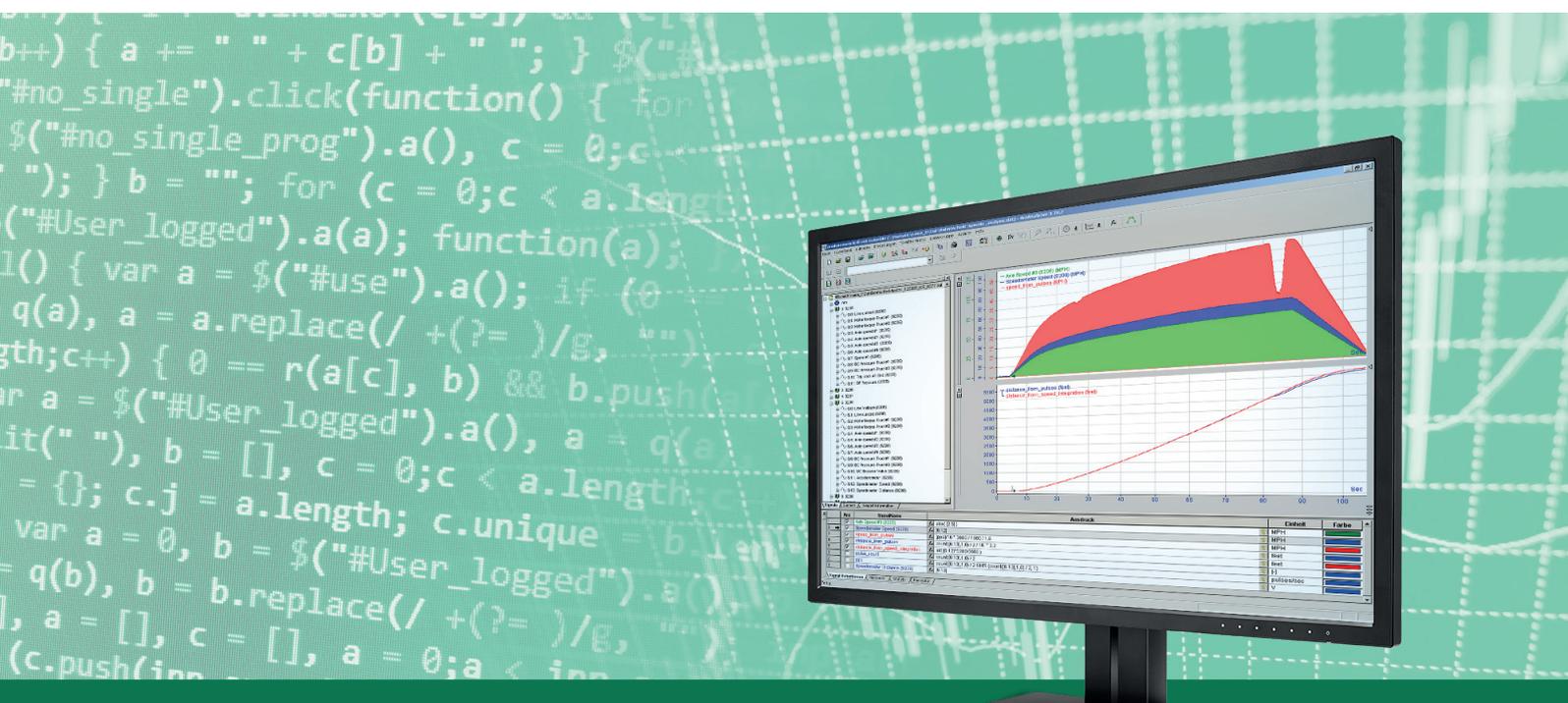




Das Ganze sehen



# ibaAnalyzer

## Редактор выражений

Руководство Часть 3  
Версия 7.0

Системы измерения для  
промышленности и электроэнергетики

---

## Производитель

iba AG  
Königswarterstr. 44  
90762 Фюрт  
Германия

## Контактные данные

Центральный офис +49 911 97282-0  
Факс +49 911 97282-33  
Служба поддержки +49 911 97282-14  
Технический отдел +49 911 97282-13  
E-Mail [iba@iba-ag.com](mailto:iba@iba-ag.com)  
Веб-страница [www.iba-ag.com](http://www.iba-ag.com)

Распространение и воспроизведение данного документа, а также использование и передача его содержания без согласия автора запрещены. Нарушение авторских прав преследуется по закону.

