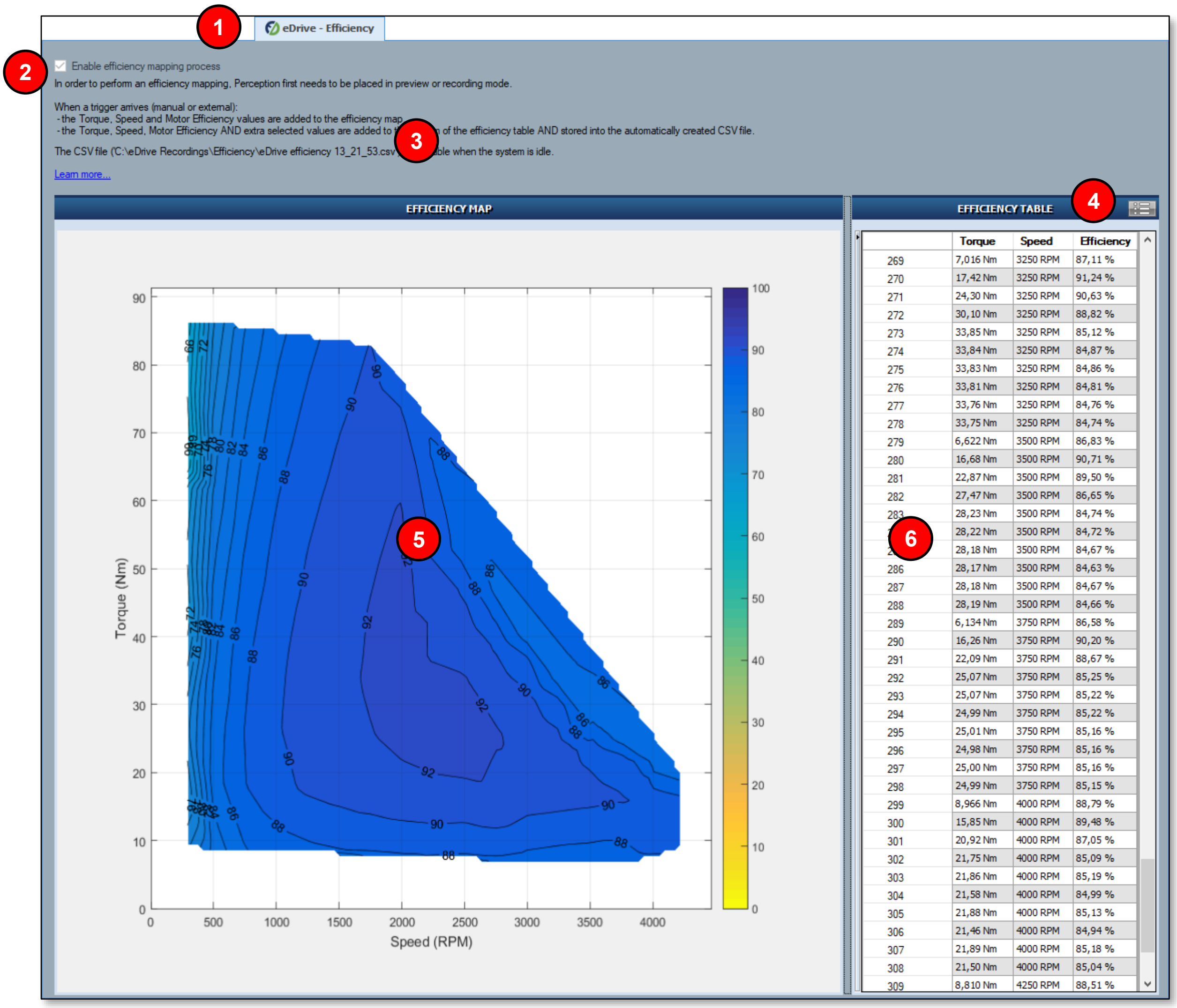
Efficiency mapping is done to measure motor efficiency at different torque / speed points and indicate where it should be used for maximum efficiency. The torque / speed points are called “set points”.
The typical setup of the hardware looks like:



To perform an efficiency mapping of an electrical motor (using manual or external triggers), eDrive has the efficiency mapping feature to support this workflow.

> ☞ ***The eDrive solution only does the setup of the GEN DAQ hardware and performs the measuring of the set points. It does NOT control the motor NOR generates the external trigger.***
>
> ☞ ***This is the recommended way to create data for maps.***
> ***This is because the setup here is by far the easiest and results are available in real time.***

An overview of the most important sections of the eDrive Efficiency sheet is shown below:



Picture H: ***eDrive – Efficiency Sheet***

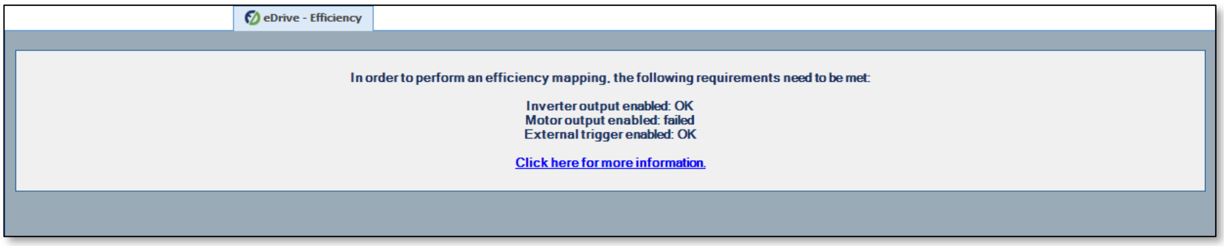
Item	Description	Item	Description	Item	Description
H-1	Selected Perception sheet **eDrive – Efficiency** is shown	H-3	Location of the CSV file in which the set point are written	H-5	Map containing the contour plot of the torque, speed and efficiency values out of the set points
H-2	Option to enable/disabled the efficiency mapping process	H-4	Setup for extra calculations to be added to the table and CSV file	H-6	Table containing all the set point values

9.3.1 *How to use the eDrive – Efficiency sheet*

The efficiency sheet is placed next to the Setup Sheet and Live Sheet and is always present.
In order to use the efficiency mapping the following requirement are needed:

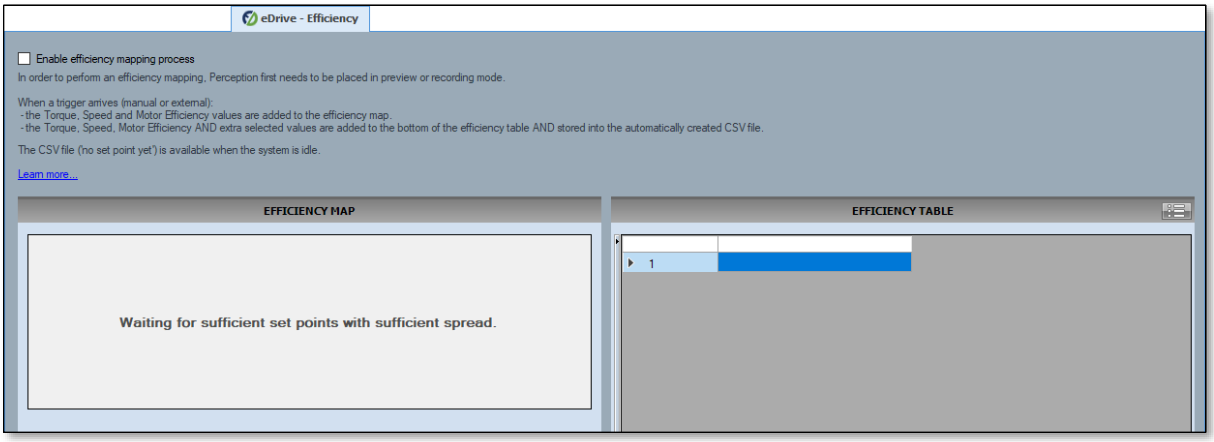
- An eDrive capable system needs to be connected
- The inverter output configuration needs to be enabled
- The motor output configuration needs to be enabled
- The external trigger need to enabled (default" enabled)
- The "MATLAB Runtime version R2016b" needs to be installed

The requirements are tracked all the time. When a setting changes which makes it impossible to perform an efficiency mapping, this is shown on the screen like below:



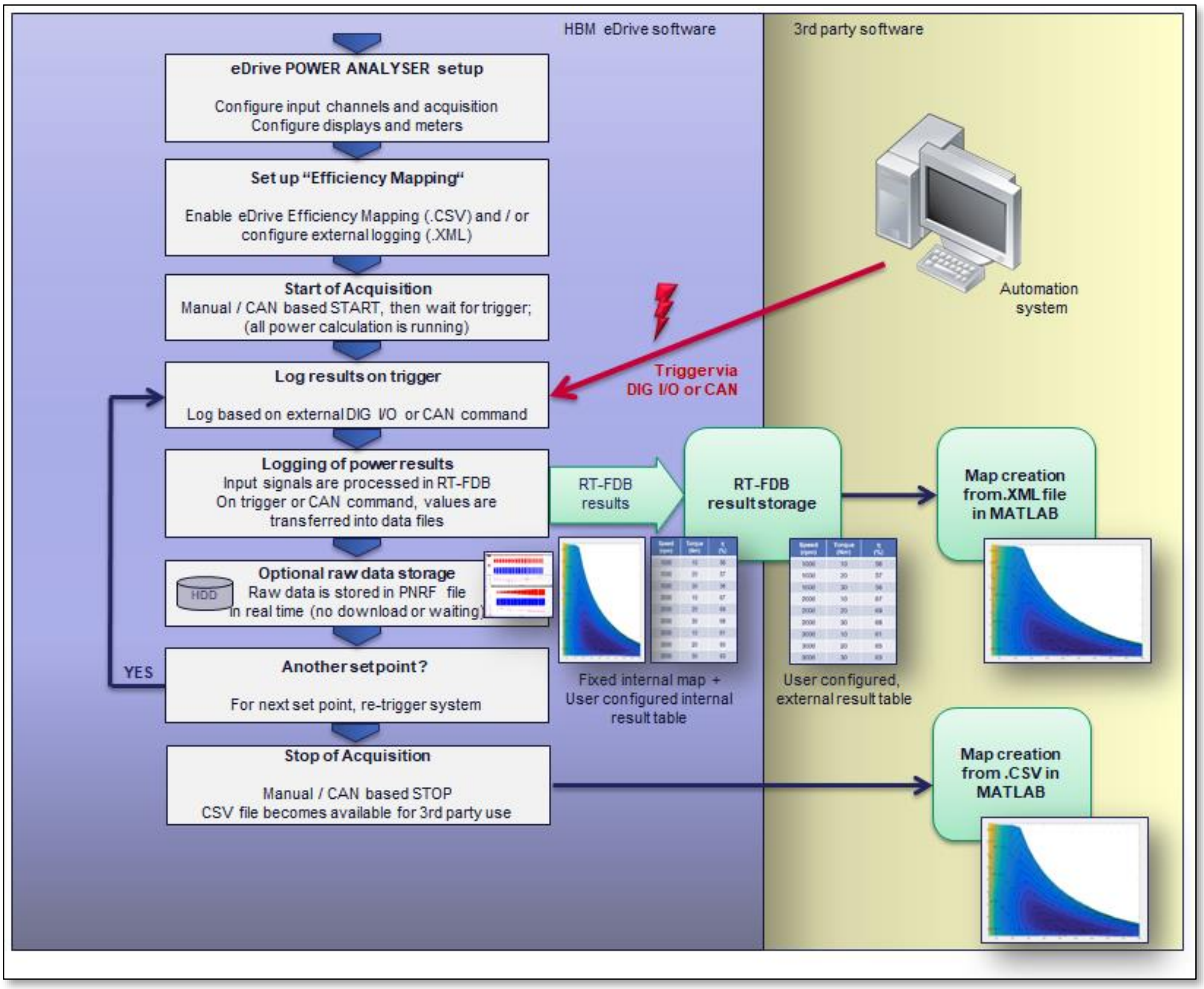
Efficiency sheet indicating requirements are not fulfilled (motor output is not used)

When the requirements are met, the text is removed and the following layout is shown:



Now it is possible to enable the efficiency mapping process.

9.3.2 How to create an efficiency map



Motor mapping process flow using "Efficiency Mapping" in a Multi-Sweep acquisition mode

To start mapping the efficiency, enable the efficiency mapping process in the "Efficiency" sheet and put Perception in PREVIEW or RECORDING mode.

- An external motor control system spins the motor to the requested speed and torque, waits a bit to become stable and sends out an external trigger to the GHS system.
- When the trigger arrives (manual or external), the Torque, Speed, Motor Efficiency and extra selected values calculated over the NEXT number of defined cycles are stored for this set point. If case an averaging period is selected (under Update rate settings), the values will be averaged using this period on top of the stored values.
- These set point values are added to the bottom of the efficiency table AND are stored into the automatically created CSV file.
  The table automatically scrolls to the last added set point values.
- A contour plot of the stored set points is also displayed if applicable. Be aware that a contour plot can only be displayed with at least 3 set points in which the torque/speed values are NOT on a mathematical straight line.

These steps can be repeated if more set points are required.
When the requested set points are measured, the GEN DAQ system can be set to idle again.

The CSV automatically gets a name and is stored in a subfolder (called "Efficiency") of the current storage location:
- In preview mode: "eDrive efficiency hh_mm_ss.csv". Where the time is the time of the first set point
- In recording mode: "<recordingname>.csv"

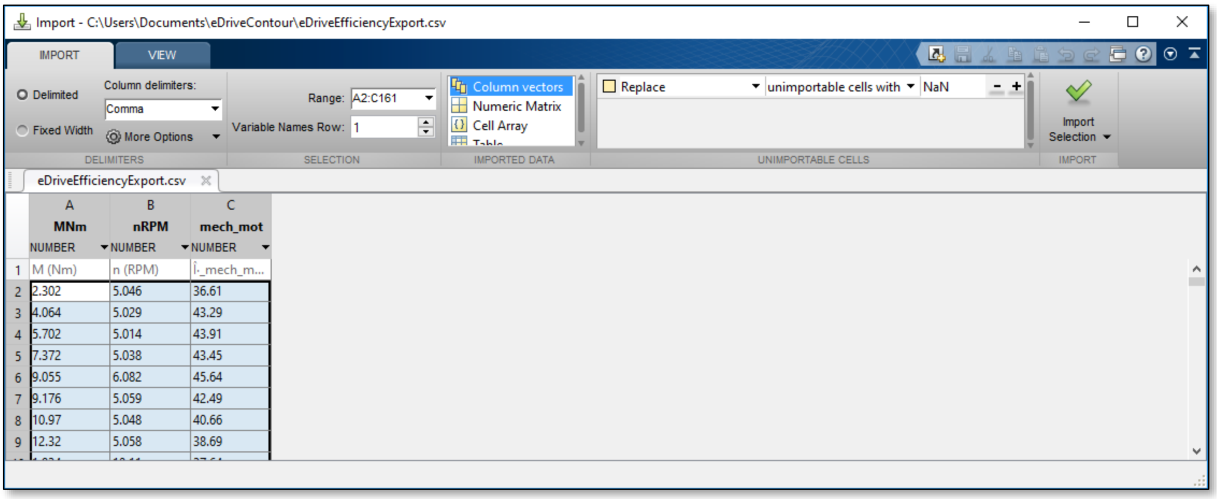
Because the process needs write access to the CSV file, the file is only available to the user when Perception is set to IDLE again.

