



# eddylab 2.0.4 manual

May 31, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Installation</b>	<b>2</b>
<b>3</b>	<b>How to Use</b>	<b>3</b>
3.1	User Interface . . . . .	3
3.2	Context Help . . . . .	3
<b>4</b>	<b>Analysis features</b>	<b>3</b>
4.1	Oscilloscope . . . . .	3
4.2	FFT . . . . .	4
4.3	Waterfall . . . . .	5
4.4	Waterfall Rpm . . . . .	5
4.5	Datalogger . . . . .	6
<b>5</b>	<b>Advanced Device Configuration</b>	<b>7</b>
5.1	Linearisation and Referencing . . . . .	7
5.1.1	Linearisation and Referencing in practice . . . . .	8
5.2	The basic Device Configuration . . . . .	10
5.3	Curve Select . . . . .	11

# 1 Introduction

eddylab is an analysis-tool for Windows used in conjunction with the TX Eddy-Current-Driver. Communication is based on the USB. The targeted area of application can be split in two parts:

- The visualisation and documentation of mechanical motion including rotary motion.
- The on-site linearisation of eddy current sensors with an active feedback system.

**Therefore the application contains five analysis features.**

1. **An Oscilloscope** - Measurements are taken in the style of a classical Oscilloscope (see section 4.1).
2. **FFT** - Spectral Analysis (2D) can be performed up to a frequency of 19000Hz (see section 4.2).
3. **Waterfall** - Spectral Analysis is expanded with a third axis - time (3D). This feature admits a more surveying impression of your mechanical spectrum (see section 4.3).
4. **Waterfall Rpm** - Spectral Analysis is expanded with a third axis - rotational speed (3D). This results in a characteristic 3D-plot of a rotating device (see section 4.4).
5. **Datalogger** - Data will be written to disc over wide ranges of time (see section 4.5).

**And three configuration features.**

1. **Linearisation** - eddy current sensors can be linearized and referenced with an incremental gauge. This admits a traceable calibration on-site (see section 5.1).
2. **The basic Device Configuration** - Basic parameters on the TX-Driver can be adjusted (edge frequency,CAN-ID...) (see section 5.2).
3. **A Curve Selector** - For multiple linearisations provided by eddylab (see section 5.3).

The TX-Driver also contains a CAN-Bus interface. A description to this interface is given in the document 'eddyCAN-Protocol'. The CAN-Bus is not related to eddylab - except for the definition of the baudrate and node-id. eddylab runs on PCs and is compatible with WindowsXP, Windows Vista and Windows 7 and further versions.

# 2 Installation

Required data can be found on the CD that came with your TX-System.

1. Connect the TX-Driver to a power-supply.
2. Connect the Device to your PC via USB: An eddylab-Device will be detected and the 'Found new Hardware Wizard' will prompt you for the Device-Driver.
3. Select the directory 'DriverD2XX' on the CD that came with your TX-System.
4. Run the 'setup.exe' in 'eddylab 2.0.x Installer' and follow the instructions

Your system now contains the new shortcut entry in the Start Menu>Programs>eddylab>eddylab 2.0.x.



Figure 1: eddylab after Startup

## 3 How to Use

### 3.1 User Interface

When loaded your screen should look like in figure 1. When connected to the USB - eddylab will automatically establish a connection to the TX-Driver. Four LEDs indicate the connection state (see figure 2). **USB** indicates a connected and powered TX-Driver. **Init** indicates initialisation. **Ready** indicates the finished initialisation. **Data** indicates an active data stream in all analysis modes. By default the startup-mode is the oscilloscope like shown in



Figure 2: Connection state

figure 1. Other modes can be accessed from the menu bar under 'Application'.

### 3.2 Context Help

A context help is given for most of the functionality. The context help can be activated in the menu bar.

## 4 Analysis features

### 4.1 Oscilloscope

Data is displayed in the style of a classical oscilloscope. Before measurements are taken consider the following aspects:

- Set the appropriate timescale (**Duration**). The range is  $14ms..5s$ .
- For sinusoidal data (in particular for small scales on a dual-channel-device) - consider to use **AC-Coupling**.
- By default the trigger-mode is set to 'auto'. In this mode data is displayed regardless whether a trigger is detected or not. For non-periodic motion (like a closing valve) it might be desirable to capture a single sequence. For this purpose set the trigger-mode to 'normal' - data will only be displayed if a trigger is detected.