© iba AG 2019, все права защищены.

Содержание данной публикации было проверено на предмет соответствия описанному аппаратному и программному обеспечению. Отклонения, однако, не могут быть исключены, поэтому гарантия на полное совпадение не предоставляется. Информация, содержащаяся в данной публикации, регулярно актуализируется. Необходимые исправления содержатся в последующих изданиях или могут быть загружены из Интернета.

Актуальную версию можно всегда найти на нашей веб-странице [www.iba-ag.com](http://www.iba-ag.com)

Версия	Дата	Ревизия - глава / страница	Автор	Версия ПО
7.0	08/2 019	Редакция	TS	7.1.0

Windows® является маркой и зарегистрированной торговой маркой корпорации Microsoft. Другие упомянутые в настоящем руководстве названия продуктов и компаний могут являться зарегистрированными торговыми марками и принадлежать соответствующим лицам.

## Содержание

<b>1</b>	<b>О данном руководстве .....</b>	<b>8</b>
1.1	Целевая аудитория .....	8
1.2	Условные обозначения.....	8
1.3	Используемые символы .....	9
1.4	Структура документации.....	10
<b>2</b>	<b>Функция и управление .....</b>	<b>11</b>
2.1	Структура .....	11
2.2	Принцип работы редактора выражений .....	12
2.3	Диагностика / обнаружение синтаксических ошибок .....	13
<b>3</b>	<b>Логические функции .....</b>	<b>15</b>
3.1	Функции сравнения .....	15
3.2	Булевы функции .....	15
3.3	Булевы функции (побитовые) .....	16
3.4	Ветвление .....	17
3.4.1	If (Если).....	17
3.4.2	Switch .....	18
3.5	Распознавание фронтов .....	19
3.5.1	OneShot.....	19
3.5.2	SetReset.....	19
3.6	Функции таймера (IEC 61131-3) .....	20
3.7	IsData / Coalesce .....	21
<b>4</b>	<b>Математические функции .....</b>	<b>22</b>
4.1	Основные арифметические операции .....	22
4.1.1	Основные арифметические операции +, -, *, /.....	22
4.1.2	Abs.....	22
4.1.3	Mod .....	22
4.1.4	Ceiling / Floor / Round (ОКРВВЕРХ / ОКРВНИЗ / ОКРУГЛ) .....	23
4.2	Интегральное и дифференциальное исчисление .....	24
4.2.1	Int .....	24
4.2.2	Diff / Dif .....	24

4.3	Степени и корни.....	25
4.3.1	Pow.....	25
4.3.2	Sqrt.....	25
4.4	Функция e и логарифмы.....	26
4.4.1	Exp.....	26
4.4.2	Log.....	26
4.4.3	Log10.....	26
4.5	Пи.....	26
4.6	Sum.....	27
4.7	Тригонометрические функции.....	28
<b>5</b>	<b>Статистические функции.....</b>	<b>29</b>
5.1	Среднее значение (Avg).....	29
5.2	Maxima (Max).....	31
5.3	Minima (Min).....	33
5.4	Стандартное отклонение (StdDev).....	35
5.5	Процентиль (Percentile).....	36
5.6	Корреляция и ковариация (Correl, CoVar).....	39
5.7	Куртозис.....	41
5.8	Асимметрия (Skewness).....	44
<b>6</b>	<b>Счет и сортировка.....</b>	<b>46</b>
6.1	Count.....	46
6.2	CountSamples.....	47
6.3	Sort.....	48
<b>7</b>	<b>Функции времени/длины.....</b>	<b>49</b>
7.1	Преобразование и передискретизация.....	49
7.1.1	ConvertBase.....	49
7.1.2	Resample (Neuabtasten).....	49
7.1.3	SampleAndHold.....	50
7.1.4	SampleOnce.....	50
7.2	Время.....	52
7.2.1	Время.....	52
7.2.2	AbsoluteTime.....	52