9.3.3 *Importing efficiency map set points into MATLAB*

To import the set points from the CSV file into MATLAB, the function “importdata” can be used in MATLAB.

>> eff_map_data =importdata(‘eDriveEfficiencyExport.csv')

Another option is to double click the .csv in the MATLAB explorer to open the “Import” wizard. Here you can configure how the data will be imported.

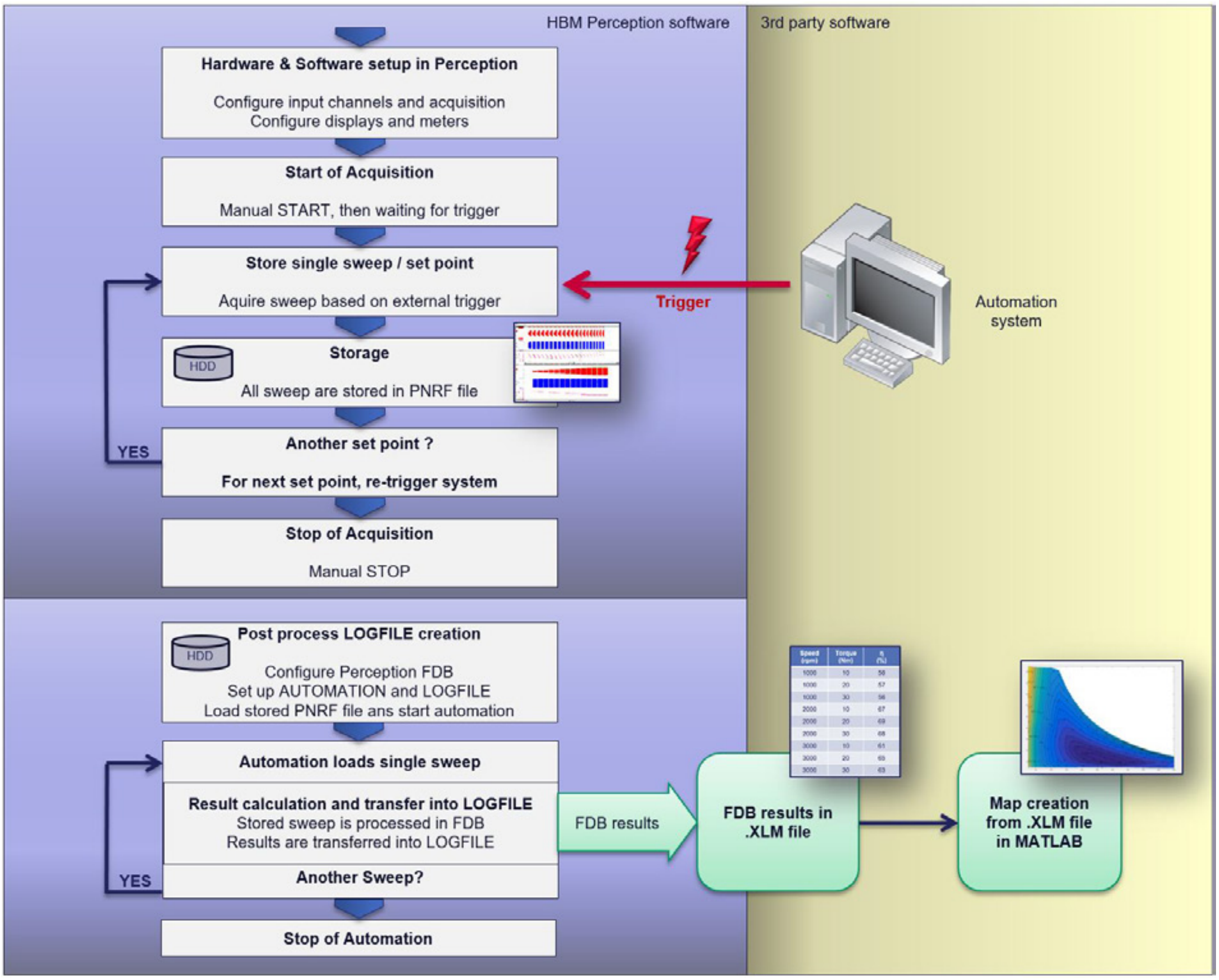


MATLAB import wizard

An example CSV file with efficiency mapping data and an MATLAB script file (.m) for creating a contour plot can be found in C:\PerceptionExampleData.
This script imports the example data and renders a contour plot (matching the one shown in Perception). This example can be extended or adapted to your personal needs.

9.4 Post process map creation in Perception using LOGFILE

The process to create maps in post process is shown below.



Process flow using Perception and the FDB in post process

BASIC STEPS:

Preparation

- Set the eDrive system into Multi-Sweep acquisition mode; select a sweep length long enough to capture at least one full cycle at lowest expected speed.

- Configure the eDrive system to measure all the relevant parameters.

  Hint: Create REVIEW FORMULAS now; they are not needed now, but will be useful later.

Test process

- Now drive the machine under test through the different set points.
  At each set point, send a trigger signal to the eDrive system.

- When finished with all set points, stop the acquisition.

Result table creation (post process)

- Load the acquired experiment (if not loaded any longer).
  This will also load the FDB formulas if these were created.

- Modify/extend the formulas if needed and add formulas to create numerical results per sweep (like @Mean).

- Configure the Automation of Perception (**Setup process display….**) to run through all sweeps.

- Configure the LOGFILE to store all relevant results per sweep.

- Start Automation (**Process display**), and wait until it has finished.

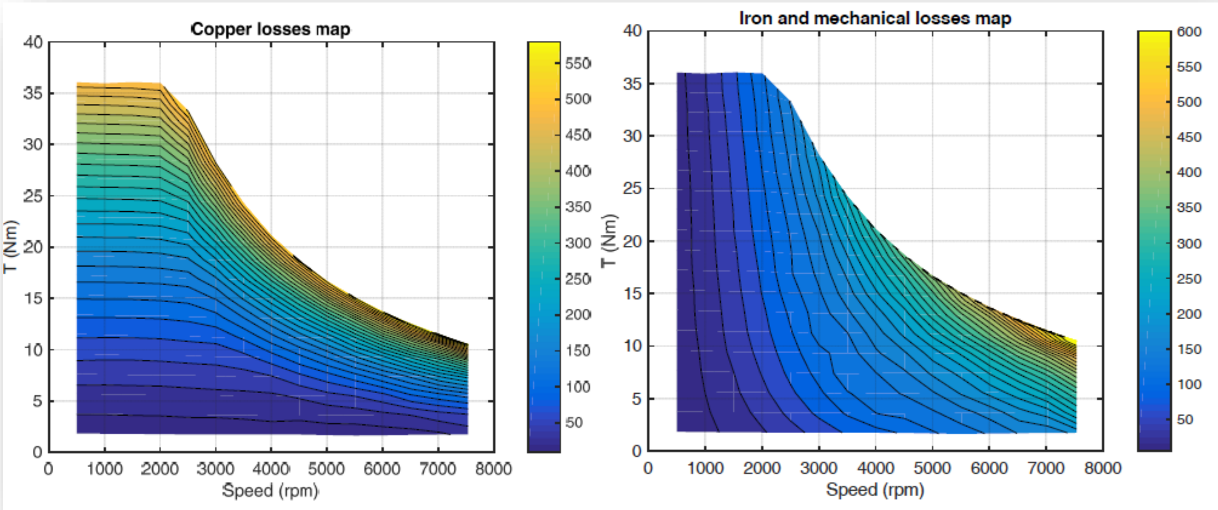
Review results and map creation

- Review the created XML file.

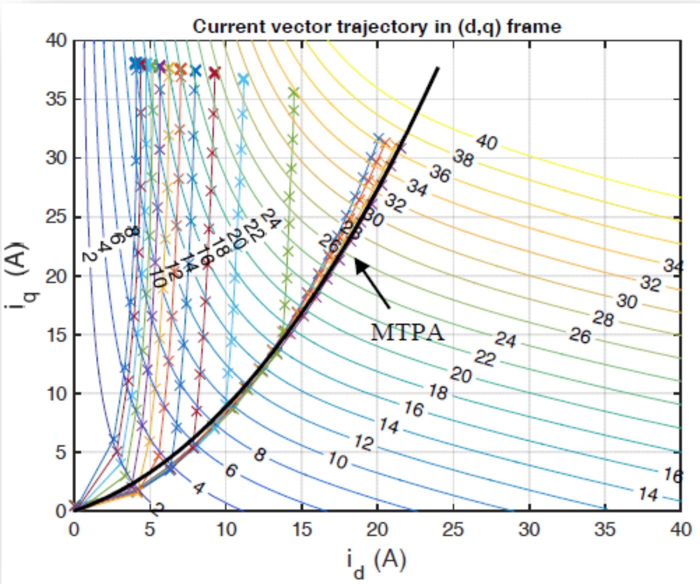
- Now you can import this XML file into a graphical display software like MATLAB and show the resulting map.

9.4.1 Some advanced maps

Below are a few examples of advanced maps where the required calculation were done in Perception. Due to the complexity of the required formulas, this map data was created in Perception post process.



Example of copper and iron losses maps



Example of MTPA (Maximum Torque Per Amp) trajectory map

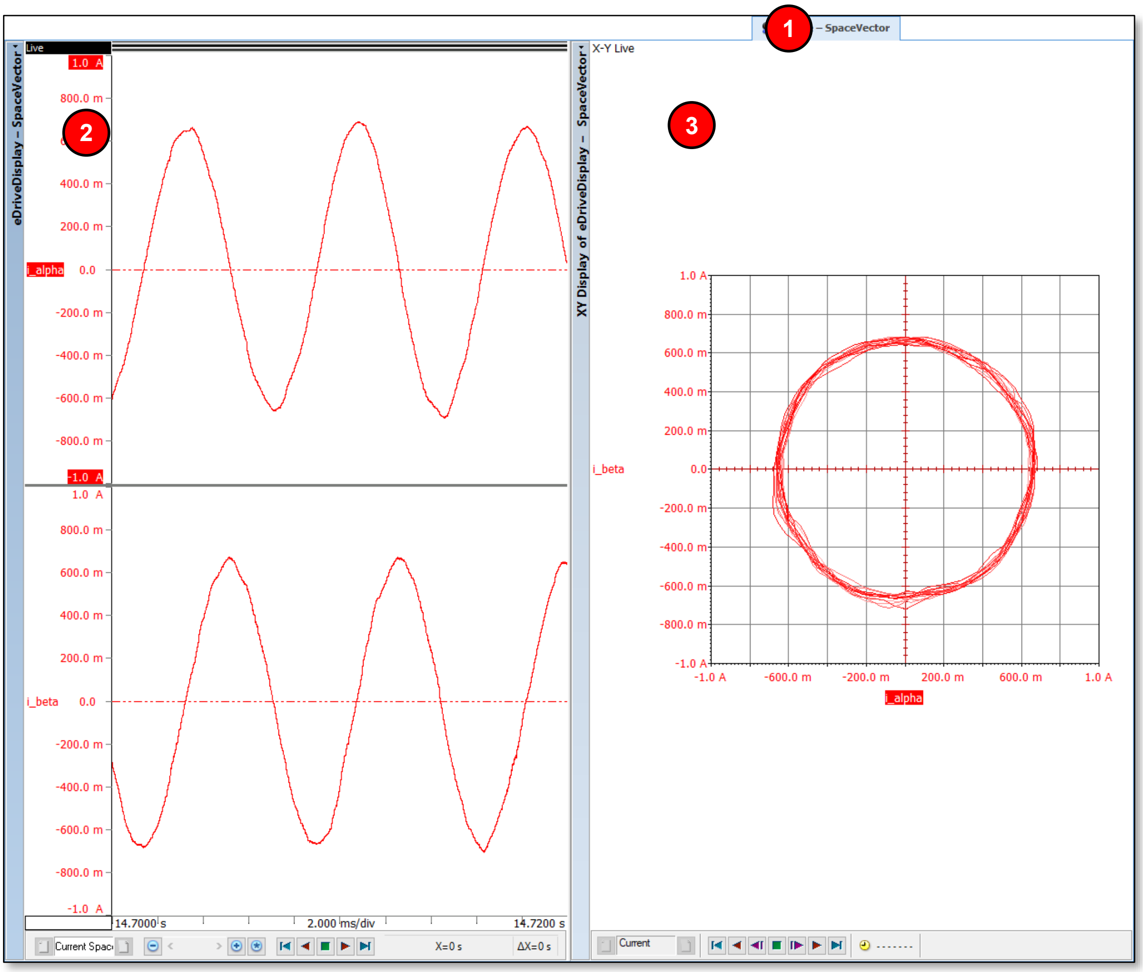
More information on how to create the data for such maps including the basic formulas can be found here: www.hbm.com/en/6207/white-paper-efficiency-and-loss-mapping-of-ac-motors/

10 The eDrive – Space Vector sheet

10.1 Overview

When the space vector formulas are enabled (see section 6.6.2), extra real-time formulas are added to calculate the space vector transformation of the 3phase inverter output currents.

The results of these calulations are shown in a special created display sheet called “eDrive - SpaceVector”. On this sheet a display and XY display are present containing the calculation results.



Picture I: ***eDrive – SpaceVector*** sheet

Item	Description	Item	Description	Item	Description
I-1	Selected Perception sheet **eDrive – SpaceVector** is shown	I-2	Display containing the alpha and beta current space vector	I-3	XY-Display containing the alpha and beta current space vector

For the picture above, the READ ONLY status of the sheets was removed and the background color of the displays was changed to white to improve printing.

For more info on how to work with displays, please refer to the Perception software manual.