The *Triggerlevel* can either be set manually or with the **Autoset Trigger**-function. Autosetting the trigger adjusts the trigger level and the vertical scale.

**Measurements** can be taken from the displayed data. For periodic motion the *frequency* and the *amplitude* can be measured. Figure 3 shows the AC-coupled signal of a triggered tuning-fork which oscillates at a frequency of 440.3Hz. The Min/Max values are -212.3 and 122 $\mu$ m.

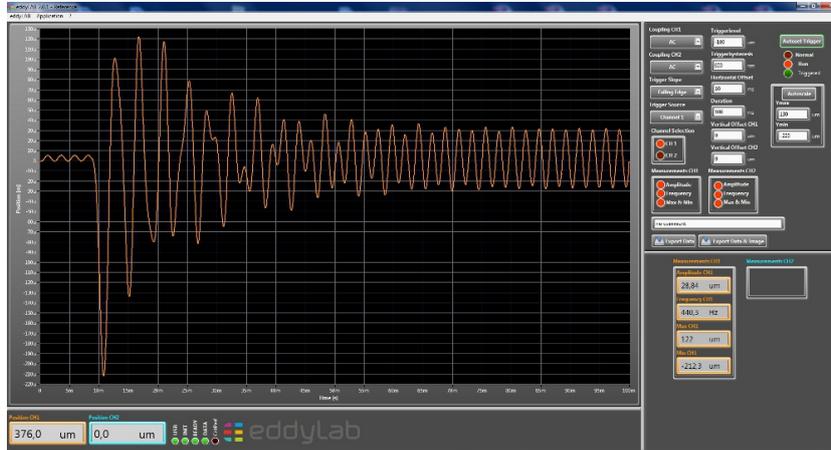


Figure 3: The AC-coupled Position of a tuning-fork.

**Documentation** of the displayed data is accomplished easily. **Export Data** writes data to 'Personal Files\ eddylab Data' in ASCII-Format. The filename is 'date\_time\_Oscilloscope\_no comment'. In order to distinguish various datasets a user-specific comment can be added instead of 'no comment'. **Export Data & Image** adds a bitmap-image of the measured data.

## 4.2 FFT

The signal's spectrum can be obtained from the FFT. The maximum scale on the frequency axis can be set with **Fmax**. Figure 4 is the FFT of a tuning-fork at a frequency of 440.2Hz with multiples at 880Hz, 1321Hz, 1761Hz and 2201Hz.



Figure 4: FFT of a tuning-fork.

**Measurements** can be taken from the spectrum. Therefore the lower threshold for frequency detection may be varied from the default value of -130dB. Depending on the actual

sensor and the cut-off frequency of the system - raising or lowering the threshold value will either exclude noise or increase the resolution for frequency-measurements.

**Documentation** is handled in the same way like in the oscilloscope-mode. *Export Data* writes data to 'Personal Files\ eddylab Data' in ASCII-Format. The filename is 'date\_time\_FFT\_no comment'. *Export Data & Image* adds a bitmap-image of the measured data.

### 4.3 Waterfall

The FFT (see section 4.2) is expanded with a time axis. The 3D-plot provides a new view to your spectrum as it can be observed over time. The third axis emphasizes small peaks above the noise floor. These small peaks can easily be missed in a 2D-plot in particular when these small peaks emerge and disappear over time. Figure 5 illustrates this effect clearly. The waterfall shows the spectrum of a tuning fork for 28 secs after triggering it. The plot obviously shows that the oscillations at  $881\text{Hz}$  and  $1320\text{Hz}$  disappear and re-emerge over time. Same as with the known FFT - the maximum scale on the frequency axis can be set with **Fmax**.

**Measurements** can be taken from the spectrum. The measurements are taken in the **analysis plane**. The analysis plane detects frequencies with their corresponding amplitudes like in the FFT. The lower threshold for frequency detection may be adjusted to fit the application.