10.2 General behavior of the displays in these sheets

The sheet is created automatically and the content is also changed dynamically when the inverter output measurement configuration is changed by the user. The sheet is set to READ ONLY, so the user cannot change the sheet.

When the acquisition state is **Preview** or **Record,** the displays show the live computed results only.
In Idle the last recorded computed results are shown

10.3 Content of the created sheet and pages

There is only 1 page called “Current SpaceVector” and it contains the alfa and beta component from the space vector transformation of the inverter output currents

11 Real time formulas

This chapter gives an overview about the structure, the process flow and some details of the **real time formula database** (RT-FDB) used in the eDrive application of HBM Perception software.

It explains how these formulas work for Perception version 7.30.

It does not explain the specific functions (@Function) of the RT-FDB in detail.
For such explanations please refer to the Perception user manual or the HELP in the RT-FDB sheet.

It also does not explain the review formulas in the Formula Database (FDB).
There are a lot of similarities between RT-FDB and FDB, but they are not the same and the formulas are not compatible (so no copy & paste between RT-FDB and FDB). But due to the similarities the working process of the FDB can be deducted from this guide as well.

Used abbreviations:

RT - FDB	=	*Real Time Formula Database*
FDB	=	*Formula Database*
Lower case characters	=	*Instantaneous values of electrical entities like u, i, p*
Capital characters	=	*RMS values of electrical entities like U, I, P*

11.1 Overall structure

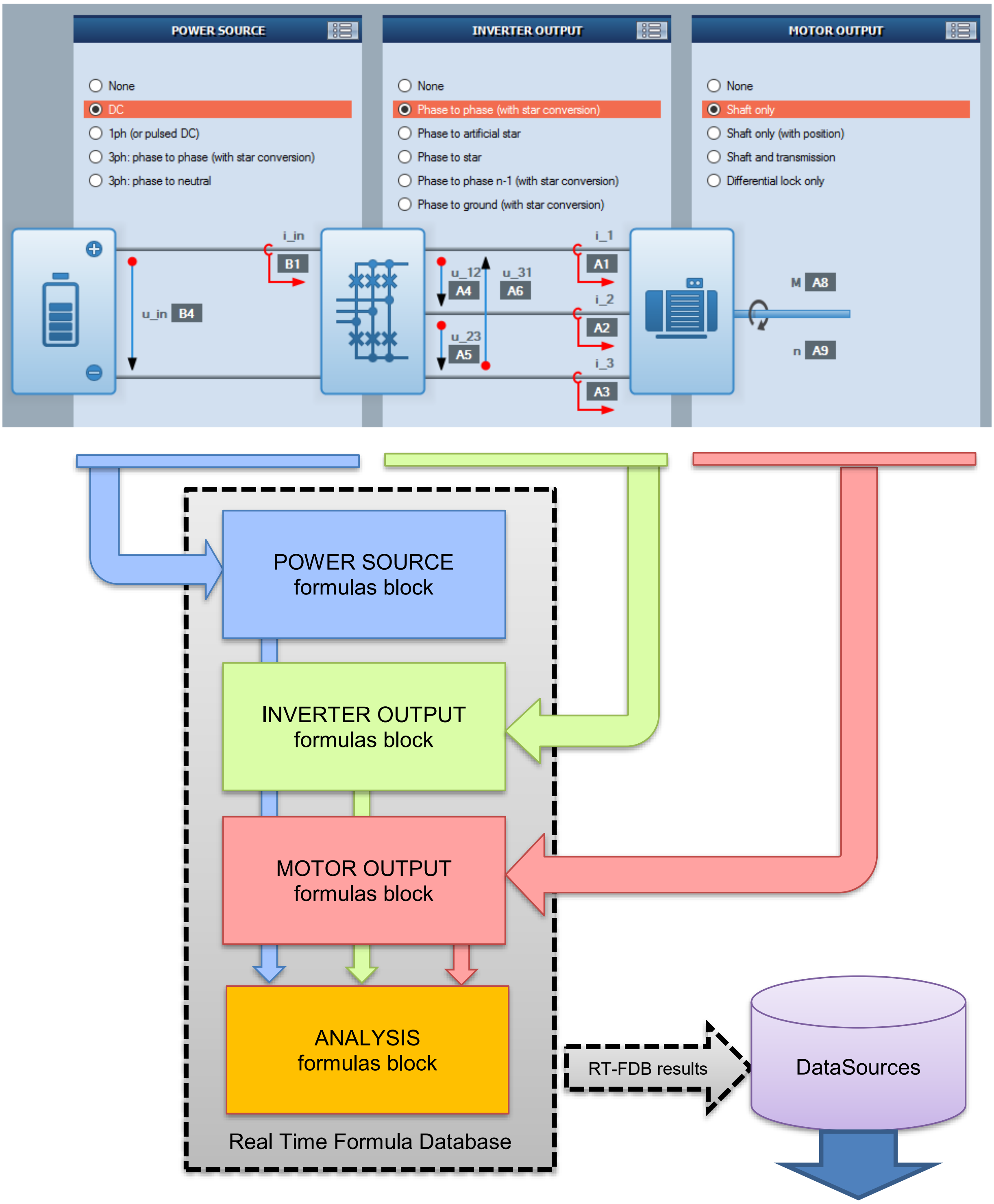
The formulas in the RT-FDB of Perception are automatically created when the eDrive option is used. Also, some parameters of the formulas are changed on the fly when certain menu entries in the eDrive user interface are made. Thus the user does not need to set up these formulas on his own.

Still the user might be interested to understand how they work. The user might also use the results of the automatically created formulas to create his own additional formulas.

The easiest way to create user RT formulas based on the eDrive RT formulas is to copy the formulas from the RT-FDB sheet into EXCEL, unload the eDrive option using the "Sheet manager" in Perception and then to copy the needed parts of the RT formulas back into Perception´s RT-FDB sheet.

11.2 From the user selections in the software to the RT-FDB blocks

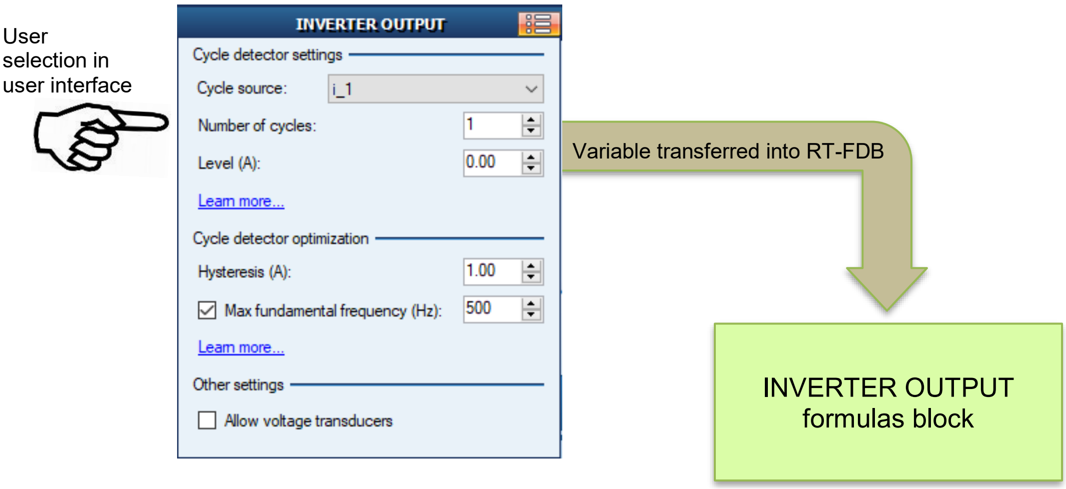
Each “block” in the eDrive user interface ”creates” its own block of RT formulas in the RT-FDB sheet.



Further use for displays, meters, RPC calls, EtherCAT and CAN transfer….

On top of this "block generation", based on the selected configuration, there are additional user inputs in the eDrive Setup sheet which are used as parameters in the RT-FDB.

An example of this is the parameter setting for cycle detection. The entries made by the user in the user interface are transferred into RT-FDB variables, and then used as parameters for the Cycle Detect functions.



This functionality is explained in more detail later in this chapter.

For simplification, only the "simple drive line" configuration will be explained in this chapter. More complex configurations like "Motor-Generator" test might have more blocks, but the fundamental concept stays the same.

11.3 The INVERTER OUTPUT formulas block

11.3.1 Overview and process flow

As the INVERTER OUTPUT block is the most important one, this block is explained first.

☞ ***All the explanations given here with respect to cycle parameter setting, cycle detection, cycle master and cycle check are valid for other blocks as well.***

☞ ***Note that in the actual RT-FDB the POWER SOURCE block (if present) is positioned on top, and the INVERTER OUTPUT block is second.***

☞ ***Only the basic selections in the user interface and their effect on the RT-FDB are explained here. For some specials like Aron (n-1) circuitry, details are not included.***

The overall explanation is done for a “Phase to (artificial) Star” configuration.
Other variants are noted but not necessarily explained in detail.

The “process flow” for the INVERTER OUTPUT formulas is shown below:

****
#region AC3_PN

Allocation of input channels to variables

List of external variables used
(variables not defined in this block)

Allocation of cycle detector settings to variables

Cycle source filter, cycle master detection and cycle checking

If selected configuration is “Phase to phase”, “Phase to ground” or “Phase to Phase n-1”:
Conversion of measured inputs to phase voltages and phase currents

RMS calculation
per phase and cumulative for all u and i

Instantaneous power calculation
per phase and cumulative

P, S, Q calculation
per phase and cumulative

λ calculation
per phase and cumulative

If selected in SYSTEM SETTINGS:
φ, cosφ and RMS calculations
per fundamental u and i of each phase. Plus current space vectors

#endregion AC3_PN
****

11.3.2 RT-FDB formulas INVERTER OUTPUT block – Phase to (artificial) Star configuration

(Main formulas only, comment lines are deleted):

i_1	Recorder_A.i_1
i_2	Recorder_A.i_2
i_3	Recorder_A.i_3
u_1	Recorder_A.u_1
u_2	Recorder_A.u_2
u_3	Recorder_A.u_3

Cycle_source	Recorder_A.i_1
Cycle_count	1
Cycle_level	0
Cycle_hyst	1
Cycle_holdoff	0.001
Cycle_filter_type	1
Cycle_cutoff_frequency	1000
Cycle_direction	0

Cycle_source_filt	@HWFilter (RTFormulas.Cycle_source ; RTFormulas.Cycle_filter_type ; RTFormulas.Cycle_cutoff_frequency)
Cycle_Master	@CycleDetect (RTFormulas.Cycle_source_filt ; RTFormulas.Cycle_count ; RTFormulas.Cycle_level ; RTFormulas.Cycle_hyst ; RTFormulas.Cycle_holdoff ; RTFormulas.Cycle_direction)
Cycle_Check	@CycleFrequency (RTFormulas.Cycle_Master)

I_1	@CycleRMS (RTFormulas.i_1 ; RTFormulas.Cycle_Master)
I_2	@CycleRMS (RTFormulas.i_2 ; RTFormulas.Cycle_Master)
I_3	@CycleRMS (RTFormulas.i_3 ; RTFormulas.Cycle_Master)
∑_I	(RTFormulas.I_1 + RTFormulas.I_2 + RTFormulas.I_3) / 3

U_1	@CycleRMS (RTFormulas.u_1; RTFormulas.Cycle_Master)
U_2	@CycleRMS (RTFormulas.u_2; RTFormulas.Cycle_Master)
U_3	@CycleRMS (RTFormulas.u_3; RTFormulas.Cycle_Master)
∑_U	(RTFormulas.U_1 + RTFormulas.U_2 + RTFormulas.U_3) / 3

p_1	RTFormulas.u_1 * RTFormulas.i_1
p_2	RTFormulas.u_2 * RTFormulas.i_2
p_3	RTFormulas.u_3 * RTFormulas.i_3
p	RTFormulas.p_1 + RTFormulas.p_2 + RTFormulas.p_3

P_1	@CycleMean (RTFormulas.p_1 ; RTFormulas.Cycle_Master)
P_2	@CycleMean (RTFormulas.p_2 ; RTFormulas.Cycle_Master)
P_3	@CycleMean (RTFormulas.p_3 ; RTFormulas.Cycle_Master)
P	RTFormulas.P_1 + RTFormulas.P_2 + RTFormulas.P_3

S_1	RTFormulas.U_1 * RTFormulas.I_1
S_2	RTFormulas.U_2 * RTFormulas.I_2
S_3	RTFormulas.U_3 * RTFormulas.I_3
S	RTFormulas.S_1 + RTFormulas.S_2 + RTFormulas.S_3

Q_1	@Sqrt (RTFormulas.S_1 * RTFormulas.S_1 - RTFormulas.P_1 * RTFormulas.P_1)
Q_2	@Sqrt (RTFormulas.S_2 * RTFormulas.S_2 - RTFormulas.P_2 * RTFormulas.P_2)
Q_3	@Sqrt (RTFormulas.S_3 * RTFormulas.S_3 - RTFormulas.P_3 * RTFormulas.P_3)
Q	RTFormulas.Q_1 + RTFormulas.Q_2 + RTFormulas.Q_3

λ_1	RTFormulas.P_1 / RTFormulas.S_1
λ_2	RTFormulas.P_2 / RTFormulas.S_2
λ_3	RTFormulas.P_3 / RTFormulas.S_3
λ	RTFormulas.P / RTFormulas.S

The formulas are identical for Phase to star and Phase to artificial star configuration.

Cycle_Master is the calculated cycle signal for the selected cycle source with the currently selected cycle parameters.

Cycle_Master always need a trace of type @HWFilter as input.

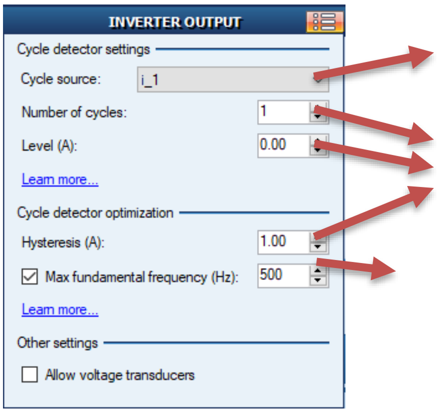
Cycle_Check is the frequency of the Cycle_Master signal. This signal gives a quick overview if all cycles are found. It must have the same wave shape as the rpm signal and must not have any peaks in it.

φ, cosφ, fundamental RMS values and current space vectors (not shown in above list) are only computed if selected in the user interface (See section 6.6.2).

11.3.3 Cycle detector settings in the RT-FDB

The cycle detect settings are critical for proper detection of the fundamental frequency and therefore for valid results. While the user enters these parameters in the user interface, they are automatically transferred as variables into the RT-FDB and used there as input parameters for some functions.

The “mapping” from the user interface into the RT-FDB is shown below.



Cycle_source	@HWFilter (Recorder_A.i_1 ; RTFormulas.Cycle_filter_type ; RTFormulas.Cycle_cutoff_frequency)
Cycle_count	1
Cycle_level	0
Cycle_hyst	1
Cycle_holdoff	0,001
Cycle_filter_type	1
Cycle_cutoff_frequency	1000

Cycle_source is the selected (and always filtered) source channel.
If “Max. fundamental frequency” is selected, the formula reflects this.

Cycle_count, **Cycle_level** and **Cycle_hyst** are the parameters for the # of cycles for averaging and the hysteresis settings.

Cycle_holdoff and **Cycle_cutoff_frequency** are the parameters for the cycle source filter function.

Cycle_filter_type defines if the filter is BESSEL (Cycle_filter_type = 1) or WIDEBAND (Cycle_filter_type = 0).

If Max. fundamental frequency not enabled, the source channel for the Cycle Detect is still the output of a @HWFilter function. But in this case the filter is set to WIDEBAND.

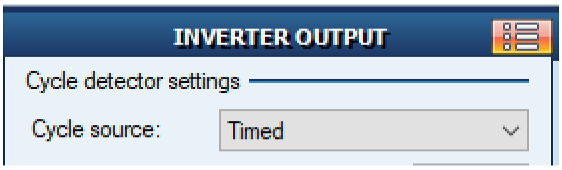
As a special, cycle detector settings can even be changed “on the fly” being in PAUSE mode. This is usually not possible for settings of the RT-FDB, which can only be changed while in STOP.

> ***If auto-timed mode in the system settings*** is enabled, values are calculated at least once a second (See section ***6.6.4.2***).

11.3.4 Time interval based calculation

The cycle source can be switched from cycle based to TIMED. Then the calculation is no longer done over detected cycles but on a fixed time interval. This might be useful for situations where cycle detection is difficult or there are just no cyclic signals.

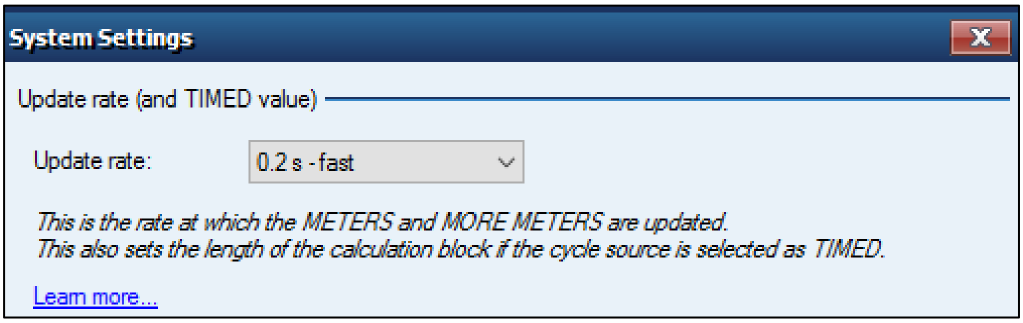
This selection is there in the INVERTER OUTPUT context menu, Cycle source dialog:



If set to TIMED, this is reflected in the RT-FDB, where the Cycle_Master formula will change to:

Cycle_Master @CycleInterval (0.2)

The TIME INTERVAL used is defined in the SYSTEM SETTINGS menu, as shown below.

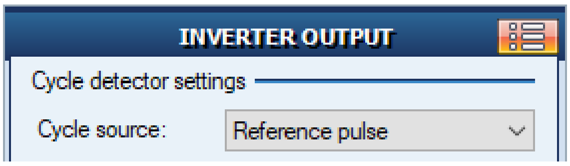


The update rate can be selected as 200 ms or 500 ms or 1 s.

> ***Using the TIMED calculation always gives a result. However, for cyclic signals it also introduces an error due to incomplete cycles being within the calculation time window.***

11.3.5 Reference pulse based calculation

The cycle source can be switched from cycle based to REFERENCE PULSE. Then the calculation is no longer done over detected cycles but based on the reference pulse coming from the speed sensor. Thus it is synchronized to a mechanical revolution of the shaft.
This might be useful for situations where cycle detection is difficult or there are just no cyclic signals.



> ***If auto-timed mode in the system settings is enabled, values are calculated at least once a second (See section 6.6.4.2).***

11.3.6 Cycle_source_filt trace

The Cycle_source_filt_trace is a filtered input signal needed as input to the cycle detect function.
This trace is always needed as first step before a cycle detect can be done.

Cycle_source_filt	@HWFilter (RTFormulas.Cycle_source ; RTFormulas.Cycle_filter_type ; RTFormulas.Cycle_cutoff_frequency)

The filter type can be set to 1=Bessel or 0=WIDEBAND.

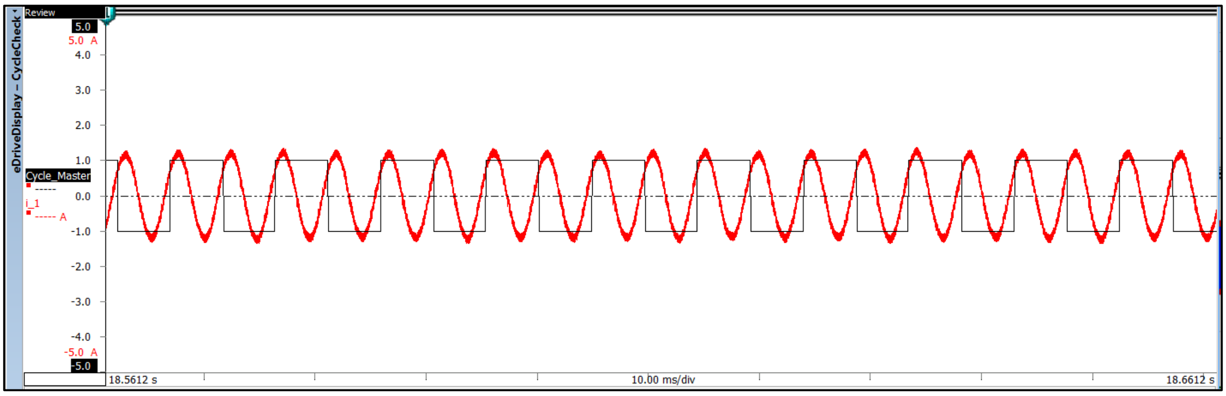
> ***The @CycleDetect function always needs the result of the @HWFilter function as input, otherwise it will not deliver a result.***

11.3.7 Cycle_Master trace

The Cycle_Master is the result of the cycle detection, done on the channel Cycle_source_filt.
It reflects the fundamental frequency over the selected number of cycles.

Cycle_Master	@CycleDetect (RTFormulas.Cycle_source_filt ; RTFormulas.Cycle_count ; RTFormulas.Cycle_level ; RTFormulas.Cycle_hyst ; RTFormulas.Cycle_holdoff ; RTFormulas.Cycle_direction)

The Cycle_Master is a square waveform indicating the cycles detected and used for analysis.
A typical Cycle_Master trace (black) and the Cycle_source trace i_1 (red) are shown below.



> ***The Cycle_Master signal as computed in the RT-FDB can be shown live and is stored so it can be shown again in review. It cannot, however, be reused in the FDB as is, but must be recalculated there.***
>
> ***If "Max fundamental frequency" is enabled, a filter will be applied to the Cycle_source signal (for cycle detection only, not for power calculation). Thus the resulting Cycle_Master trace will be phase shifted compared to the Cycle_source channel. This has no effect on the computed results, as these need to be done over a CYCLE, and not necessarily from zero crossing to zero crossing.***

11.3.8 Cycle_Check trace

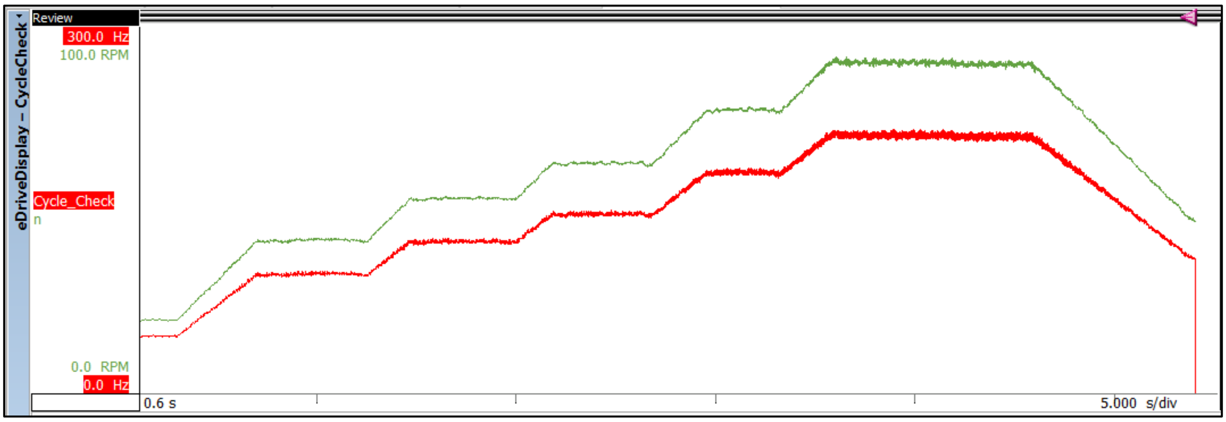
The Cycle_Check trace is the frequency of the Cycle_Master trace.

Cycle_Check	@CycleFrequency (RTFormulas.Cycle_Master)

As the Cycle_Check trace is basically the same wave shape as the trace n (= rpm), it is easy to check if the cycle detection worked by comparing these two traces.
Missed cycles would lead to a “jump” in the frequency to half of the value or less, so a big peak downwards. Extra, wrong cycles would lead to a “jump” in the frequency to double the value or more, so a big peak upwards.

A “smooth” Cycle_Check trace following the wave shape of the n trace is an immediate proof that the cycle detection worked correctly.

A typical Cycle_Check trace (red) and the corresponding trace n (green) are shown below.

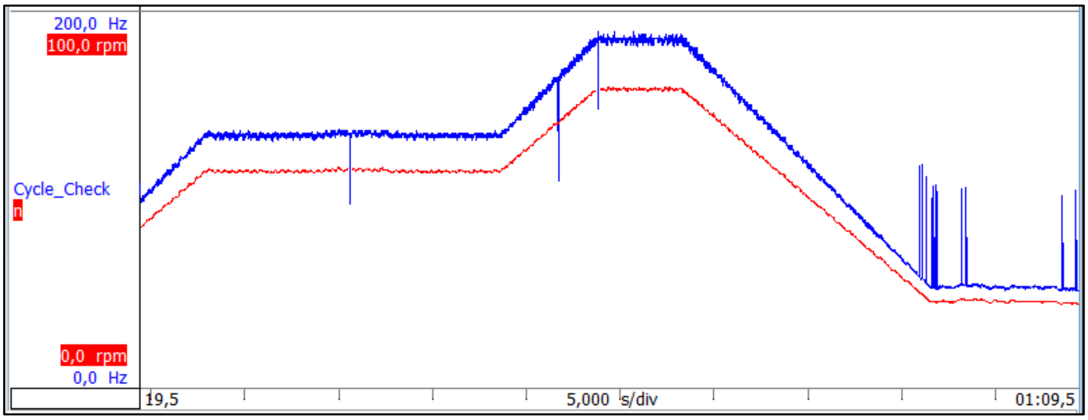


Always check the Cycle_Check trace against the n trace. They must have the same wave shape and the Cycle_Check trace must not have any peaks. This proves that the cycle detection worked correctly.

The ratio between the Cycle_Check trace and the n trace is defined by the number of pole pairs and the number of cycles used for cycle detection. Also, one is scaled per minute (rpm), the other one per second (Hz).
For asynchronous machines, also the slip is a factor between the two trace values.

If auto-timed mode in the system settings is enabled (See section 6.6.4.2) and kicks-in during a recording, it can be clearly recognized by an exact cycle check value of 1.00.

The picture below shows the Cycle_Check trace (blue) of a failed cycle detection (peaks in signal).


200,0 Hz
100,0 rpm
Cycle_Check
n
0,0 rpm
0,0 Hz
19,5
5,000 s/div
01:09,5

11.3.9 Phase to Phase configuration

For a “Phase to phase” configuration, a “delta to star” conversion is performed for the voltages.

In this case the recorded channels are different (as they now are “phase to phase” voltages) as shown below.

i_1	Recorder_A.i_1
i_2	Recorder_A.i_2
i_3	Recorder_A.i_3
u_12	Recorder_A.u_12
u_23	Recorder_A.u_23
u_31	Recorder_A.u_31

Then a “delta to star“ conversion is performed in the RT-FDB which results in the “normal” phase to star voltages.

u_1	(RTFormulas.u_23 + 2 * RTFormulas.u_12) / 3
u_2	(RTFormulas.u_23 - RTFormulas.u_12) / 3
u_3	((RTFormulas.u_12 + 2 * RTFormulas.u_23) / 3) * -1

After this intermediate step all the calculations are nearly the same as for a “phase to star” configuration. Only other difference is that later on the RMS values for the phase to phase voltages are also calculated with extra formulas in the RT-FDB, as shown below.

U_12	@CycleRMS (RTFormulas.u_12; RTFormulas.Cycle_Master)
U_23	@CycleRMS (RTFormulas.u_23; RTFormulas.Cycle_Master)
U_31	@CycleRMS (RTFormulas.u_31; RTFormulas.Cycle_Master)
∑_U_PP	(RTFormulas.U_12 + RTFormulas.U_23 + RTFormulas.U_31) / 3

> ***The MAIN METERS of the eDrive sheets always show the “Phase voltages”, even in a “Phase to phase” configuration. In case the “Phase to Phase RMS values” should be shown, these are available in the data sources and can be shown in the MORE METERS section.***

11.3.10 Phase to Phase n-1 configuration

The concept to deal with this configuration is similar to what was described for the “Phase to Phase” configuration.