**Documentation** of the displayed data is accomplished with **Export Image**. The filename is 'date\_time\_Waterfall\_no comment' in 'Personal Files\ eddylab Data'.

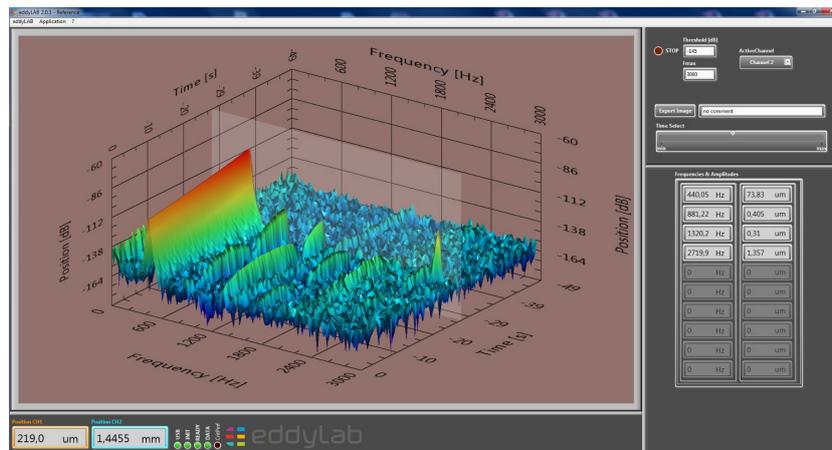


Figure 5: Waterfall of a tuning-fork.

### 4.4 Waterfall Rpm

**This function is only available in conjunction with a rotational incremental encoder. This encoder has to be connected to the TX-Driver and must be mounted on the shaft of interest.**

Analogous to the time-based waterfall - the waterfall rpm is the FFT with a third axis. This axis is the rotational speed of a rotating shaft. The correlation of rotational speed and FFT results in a characteristic 3D-plot. The plot may characterise the state of a rotating system depending on loads, oil-pressure, wear and similar aspects. This admits comparisons of the same system at different times under varying conditions. Figure 6 is the rpm-based waterfall of a shaft with a resonance frequency of 6000rpm. The amplitude peaks along the ascending and descending path of the rotational speed. Same as with the known FFT - the maximum scale on the frequency axis can be set with **Fmax**. **Rpm max** and **min** set the borders of the rpm axis.

**Measurements** can be taken from the spectrum in identical fashion to the waterfall.

**Documentation** of the displayed data is accomplished with **Export Image**. The filename is 'date\_time\_WaterfallRpm\_no comment' in 'Personal Files\ eddylab Data'.

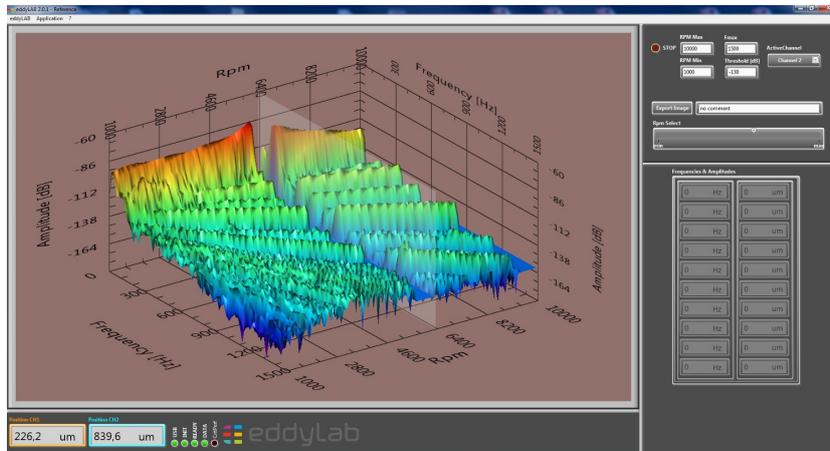


Figure 6: Rpm based waterfall of an overcritical rotating shaft.