The measured signals in this configuration are only two voltages and two currents, as shown below.

i_1	Recorder_A.i_1
i_3	Recorder_A.i_3
u_12	Recorder_A.u_12
u_32	Recorder_A.u_32

Then the conversion is calculated to get to the phase voltages and the third current.

u_1	RTFormulas.u_12 - ((RTFormulas.u_12 + RTFormulas.u_32) / 3)
u_2	(((-1) * RTFormulas.u_12) - RTFormulas.u_32) / 3
u_3	RTFormulas.u_32 - ((RTFormulas.u_12 + RTFormulas.u_32) / 3)
i_2	(-1) * RTFormulas.i_1 - RTFormulas.i_3

From here all the formulas are the same as for Phase to Phase.

11.3.11 Phase to Ground configuration

Again, the concept to deal with this configuration is similar to what was described for the "Phase to Phase" configuration.

The measured signals here are the three currents and the three voltages with respect to ground, as shown below.

i_1	Recorder_A.i_1
i_2	Recorder_A.i_2
i_3	Recorder_A.i_3
u_1G	Recorder_A.u_1G
u_2G	Recorder_A.u_2G
u_3G	Recorder_A.u_3G

First a conversion is calculated to get to the phase voltages.

u_1	(2 * RTFormulas.u_1G - RTFormulas.u_2G - RTFormulas.u_3G) / 3
u_2	(2 * RTFormulas.u_2G - RTFormulas.u_3G - RTFormulas.u_1G) / 3
u_3	(2 * RTFormulas.u_3G - RTFormulas.u_1G - RTFormulas.u_2G) / 3

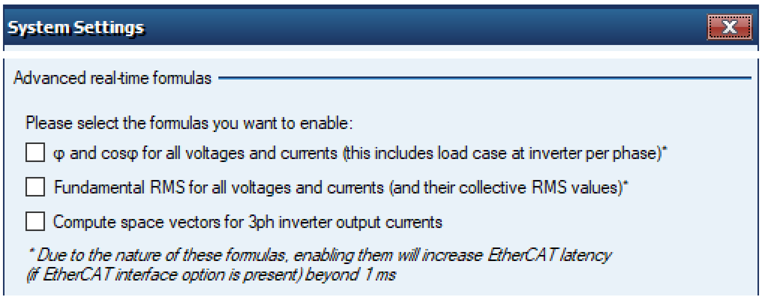
From here all the formulas are the same as for Phase to Phase.

Note that the conversion for Phase to Ground is different from the conversion for Phase to Phase: as the voltage reference is different (ground or star potential), the formulas are different.

The Phase to ground measurement is typically used with High Voltage passive dividers. As these require a grounded terminal, each phase is then measured to ground.

11.3.12 φ, cosφ, fundamental RMS and space vector calculations

The calculation of φ, cosφ, the RMS values of the fundamentals and space vector for inverter output can be enabled in the SYSTEM SETTINGS menu of the eDrive sheet. These calculations are disabled by default since they are not always needed and they consume a lot of computing power.



If enabled, the respective formulas are inserted into the INVERTER OUTPUT formula block at the very end.

Min_fund_frequency	10
φ_fund_1	@CycleFundamentalPhase (RTFormulas.u_1 ; RTFormulas.i_1 ; RTFormulas.Cycle_Master ; RTFormulas.Min_fund_frequency)
φ_fund_2	@CycleFundamentalPhase (RTFormulas.u_2 ; RTFormulas.i_2 ; RTFormulas.Cycle_Master ; RTFormulas.Min_fund_frequency)
φ_fund_3	@CycleFundamentalPhase (RTFormulas.u_3 ; RTFormulas.i_3 ; RTFormulas.Cycle_Master ; RTFormulas.Min_fund_frequency)
Cosφ_fund_1	@Cosine (RTFormulas.φ_fund_1)
Cosφ_fund_2	@Cosine (RTFormulas.φ_fund_2)
Cosφ_fund_3	@Cosine (RTFormulas.φ_fund_3)
Min_fundrms_frequency	10
U_fund_1	@CycleFundamentalRMS (RTFormulas.u_1 ; RTFormulas.Cycle_Master ; RTFormulas.Min_fundrms_frequency)
U_fund_2	@CycleFundamentalRMS (RTFormulas.u_2 ; RTFormulas.Cycle_Master ; RTFormulas.Min_fundrms_frequency)
U_fund_3	@CycleFundamentalRMS (RTFormulas.u_3 ; RTFormulas.Cycle_Master ; RTFormulas.Min_fundrms_frequency)
	Computing the mean (or collective) voltage
∑_U_fund	(RTFormulas.U_fund_1 + RTFormulas.U_fund_2 + RTFormulas.U_fund_3) / 3
I_fund_1	@CycleFundamentalRMS (RTFormulas.i_1 ; RTFormulas.Cycle_Master ; RTFormulas.Min_fundrms_frequency)
I_fund_2	@CycleFundamentalRMS (RTFormulas.i_2 ; RTFormulas.Cycle_Master ; RTFormulas.Min_fundrms_frequency)
I_fund_3	@CycleFundamentalRMS (RTFormulas.i_3 ; RTFormulas.Cycle_Master ; RTFormulas.Min_fundrms_frequency)
	Computing the mean (or collective) current
∑_I_fund	(RTFormulas.I_fund_1 + RTFormulas.I_fund_2 + RTFormulas.I_fund_3) / 3
	Computing the space vector signals
i_alpha	@SpaceVectorTransformation (RTFormulas.i_1 ; RTFormulas.i_2; RTFormulas.i_3;0)
i_beta	@SpaceVectorTransformation (RTFormulas.i_1 ; RTFormulas.i_2; RTFormulas.i_3;1)

Min_fund_frequency and **Min_fundrms_frequency** define the lowest fundamental frequency the calculation of φ, cosφ and the fundamental RMS values works for, and it is fixed at “10”. If the fundamental frequency of the Cycle_source falls below this value, then no results are created.

The load indicators like  (L= inductive load) in the eDrive sheet are only present if φ and cosφ calculations are enabled, as the energy flow is deducted from φ per phase.

Φ Value	Result
$0 < \varphi <= 0.5\pi$	Inductive motor
$0.5\pi < \varphi <= 1\pi$	Inductive generator
$1\pi < \varphi < 1.5\pi$	Capacitive generator
$1.5\pi < \varphi <= 2\pi$	Capacitive motor

11.4 The POWER SOURCE formulas block

The POWER SOURCE formula block is only there if a selection other than NONE is made in the configuration of the POWER SOURCE setup.

11.4.1 DC configuration

If DC is selected for the POWER SOURCE block, the content is pretty straightforward.

****

#region DC_in

Allocation of input channels to variables

List of external variables used
(variables not defined in this block)

RMS calculation for u_in and i_in

MEAN calculation for u_in and i_in

Input power P_in calculation

#endregion DC_in
****

11.4.2 RT-FDB formulas POWER SOURCE – DC configuration

Cycle_Master_in	RTFormulas.Cycle_Master
Cycle_Check_in	RTFormulas.Cycle_Check
I_in	@CycleRMS (RTFormulas.i_in ; RTFormulas.Cycle_Master_in)
U_in	@CycleRMS (RTFormulas.u_in ; RTFormulas.Cycle_Master_in)
i_in_mean	@CycleMean (RTFormulas.i_in ; RTFormulas.Cycle_Master_in)
u_in_mean	@CycleMean (RTFormulas.u_in ; RTFormulas.Cycle_Master_in)
P_in	@CycleMean (RTFormulas.u_in * RTFormulas.i_in ; RTFormulas.Cycle_Master_in)

☞ ***For all cycle based calculations, the Cycle_Master signal from the INVERTER OUTPUT block is used.***

☞ ***For a true, constant DC signal, both RMS and MEAN values would be the same. In reality, a DC signal is usually not a constant DC but superimposed with some AC ripple or noise. In this case, MEAN and RMS values might be different.***

11.4.3 AC 1ph (or pulsed DC) configuration

If 1ph is selected for the POWER SOURCE block, the content is first calculating the cycle detection, and then all the results. So it is very similar to most INVERTER OUTPUT formula blocks.

****

#region AC_1in

Allocation of input channels to variables

List of external variables used
(variables not defined in this block)

Allocation of cycle detector settings to variables

Cycle source filter, cycle master detection and cycle checking

RMS calculation
for u and i

Instantaneous power calculation

P, S, Q calculation

λ calculation

If selected in SYSTEM SETTINGS:
φ and cosφ calculation

#endregion AC_1in
****

11.4.4 RT-FDB formulas POWER SOURCE – 1ph configuration

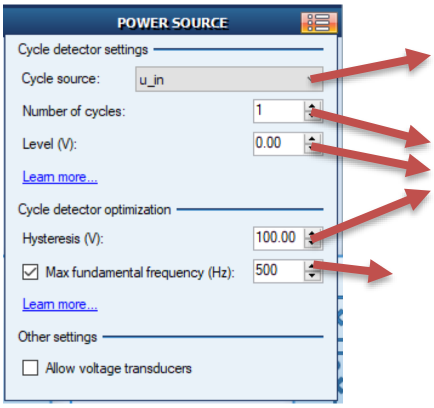
Cycle_source_in	@HWFilter (Recorder_B.u_in ; RTFormulas.Cycle_filter_type_in ; RTFormulas.Cycle_cutoff_frequency_in)
Cycle_count_in	1
Cycle_level_in	0
Cycle_hyst_in	1
Cycle_holdoff_in	0.001
Cycle_filter_type_in	1
Cycle_cutoff_frequency_in	1000
Cycle_Master_in	@CycleDetect (RTFormulas.Cycle_source_in ; RTFormulas.Cycle_count_in ; RTFormulas.Cycle_level_in ; RTFormulas.Cycle_hyst_in ; RTFormulas.Cycle_suppress_in ; 0 ; RTFormulas.Cycle_filter_type_in ; RTFormulas.Cycle_cutoff_frequency_in)
Cycle_Check_in	@CycleFrequency (RTFormulas.Cycle_Master_in)
I_in	@CycleRMS (RTFormulas.i_in ; RTFormulas.Cycle_Master_in)
U_in	@CycleRMS (RTFormulas.u_in ; RTFormulas.Cycle_Master_in)
P_in	@CycleMean (RTFormulas.u_in * RTFormulas.i_in ; RTFormulas.Cycle_Master_in)
S_in	RTFormulas.I_in * RTFormulas.U_in
Q_in	@Sqrt ((RTFormulas.S_in * RTFormulas.S_in) – (RTFormulas.P_in * RTFormulas.P_in))
λ_in	RTFormulas.P_in / RTFormulas.S_in
φ _fund_in	@CycleFundamentalPhase (RTFormulas.u_in ; RTFormulas.i_in ; RTFormulas.Cycle_Master_in ; RTFormulas.Cycle_cutoff_frequency_in)
Cos φ _fund_in	@Cosine (RTFormulas.φ_fund_in)

> ☞ ***As for the INVERTER BLOCK results, computation of φ and cosφ are optional which can be enabled in the SYSTEM SETTINGS dialog of the eDrive user interface.***

11.4.5 Cycle detector settings in the RT-FDB for 1ph configuration

The cycle detect settings are critical for proper detection of the fundamental frequency and therefore for valid results. While the user enters these parameters in the user interface, they are automatically transferred as variables into the RT-FDB and used there as input parameters for some functions.

The “mapping” from the user interface into the RT-FDB is shown below.



Cycle_source_in	@HWFilter (Recorder_B.u_in ; RTFormulas.Cycle_filter_type ; RTFormulas.Cycle_cutoff_frequency)
Cycle_count_in	1
Cycle_level_in	0
Cycle_hyst_in	1
Cycle_holdoff_in	0,001
Cycle_filter_type_in	1
Cycle_cutoff_frequency_in	1000

Cycle_source_in is the selected (and potentially filtered) source channel.
If “Max. fundamental frequency” is selected (and thus a filter is enabled), the formula reflects this.

Cycle_count_in, **Cycle_level_in** and **Cycle_hyst_in** are the parameters for the # of cycles for averaging and the hysteresis settings.

Cycle_holdoff_in and **Cycle_cutoff_frequency_in** are the parameters for the cycle source filter function.

Cycle_filter_type_in defines if the filter is on (Cycle_filter_type = 1) or off (Cycle_filter_type = 0).

11.4.6 Cycle_Master_in and Cycle_Check_in

These are exactly the same parameters as calculated for the INVERTER OUTPUT block, but for the POWER SOURCE block.
So the description of how these signals can be obtained and their meaning is described in the INVERTER OUTPUT block.
Only difference is the name, as these parameters for the POWER BLOCK have a suffix “_in” at the end, so Cycle_Master_in and Cycle_Check_in.

11.4.7 3ph: Phase to Phase configuration

The formulas for this configuration are the same as for the INVERTER OUTPUT – Phase to Phase configuration.

11.4.8 3ph: Phase to Neutral configuration

The formulas for this configuration are the same as for the INVERTER OUTPUT – Phase to Star configuration.

11.5 The MOTOR OUTPUT formulas block

The MOTOR OUTPUT formula block is only there if a selection other than NONE is made in the configuration of the MOTOR OUTPUT setup.

11.5.1 "Shaft only" and "Shaft only (with position)" configurations

These two configurations are identical except for the generation of the reset pulse for the angle. This is explained later in this chapter.

****
#region Motor_shaft_only

Variable allocation

List of external variable used

Definition of the parameters for instantaneous traces

Calculation of Cycle based torque and speed

Calculation of Timer based (instantaneous) torque and speed

Calculation of mechanical power P_mech

#endregion Motor_shaft_only
****

11.5.2 RT-FDB formulas MOTOR OUTPUT – "Shaft only" and "Shaft only (with position) configurations

M_raw	Recorder_A.M_raw
γ_mech	Recorder_A.γ_mech
Cycle_Master_mech	RTFormulas.Cycle_Master
Cycle_Check_mech	RTFormulas.Cycle_Check
Instantaneous_time	0.001
Cycle_Master_inst	@CycleInterval (RTFormulas.Instantaneous_time)
M	@CycleMean (RTFormulas.M_raw ; RTFormulas.Cycle_Master_mech)
n	@CycleRPM (RTFormulas.γ_mech ; RTFormulas.Cycle_Master_mech)
M_inst	@CycleMean (RTFormulas.M_raw ; RTFormulas.Cycle_Master_inst)
n_inst	@CycleRPM (RTFormulas.γ_mech ; RTFormulas.Cycle_Master_inst)
P_mech	2 * System.Constants.Pi * RTFormulas.n / 60 * RTFormulas.M

M_raw is an internal "raw" signal calculated by the Counter / Timer hardware used for the torque signal. It has no "meaning" to a user but needs further processing as explained later in this chapter.

γ_mech is the mechanical angle, also calculated by Counter / Timer hardware.

P_mech is the mechanical power calculated using the cycle based M and n traces and thus is also cycle based using the Cycle_Master trace of the INVERTER OUTPUT. Only by using the same cycle as used for electrical P computation ensures proper efficiency calculation.

11.5.3 *RPM calculation*

The eDrive system works with hardware setups using an angular encoder to measure rpm/speed and angle.

Technically an encoder delivers pulses per xx degrees of rotation.
These pulses are connected to a Counter / Timer input of the eDrive system.
This input uses quadrature decoding to increase the resolution and then generates the angle signal ɣ_mech based on the encoder signal.

The configuration “Shaft only (with position)” uses the reference pulse to reset the angle trace to 0°.

The configuration “Shaft only” has no reference pulse, thus the angle trace counts up to 360° and then automatically resets to 0°. The resulting angle trace ɣ_mech is therefore not referenced to any “real” mechanical or electrical angle.

> ***For “Shaft only (with position)”, the 0° of the resulting angle trace can be adjusted to match the electrical 0° using the “Angle Offset” entry of the eDrive user interface. This entry is in the MOTOR OUTPUT setup menu, and can be used to shift the ɣ_mech trace.***
>
> ***The MOTOR POLE PAIRS in the same menu can be used to convert the mechanical angle to the electrical angle. This, however, does not happen in hardware but requires user formulas in the RT-FDB (using the @MODULO function) and the pole pairs as parameters for this.***

From the angle trace ɣ_mech the rpm trace n is computed.

n	@CycleRPM (RTFormulas.ɣ_mech ; RTFormulas.Cycle_Master_mech)

This function @CycleRPM requires an angle trace as input.
The conversion is then done per cycle using the Cycle_Master signal from the INVERTER OUTPUT block.
This is needed (together with a torque signal also computed over this cycle) to compute a mechanical power P_mech over the same cycle as the electrical power P. Only then does the efficiency calculation in the ANALYSIS BLOCK give proper results when the system is operating in a dynamic way.

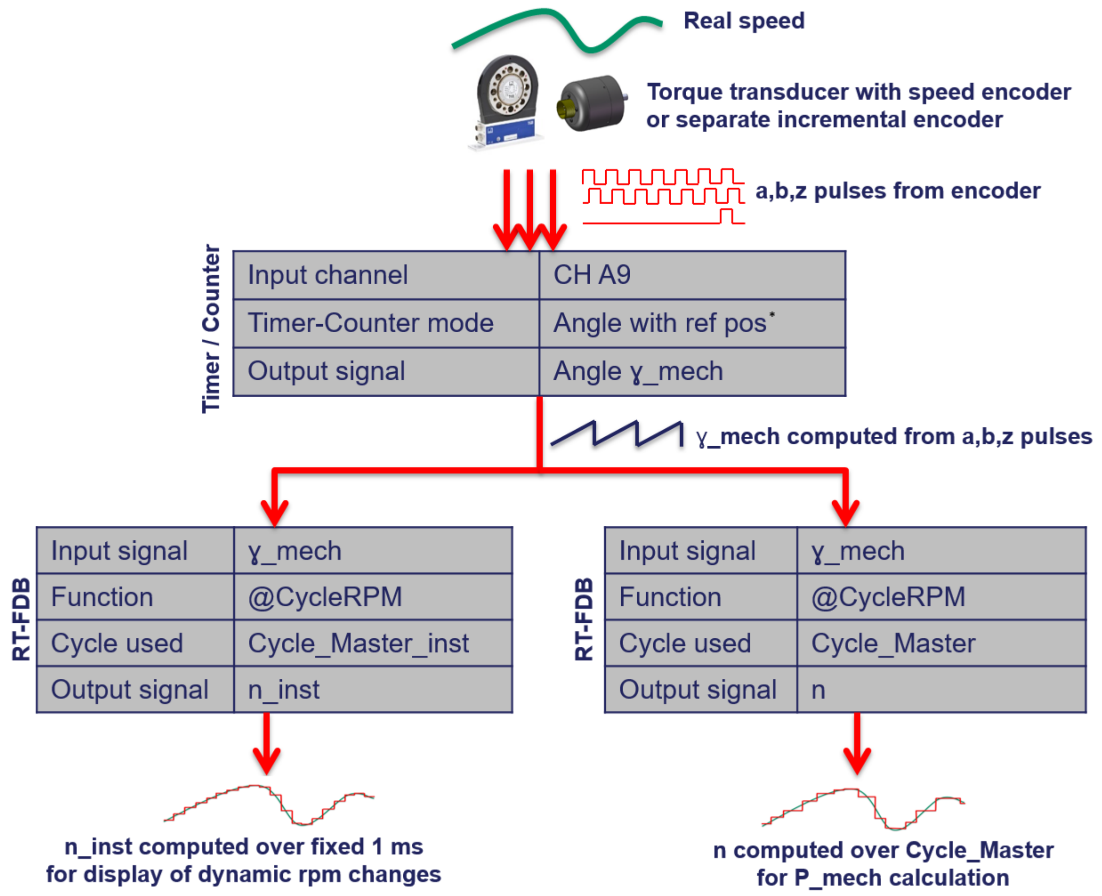
From the angle trace ɣ_mech the rpm trace n_inst is also computed.

Instantaneous_time	0.001
Cycle_Master_inst	@CycleInterval (RTFormulas.Instantaneous_time)
n_inst	@CycleRPM (RTFormulas.ɣ_mech ; RTFormulas.Cycle_Master_inst)

This is done in order to see higher frequency variations in the rpm signal (faster than the Cycle_Master interval). The n_inst calculation is done over a time constant of 1 ms.

Instantaneous_time is the time period used to generate the Cycle_Master_inst signal.
Cycle_Master_inst is a time based cycle signal used for computation of instantaneous rpm n_inst.

The whole process for angle and rpm calculation is shown below, for “Shaft only (with position)” configuration:



* Angle Quadrature with ref pos

Single frequency speed sensors are also supported. These can be created by adding a new frequency sensor of “Uni Directional” type in the Sensor Database.

11.5.4 Torque calculation – torque transducers with frequency output (HBM T12/T40B)

The eDrive system can use a torque transducer with frequency output for torque measurement. Technically this torque transducer delivers a frequency proportional to the torque applied.

This frequency signal is connected to a Counter / Timer input of the eDrive system.

Then this is converted to torque in a two-step process:

First the counter / timer input counts the frequency with a gate time of 1 µs (a time much shorter than a typical interval time of a torque transducers output frequency). The resulting trace M_raw looks a bit strange with lots of intervals without count and then an interval with a single count. This is done to maintain a short counting time, while all the information is “encoded” in the trace M_raw.

Second this trace M_raw is averaged over the Cycle_Master of the INVERTER OUTPUT block, so that the resulting trace M has the same cycle as all the electrical parameters.

> ☞ ***Visualizing the trace M_raw is of no use, as the trace itself only contains “encoded” information, so it looks “strange”, and is of no use without further processing.***
>
> ☞ ***The trace M_raw contains all the needed information for torque measurement. Decoding this information using the CycleMEAN function gives proper M and M_inst traces.***

From the raw torque trace M_raw the torque trace M is computed.

M	@CycleMEAN (RTFormulas.M_raw ; RTFormulas.Cycle_Master_mech)

This function @CycleMEAN (if used with an Counter / Timer trace as input) does a special calculation not only averaging the pulses in M_raw but also taking account the extra timing information which is encoded in the height of the pulses, giving an extremely precise result.

The averaging is using the Cycle_Master signal from the INVERTER OUTPUT section.
This is needed (together with a rpm/speed signal also over this cycle) to compute the mechanical power P_mech over the same cycle as the electrical power P. Only then does the efficiency calculation in the ANALYSIS BLOCK give proper results when the system is operating in a dynamic way

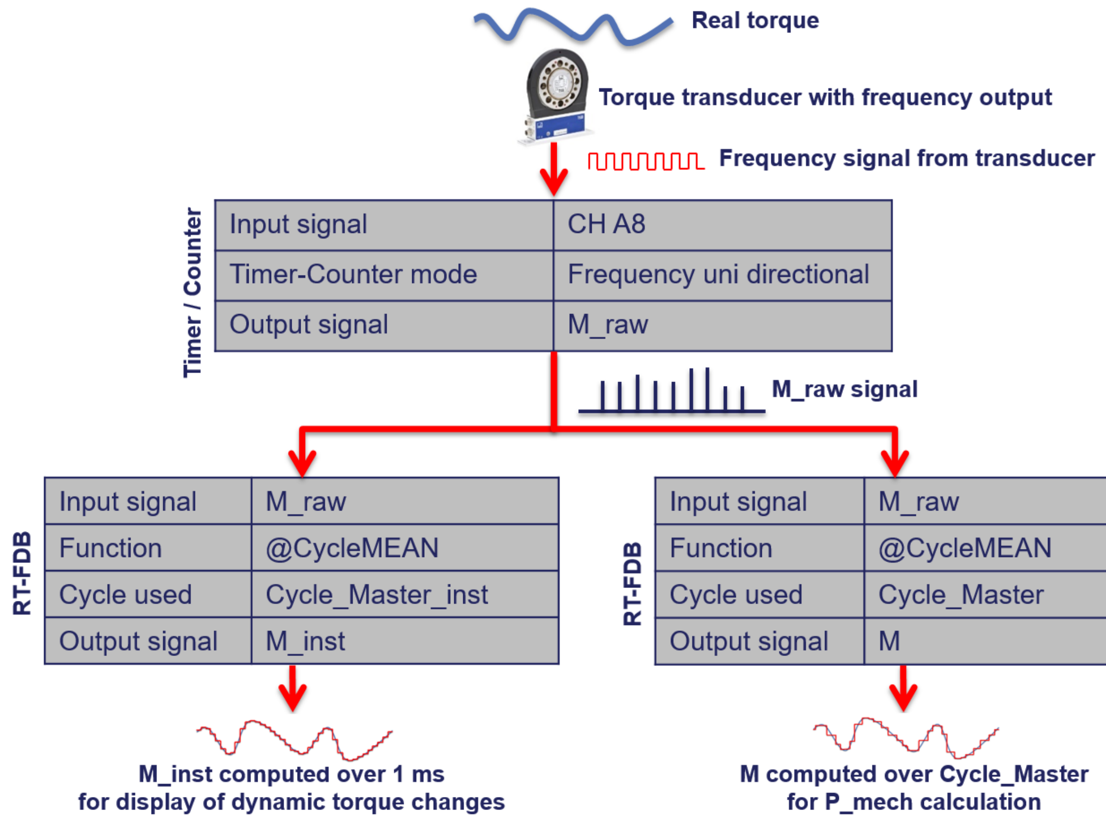
From the raw torque trace M_raw also the trace M_inst is computed.

Instantaneous_time	0.001
Cycle_Master_inst	@CycleInterval (RTFormulas.Instantaneous_time)
M_inst	@CycleMEAN (RTFormulas.M_raw ; RTFormulas.Cycle_Master_inst)

This is done in order to see higher frequency variations in the torque signal (faster than the Cycle_Master interval). The M_inst calculation is done over time constant of 1 ms.

Instantaneous_time is the time period used to generate the Cycle_Master_inst signal.
Cycle_Master_inst is a time based cycle signal used for computation of instantaneous torque M_inst.

The whole process for torque calculation is shown below:



11.5.5 Torque calculation – torque transducers with analog voltage output

The eDrive system can also use a torque transducer with voltage output to compute mechanical power P_mech.

In order to do so, the selection "Simple driveline with analog out torque transducer" needs to be enabled in the menu eDrive Setup // Select eDrive application.

After selecting this the analog channel Ch_B2 is used (in a hardware set up with 2 x GN610B boards, otherwise it might be Ch_C2) as input for the analog voltage representing torque.
The formulas remain the same, only the source of the torque signal is different.

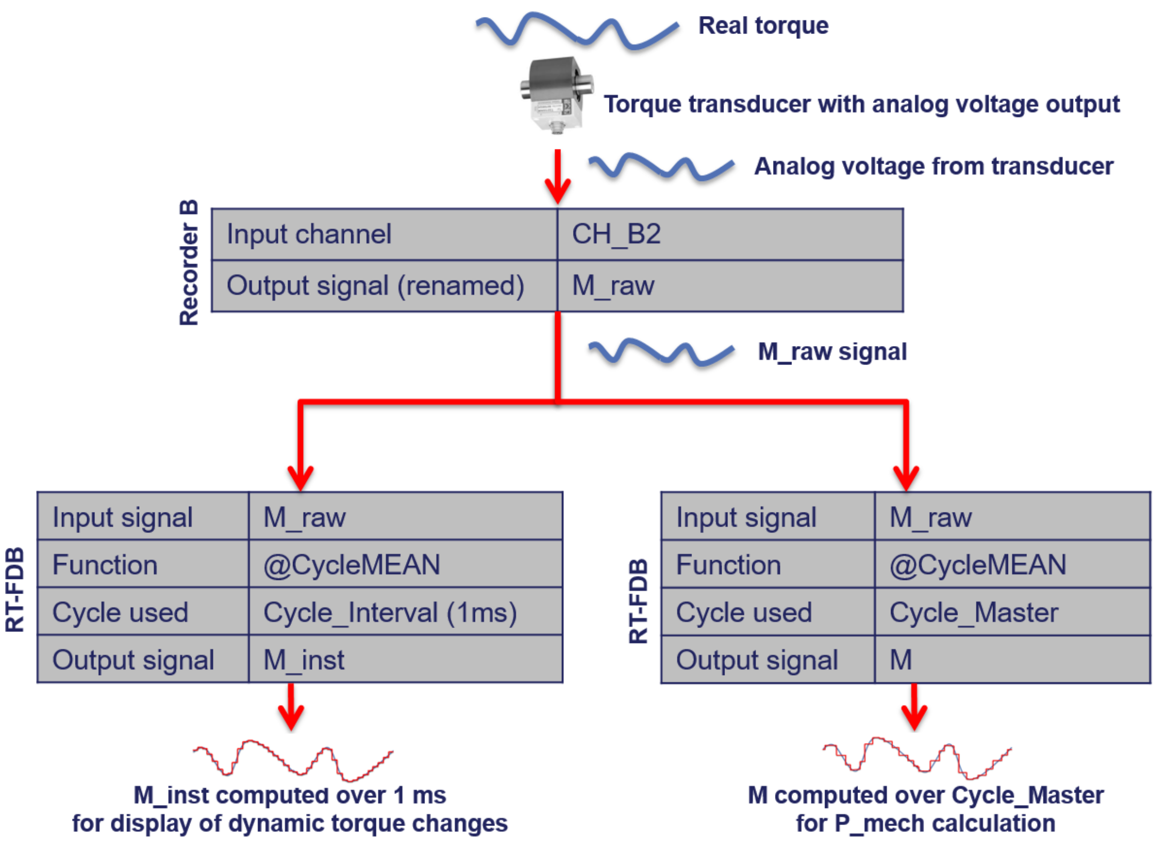
M_raw	Recorder_B.M_raw

So now Recorder B and Channel M_raw (Input channel Ch_B2) are used.
Note that M_raw is no longer the output signal of the timer / counter but an analog input channel.