#### 4.5 Datalogger

A datalogger can be configured in order to observe slow mechanical motion over wide ranges of time. A typical example would be temperature induced movement or a tensile test. *Start logger* writes data to disc in 'date\_time\_Log\_no comment'. The sampling rate ranges from 100ms..10s. The timescale in the display ranges from 1min..60min. Same as with the oscilloscope and the FFT an image can be exported.

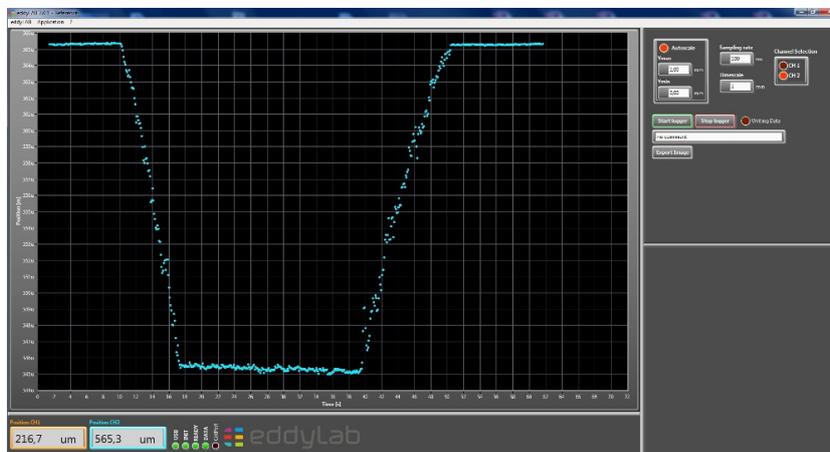


Figure 7: Slow mechanical motion displayed with the datalogger

## 5 Advanced Device Configuration

### 5.1 Linearisation and Referencing

This section puts emphasis on one of the most important features on the TX-Driver. A well known issue when it comes to eddy current measurements is the strong sensitivity to varying target material and pre-attenuation. The accuracy of eddy current sensors is a strong function of the material they are looking at. The maximum scaling error under varying material can be 20% or more. The linearity error can be 7% or more.

Another severe error source affecting the accuracy is pre-attenuation. This effect has to be taken in account when the sensor is mounted in narrow gaps and holes. The reason of this error is given with the shape of the sensing electromagnetic field. The basic idea of the sensor is to measure in one dimension with an electromagnetic field. If the shape of the field 'collides' with an object other than the target - the measured result will be incorrect. This effect is called pre-attenuation. The error due to pre-attenuation is hard to predict - but in most cases higher than expected. Figure 8 reflects the typical error behaviour on eddy current systems (red dotted line).

The TX-Driver in conjunction with eddylab resolves these issues with an integrated linearisation procedure. The backbone of the method is an interface to a linear encoder on the TX-Driver. This encoder is used as a reference signal. The reference can be used to either

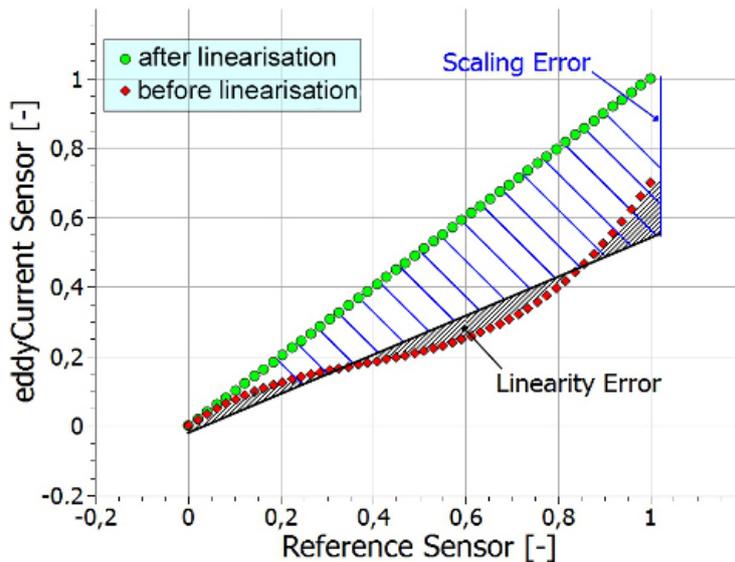


Figure 8: Illustration of typical errors on eddy current systems.

prove the accuracy of the sensor or to linearise the sensor. The reference system is directly connected to the TX-Driver and therefore admits a traceable calibration. The interface on the TX-Driver also contains a power supply. All in all - the TX-Driver is capable of referencing its own sensors on-site.