The rest of the formulas remain the same.

M	@CycleMEAN (RTFormulas.M_raw ; RTFormulas.Cycle_Master_mech)
Instantaneous_time	0.001
Cycle_Master_inst	@CycleInterval (RTFormulas.Instantaneous_time)
M_inst	@CycleMEAN (RTFormulas.M_raw ; RTFormulas.Cycle_Master_inst)

The whole process for torque calculation using an analog voltage signal is shown below:



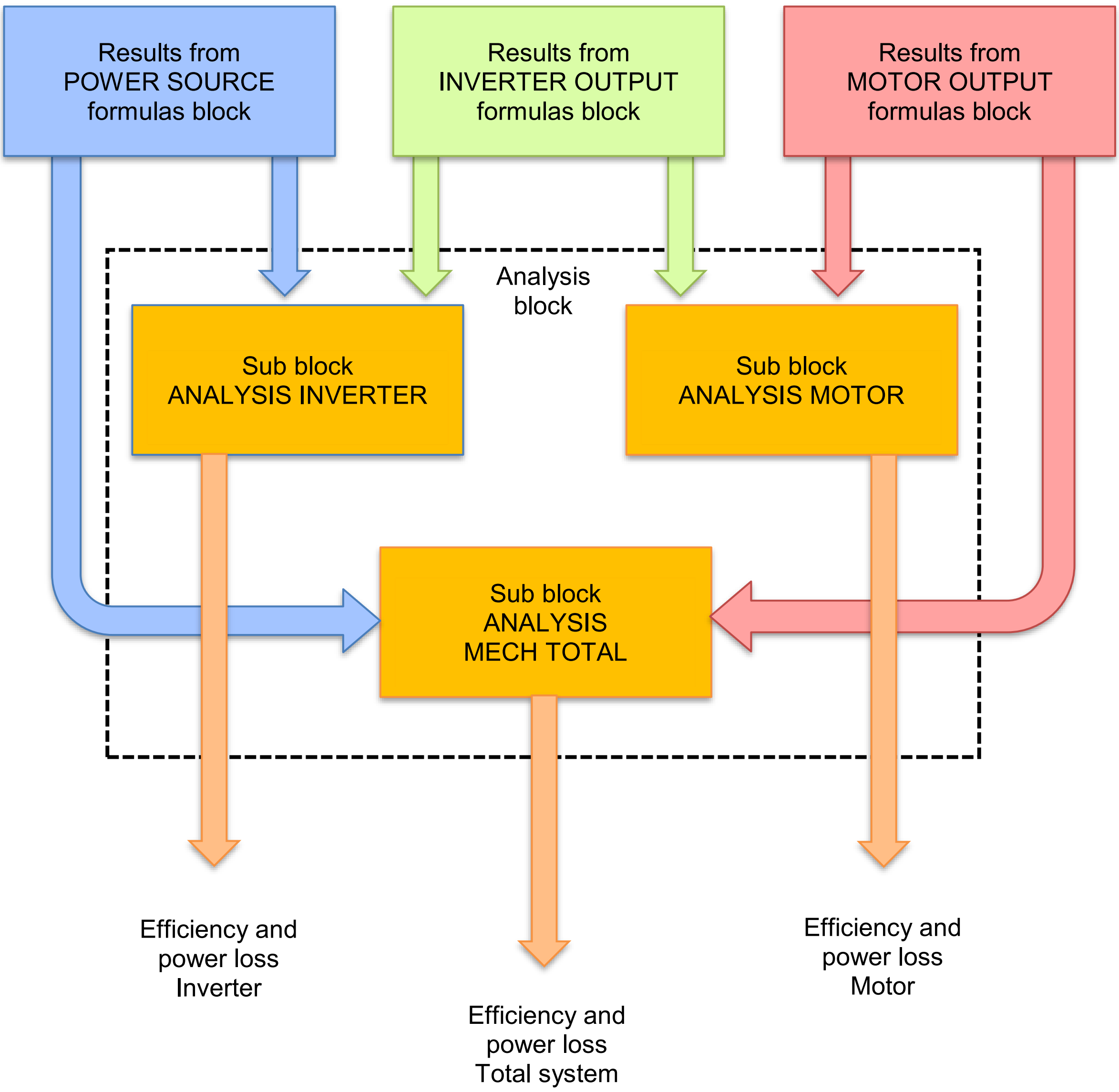
11.6 The ANALYSIS formulas block

The ANALYSIS formula block is the only one which is not directly linked to a single setting in the eDrive setup sheet or input signals. It is completely derived from the other blocks, as only performing cross block calculations.
The ANALYSIS formula block only exists only if either POWER SOURCE or MOTOR OUTPUT blocks are available. If only INVERTER OUTPUT is enabled, there is no cross block analysis to be done and the whole ANALYSIS block does not exist.

The ANALYSIS formula block is composed of up to three sub blocks:

- Analysis Inverter -> Calculations between POWER SOURCE and INVERTER OUTPUT
- Analysis Motor -> Calculations between INVERTER OUTPUT and MOTOR OUTPUT
- Analysis Mech total -> Calculations between POWER SOURCE and MOTOR OUTPUT

11.6.1 ANALYSIS block flowchart



11.6.2 *Sub block Analysis Inverter*

In this block only the inverter is analyzed. This block only exists if a POWER SOURCE block is present.

****
#region ANALYSIS INVERTER

Variable allocation

List of external variable used

INVERTER efficiency computation
in motor and generator mode

Power loss for INVERTER calculation

#endregion ANALYSIS INVERTER
****

The efficiency and the power loss in the inverter are calculated from the power results in the blocks INVERTER OUTPUT and POWER SOURCE.

η_inv_mot	(RTFormulas.P / RTFormulas.P_in) * 100
η_inv_gen	(RTFormulas.P_in / RTFormulas.P) * 100
P_loss_inv	RTFormulas.P_in - RTFormulas.P

Depending on whether the system is in motor mode or generator mode, there are different formulas for the efficiency. So the block calculates two efficiencies, one for “motor mode” and one for “generator mode”.

η_inv_mot is the inverter efficiency in motor mode, **η_inv_gen** the inverter efficiency in generator mode. One of the two is always smaller 100%, one is larger 100%.
One the one smaller than 100% is valid and also indicates the mode the system is currently in.

P_loss_inv is the power loss of the inverter. If this is positive the system is in motor mode, if it is negative (negative loss = gain) the system is in generator mode.

11.6.3 Sub block Analysis Motor

In this block only the motor is analy7ed. This block only exists if a MOTOR OUTPUT block is present.

****
#region ANALYSIS MOTOR

Variable allocation

List of external variable used

MOTOR efficiency computation
in motor and generator mode

Power loss for MOTOR calculation

#endregion ANALYSIS MOTOR
****

The efficiency and the power loss in the motor are calculated from the power results in the blocks INVERTER OUTPUT and MOTOR OUPUT.

η_mech_mot	(RTFormulas.P_mech / RTFormulas.P) * 100
η_mech_gen	(RTFormulas.P / RTFormulas.P_mech) * 100
P_loss_mech	RTFormulas.P - RTFormulas.P_mech

Depending on whether the system is in motor mode or generator mode, there are different formulas for the efficiency. So the block calculates two efficiencies, one for “motor mode” and one for “generator mode”.

η_mech_mot is the motor efficiency in motor mode, **η_mech_gen** the motor efficiency in generator mode. One of the two is always smaller 100%, one is larger 100%.
One the one smaller than 100% is valid and also indicates the mode the system is currently in.

P_loss_mech is the power loss of the motor. If this is positive the system is in motor mode, if it is negative (negative loss = gain) the system is in generator mode.

11.6.4 Sub block Analysis Mech total

In this block only the whole system – inverter and motor - is analyzed. This block only exists if both a POWER SOURCE and a MOTOR OUTPUT block are present.

****
#region ANALYSIS MECH TOTAL

Variable allocation

List of external variable used

MECH TOTAL efficiency computation
in motor and generator mode

Power loss for MECH TOTAL calculation

#endregion ANALYSIS MECH TOTAL
****

The efficiency and the power loss in the complete system are calculated from the power results in the blocks POWER SOURCE and MOTOR OUPUT.

η_total_mot	(RTFormulas.P_mech / RTFormulas.P_in) * 100
η_total_gen	(RTFormulas.P_in / RTFormulas.P_mech) * 100
P_loss_total	RTFormulas.P_in - RTFormulas.P_mech

Depending on whether the system is in motor mode or generator mode, there are different formulas for the efficiency. So the block calculates two efficiencies, one for “motor mode” and one for “generator mode”.

η_total_mot is the overall efficiency in motor mode, **η_mech_gen** the overall efficiency in generator mode. One of the two is always smaller 100%, one is larger 100%.
One the one smaller than 100% is valid and also indicates the mode the system is currently in.

P_loss_total is the power loss of the complete system, inverter and motor. If this is positive the system is in motor mode, if it is negative (negative loss = gain) the system is in generator mode.

11.7 Data types in the RT-FDB

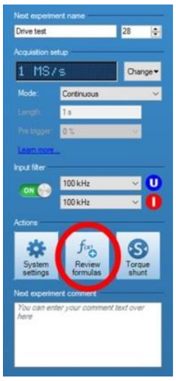
Type	Description	Example formula	Remark
SyncAnalog	Synchronous analog channel; either an input or the result of "sample math"	Recorder_A.i_1	
SyncFilteredAnalog	HW-filtered analog channel; typically the filtered Cycle_source channel, then used as input for Cycle_detect	Cycle_source_filt	• No live data • Can only be used once
Cycles	Result of the cycle detection; only timing information about cycle start and stop; not a real channel, so no further usage outside RT-FDB and eDrive.	Cycle_Master	• Cannot be reused as cycle master in post process
Async	Asynchronous analog channel; typically the result of cycle math, like "RMS"	I_1	
Scalar	Number in the RT-FDB; typically used to hand over user made entries as arguments into functions	Cycle_count	
SyncTcFreq	Counter timer channel; result of a frequency counting	M_raw	
SynchTcAngle	Counter timer channel; result of an angle computation	ɣ_mech	
SyncEventBit	Event bit channel; typically used as input channel for the timer counter channels	rpm_A_side_A_pulse	
SyncBool	Result of an boolean computation on synchronous data.		Not used in eDrive
ASyncBool	Result of an boolean computation on asynchronous data.		Not used in eDrive

12 Analyzing stored raw data

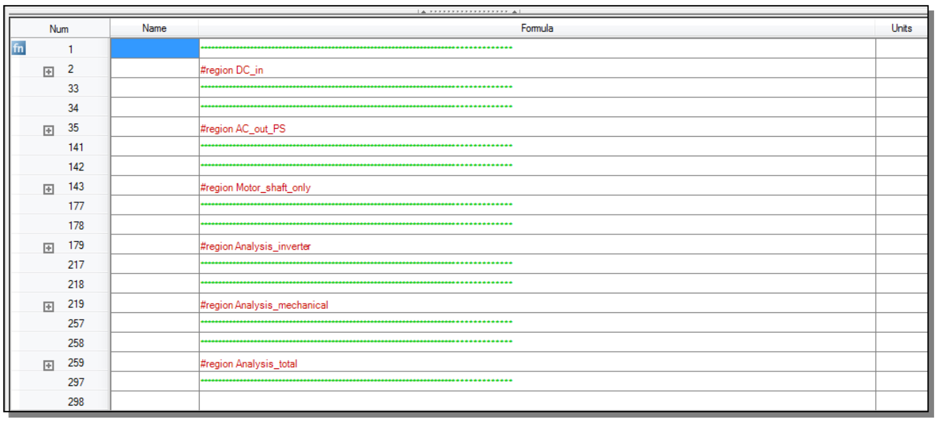
In some cases, it might be of interest to use the raw data and to re-analyze these using the Perception formula database. This could be done for special analysis, or to re-run the calculations with different (post process) filter settings or averaging cycles. Also, some “heavy” calculations like space vector transformation, dq0 transformation or airgap torque might be done only in post process.

> ***Now we start using the Formula Database in Perception.***
>
> ***This is different than the Real Time Formula Database which was computing the real time results we have seen and used so far.***
>
> ***The Real Time Formula Database “lives” in the hardware and is “transparent” to the user when using eDrive, unless he enters user formulas; it is executed in real time. The Formula Database “lives” in software and is executed post process.***

For review and analysis purposes, we can now create the needed formulas.
To do so press ***Review Formulas*** in the ***eDrive – Setup*** sheet.



A warning will pop up telling you that you might overwrite existing formulas; just click OK here.
Then the Formula sheet will show up and will be populated with formulas as shown below.