The reference system (Magnescale DK-Series) we provide has a resolution of  $100nm$  and an accuracy of  $0.5\mu m$  over  $10mm$ . The clamping diameter is  $8mm$ . Hence the DK-Series provides a system that can easily be mounted in a plentitude of applications. Furthermore we provide a small linear stage with a micrometer screw (Figure 9). The stage has a slot for the linear encoder a prism shaped socket for eddy current sensors and a quick release socket for different targets. The minimum target size is  $50x50mm$  and the maximum is  $70x70mm$ . We recommend a target thickness of  $5mm$ .



Figure 9: Portable linear stage with a linear encoder (Magnescale) - eddy current sensor and a target with the dimensions  $50 \times 50 \times 5 \text{ mm}$ .

### 5.1.1 Linearisation and Referencing in practice

The requirements to linearise or reference a sensor are as follows:

- The TX-Driver with reference input and a USB connection to a PC with a windows operating system.
- An incremental linear encoder. We recommend to use Magnescale's DK-Series.
- eddylab-Reference 2.0.x
- A setup or method that aligns the eddy current sensor with the linear encoder. **Remember to consider pre-attenuation effects.** If your sensor will be mounted in a constricted environment you need to consider this environment for the linearisation. This can be achieved in two ways. The sensor is either linearised within the desired area of application. This procedure is only possible if the target can be moved or actuated. Or if the target cannot be moved in the application (this might be the case in applications that measure on fixed shafts) it is required to linearise the sensor in a dummy version of the pre-attenuating environment.

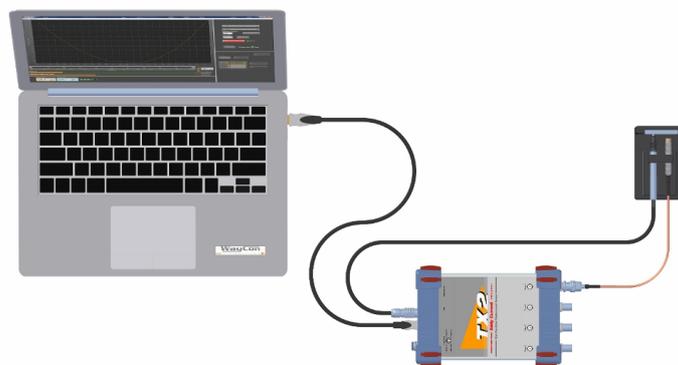


Figure 10: Setup for the Linearisation on-site.

**The linearisation module in eddylab** can be accessed from the menu bar under application. The module will only be loaded if the TX-Driver is connected to an incremental linear encoder (Magnescale's DK-Series). Furthermore - the reference input needs to be configured

for linear encoders (this is the default setting). An error message will be shown if the interface is configured for rotary encoders. The rotary/linear option can be set in configuration under application from the menu bar.

**eddylab contains a context help.** This context help is activated from the menu bar under '?'. Get information from buttons by right clicking on them.

The linearisation module is split up in two modes:

- The actual **linearisation mode**
- and the **prove mode**.

The default mode after loading all parameters from the TX-Driver and the LED **'TX System State'** switched from **'Busy'** to **'Ready'** - is the prove mode.

**The prove mode** is used to show up the absolute error over a user-defined measuring range. This feature admits a traceable calibration. First of all - the procedure to prove a sensor requires the definition of the beginning and the end of the measuring range and the desired number of positions to be proved (Buttons: Set Reference To Measuringrange Beginn; -End; Number of Positions). Based on this definition eddylab suggests a set of equidistant positions that need to be taken over with the 'Add Current Position' button. The new position will only be added if it is located within a certain range based on the generated positions. If the traffic lights indicate orange or green a position can be taken over. Once finished a chart with the absolute error is shown (see figure 11).

The maximum absolute error is  $-320\mu\text{m}$  in our example. The bottom right window also shows other error definitions like the linearity and the scaling error. The logical consequence from this result is a method to correct this error. The method is the linearisation.

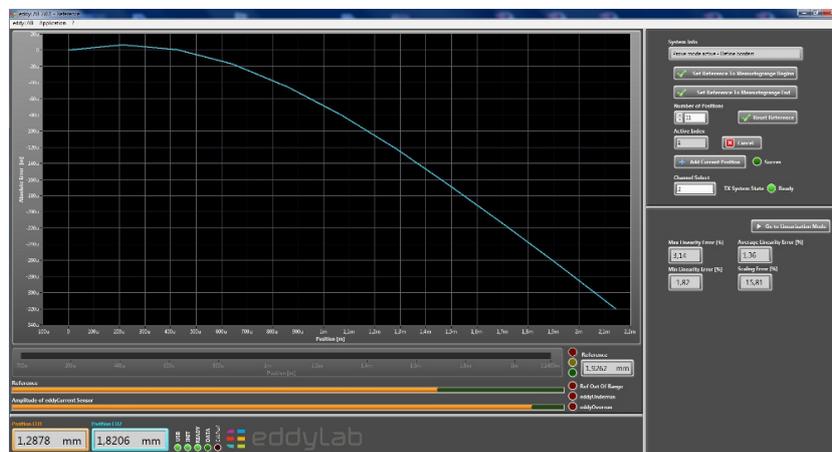


Figure 11: Error chart of an eddy current sensor measuring on a target other than calibrated.

**'Go to Linearisation Mode'** in the bottom right window switches to the linearisation mode. If the accuracy was proved as a previous step - eddylab will ask the question: *'There is a curve in memory from the previous run. Do you want to adapt the data to linearise the sensor?'* This feature facilitates the entire procedure significantly. After pressing **'Yes'** the data that is available from proving the accuracy will be used to linearise the sensor. Therefore the data will be written to the TX-Driver. The **'TX System State'** LED will indicate **'Busy'** while data is written to the TX-Driver.

Once written - the LED will indicate **'Ready'** and the table in the bottom right window will indicate a new user-defined curve in passive state with channel of origin (see figure 12). The

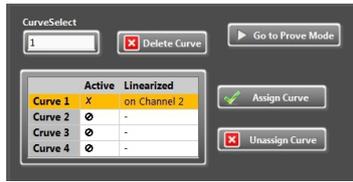


Figure 12: The table reflects the number of available curves on the TX-Driver - the origin and the active or passive state.

curve becomes valid by assigning it (button 'Assign Curve'). This initiates a reboot of the TX-Driver and the 'TX System State' will be 'Busy' again. Additionally eddylab will go on to the prove mode in order to check if the linearisation was successful. After repeating the prove procedure two error curves will be shown like in figure 13. The resulting error in our example is almost negligible.

If the accuracy was not proved as a previous step - the data for the linearisation has to be acquired. Unsurprisingly - this procedure is identical to the prove procedure. After the user-defined positions are taken over the curve is written to the TX-Driver. Activate the curve by assigning it. If a curve is needless - delete it (button **Delete Curve**). A curve can only be deleted if it is not active - therefore a curve must be unassigned before it can be deleted.