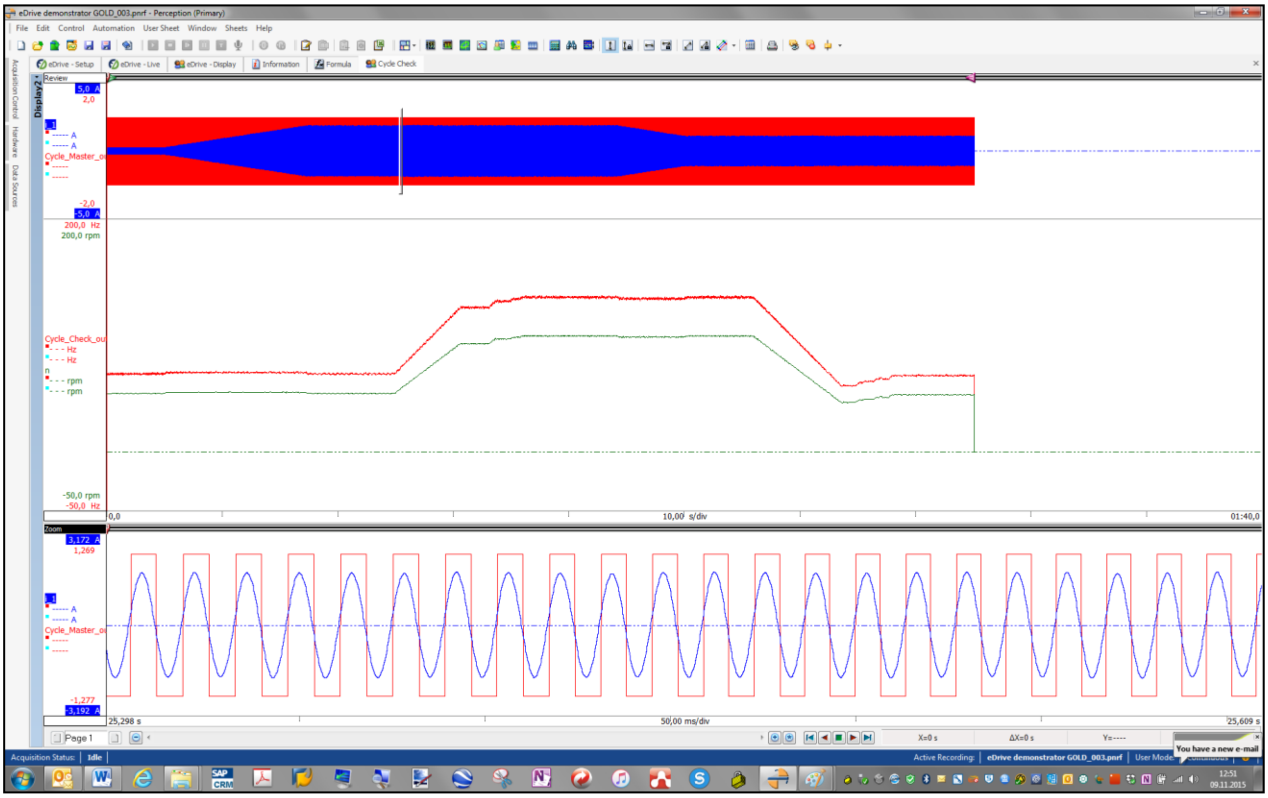
> ***The Formulas represent the current selections in eDrive – Setup. If you change this setup later on, make sure to recreate formulas by just pressing the button again.***
>
> ***User entered formulas are NOT recreated.***

12.1 Checking the cycle detection for review and analysis

There is an easy method to check if the system did detect all cycles.
In the formulas, the cycle detection function @CycleDetect detects the cycles and creates the signal **Cycle_Master_out** (for Inverter Output), showing a squarewave indicating all detected cycles.

Using this signal as input, the function @CycleFrequency is used to compute the frequency of the found cycles. The result is called **Cycle_Check_out**.

This frequency signal **Cycle_Check_out** must be of the same waveform as the trace **n**.
The difference is the number of pole pairs, and in case of an asynchronous machine, the slip.



The picture above is a user defined sheet and needs to be created by the user.

It shows from top to bottom:
* Cycle source **i_1** (blue) overlaid with **Cycle_Master_out** (red)
* **Cycle_Check_out** (red) overlaid with rpm Signal **n** (green)
* Detail zoom of cycle source **i_1** (blue) overlaid with **Cycle_Master_out** (red)

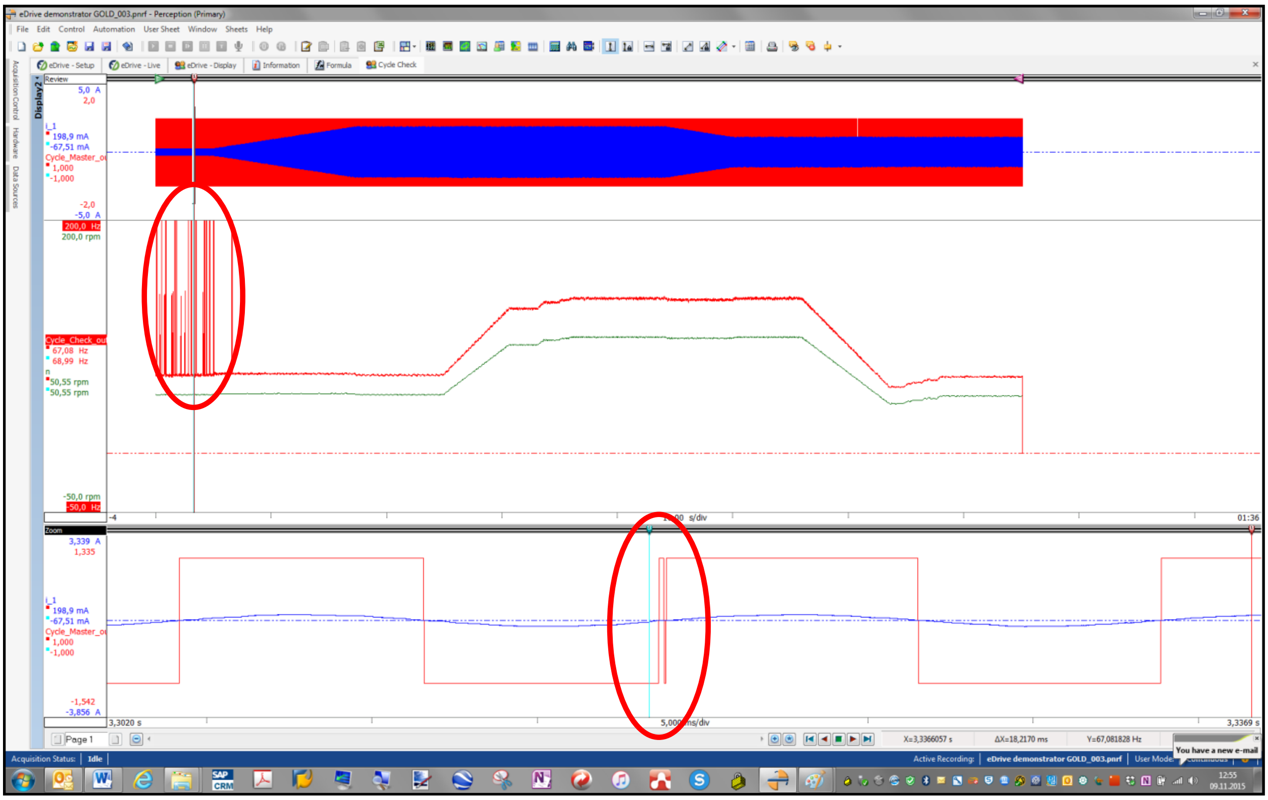
There are two possible failures of the cycle detection:

* Cycle not detected

This would lead to one or more missing cycles in the **Cycle_Master_out** signal.
The frequency signal of this, the **Cycle_Check_out,** would then halfen (or less) from one cycle to the next.
This would show a negative spike on the **Cycle_Check_out** signal.

* Wrong cycles detected

This would lead to one or more extra cycles in the **Cycle_Master_out** signal.
The frequency signal of this, the **Cycle_Check_out,** would then double (or more) from one cycle to the next.
This would show a positive spike on the **Cycle_Check_out** signal.



The picture above shows a failed cycle detection.

The positive spikes at the beginning of **Cycle_Check_out** indicate extra cycles detected.

The zoomed window shows the rising edge of **Cycle_Master_out** with the extra cycle (red circle).

Typically in eDrive such wrong cycle detections are very rare. The failure shown here was forced by disabling all Cycle optimization settings and setting the hysteresis to a value smaller than the noise.

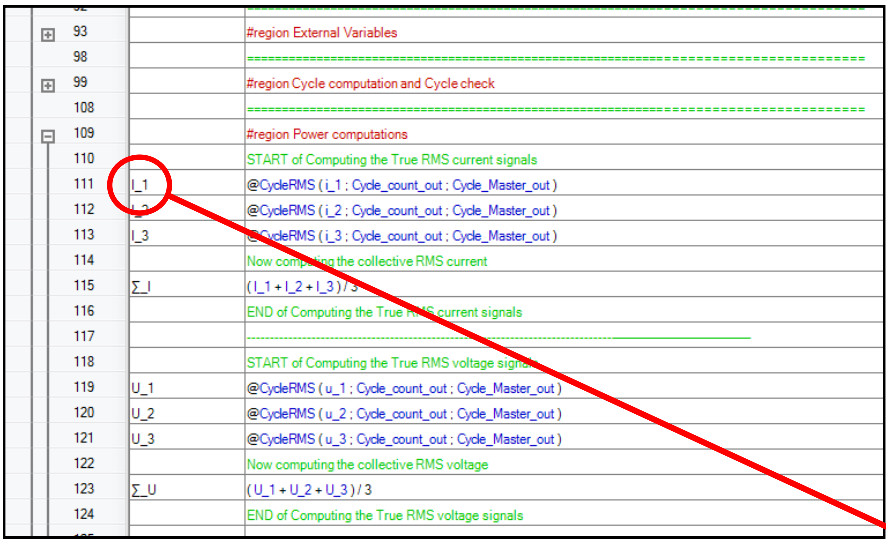
Before going into analysis, it is recommended to check the Cycle_Check_out signal.

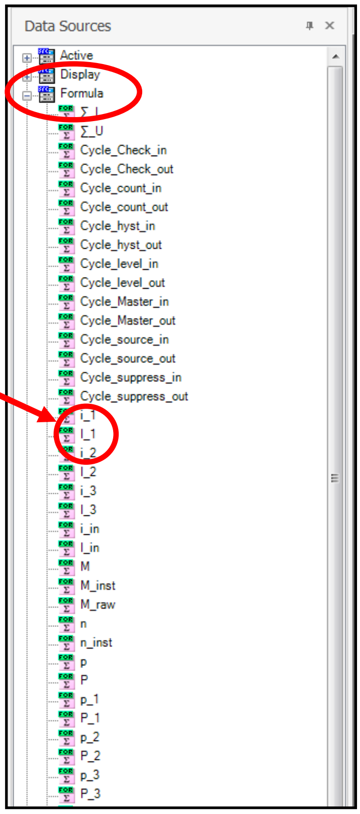
12.2 A typical user defined display for analysis

There are no displays created for post run analysis by the software.
The user can create these on his own in any way he wants.

> ***For creating user sheets and displays, please refer to the Perception software user manual.***

As the formulas are now created in the FORMULA sheet, there are new variables created like **I**_1, U_1 and so on. These variable created by formulas can be found in the DataSources, in the branch **Formula**.



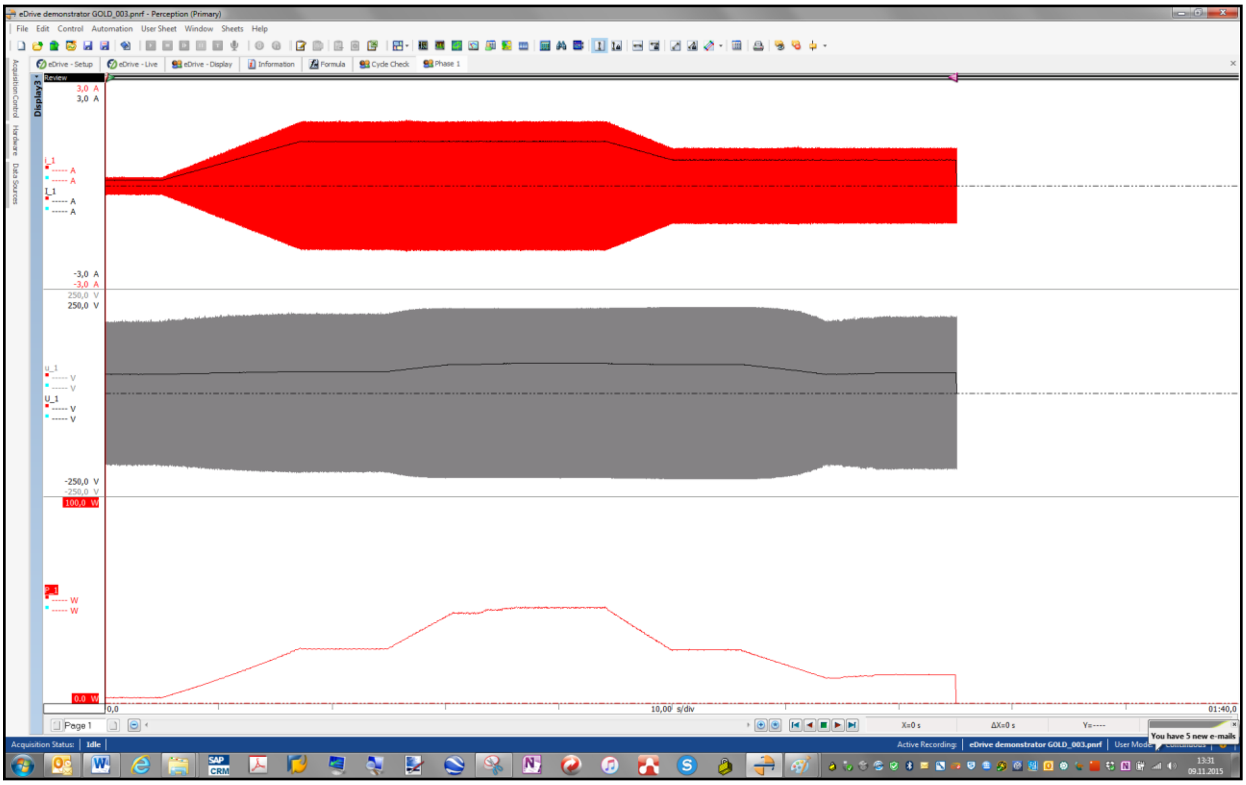


Screen above shows a portion of the **Formulas** created.

Screen to the right shows the DataSources Navigator, with the **Formula** branch expanded, and the list of computed results in this branch.

> ***The formulas as shown above for post run analysis look similar to the formulas used in the Real Time Formula Database, BUT THEY ARE NOT THE SAME. So you cannot copy and paste formulas from one formula database to the other one.***
>
> ***For more info on how to work with formulas, please refer to the Perception Formula database software manual.***
>
> ***For more info on how to work with DataSources and with user configured displays, please refer to the Perception software manual.***

After creating a new user sheet these variables can be dragged and dropped into the new sheet. A display will be created automatically by doing so. A typical analysis display might look like the one below:



The above picture shows an example review sheet of computed results, from top to bottom:

- Instantaneous current i_1 and RMS current I_1
- Instantaneous voltage u_1 and RMS voltage U_1
- True power P_1

> ***Other than the RT-FDB formulas (which are protected), all FDB formulas can be edited by the user. By doing so all results are immediately updated, so that different analysis can be performed pretty easily.***

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

A05300_02_E00_00