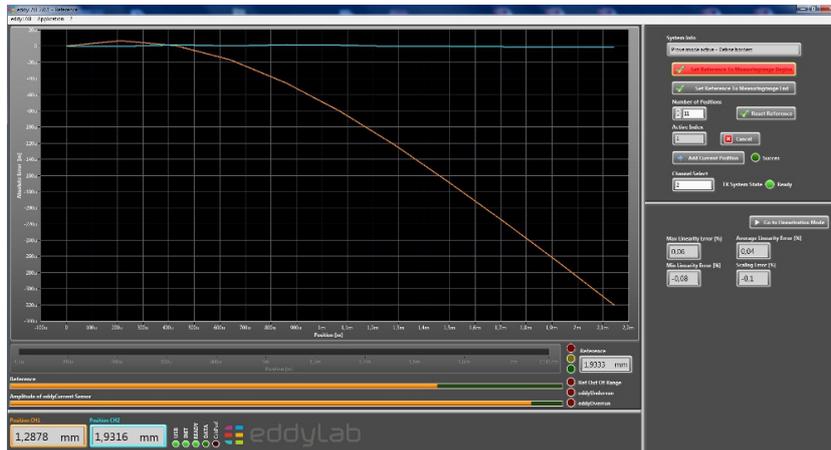


Figure 13: Error chart of an eddy current sensor before and after linearisation.

## 5.2 The basic Device Configuration

The device configuration can be accessed from the menu bar under application. The available parameters are listed with short explanations. Modify these parameters with a double click on the middle column.

**A lowpass filter:** on the TX-Driver can be defined. The selected cut-off frequency should be higher than the highest frequency in your expected spectrum. Otherwise the measured amplitudes will be attenuated. For near static or static measurements set the cut-off frequency to its lowest value - this will increase the resolution.

**The start-up mode:** can be defined. This option can be used to define the kind of operation after power-on. This is in particular interesting for the Can-Bus.

**The Can bitrate:** can be defined in eddylab. All nodes on the network need to be configured with the same bitrate. A restart is required after modifying this value to become valid.

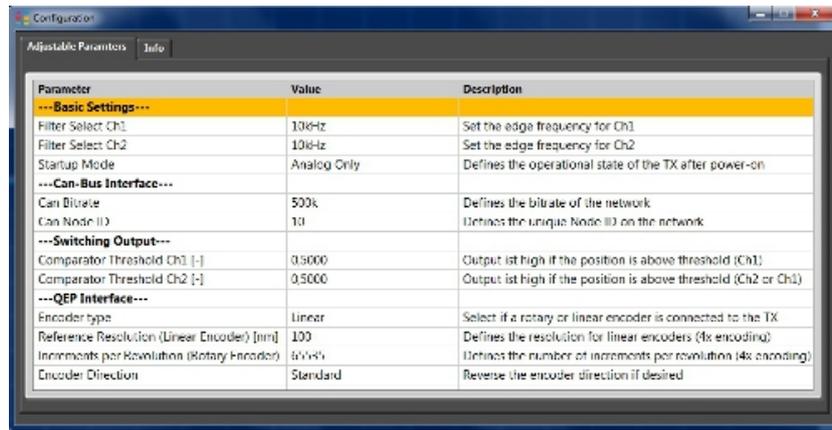


Figure 14: The configuration window.

**The Node-ID:** is the unique 7-bit-identifier on the network (value range 1..127). A restart is required after modifying this value to become valid.

**A comparator value:** can be defined for every channel. The value range is 0..1. A comparator value of 0.5 means that the output will be switched at 50% of the sensor's measuring range. If the sensor's position is above 50% the output will be high - otherwise low. For single channel devices both comparator values are assigned to one channel. This permits the definition of a low and a high threshold for one channel.

**The encoder type:** configures the encoder interface. If set to rotary encoder - the device assumes a rotary encoder. **The linear encoder needs to be set for the linearisation.**

**The Encoder Resolution:** is the resolution of a linear encoder (4x encoding) in [nm]. A typical value is 100 nm.

**Counts per Revolution:** defines the number of increments of a rotary encoder (4x encoding). The value range is 1..65535.

**The encoder direction:** can be reversed if desired.

### 5.3 Curve Select

The curve selector can be accessed from the menu bar under application. By default - the TX-Driver is delivered with one curve per channel i.e. one material per channel. Additionally we provide two more curves per channel by factory if desired. This is a useful option if the materials are known and if no verification (no on-site calibration with the DK-Series) of the overall accuracy is needed. A pre-defined curve becomes valid by assigning it (button 'Assign Curve'). This initiates a reboot of the TX-Driver and the buttons will be grayed out. After the reboot the 'Curve-Select' window can be closed and the selected curve will be active. The selected curve will also be active after power-on.

It is important to consider that the achieved overall accuracy can not be validated with this function.