7.3	Пересчет с базы времени на базу длины .....	53
<b>8</b>	<b>Действия с осью X.....</b>	<b>55</b>
8.1	Смещение вдоль оси X .....	55
8.2	XCutRange / XCutValid.....	56
8.3	XMarkRange / XMarkValid.....	57
8.4	XMirror / XStretch / XStretchScale .....	59
8.5	XFirst / XLast / XNow .....	62
8.6	XSize / XSumValid .....	63
8.7	XValues / YValues.....	63
8.8	VarDelay.....	64
8.9	XY.....	64
8.10	XMarker1 / XMarker2 .....	65
8.11	XBase / XOffset .....	66
8.12	FillGaps .....	66
8.13	XAlignFft .....	67
<b>9</b>	<b>Операции с вектором.....</b>	<b>69</b>
9.1	GetFirstIndex / GetLastIndex .....	69
9.2	GetRows.....	69
9.3	GetZoneCenters.....	70
9.4	GetZoneOffset .....	70
9.5	GetZoneWidths.....	70
9.6	MakeVector .....	70
9.7	SetZoneWidths .....	71
9.8	VectorAvg.....	71
9.9	VectorKurtosis.....	72
9.10	VectorMarkRange .....	72
9.11	VectorMin / VectorMax .....	72
9.12	VectorPercentile .....	73
9.13	VectorSkewness.....	73
9.14	VectorStdDev .....	73
9.15	VectorSum .....	74
9.16	VectorToSignal / SignalToVector .....	74

9.17	Traverse / TraverseW .....	75
9.18	VectorPolynomial / VectorLSQPolyCoef.....	76
<b>10</b>	<b>Текстовые функции .....</b>	<b>77</b>
10.1	InfofieldText / ChannelInfoFieldText / ModuleInfoFieldText.....	77
10.2	TextCompare / CompareText .....	78
10.3	ToText / FromText .....	80
10.4	TrimText .....	82
10.5	ConcatText .....	82
10.6	CharValue .....	82
10.7	CountText / TextLength.....	82
10.8	DeleteText / InsertText / ReplaceText .....	83
10.9	MidText / FindText .....	84
<b>11</b>	<b>Различные функции .....</b>	<b>85</b>
11.1	Debounce .....	85
11.2	Envelope .....	86
11.3	False / True .....	87
11.4	GetBit / GetBitMask.....	88
11.5	HighPrecision .....	89
11.6	InfoField / ChannelInfoField / ModuleInfoField.....	90
11.7	LimitAlarm .....	92
11.8	ManY.....	93
11.9	Rand (СЛЧИС).....	94
11.10	Sign .....	94
11.11	Технострока .....	95
11.12	WindowAlarm.....	95
11.13	YatX / SetYatX.....	96
11.14	PulseFreq.....	97
<b>12</b>	<b>Функции фильтра .....</b>	<b>98</b>
12.1	LP .....	98
12.2	PreWhiten.....	98
<b>13</b>	<b>Технологические функции .....</b>	<b>99</b>

13.1	ChebyCoef .....	99
13.2	CubicSpline .....	99
13.3	LSQPolyCoef.....	100
13.4	Полиномиальная .....	102
13.5	LSQExponentialCoef .....	102
13.6	Экспоненциальная.....	102
<b>14</b>	<b>Спектральный анализ (операции БПФ) .....</b>	<b>103</b>
14.1	FftInTimeAmpl / FftInTimePower .....	103
14.2	FftOrderAnalysisAmpl / FftOrderAnalysisPower .....	104
14.3	FftPeaksInTimeAmpl / FftPeaksInTimePower .....	106
14.4	FftAmpl / FftPower .....	107
14.5	FftComplex .....	108
14.6	FftReal / FftRealInverse .....	109
14.7	AWeighting / DbScale .....	109
14.8	IntSpectrum .....	110
<b>15</b>	<b>Электрические функции.....</b>	<b>111</b>
15.1	СКЗ / Eff .....	111
15.2	Функции для сети "треугольник» .....	112
15.3	Функции для сети "звезда» .....	115
15.4	Гармонические функции .....	117
15.4.1	TIF .....	119
<b>16</b>	<b>Техподдержка и контакты.....</b>	<b>121</b>

# 1 О данном руководстве

Данное руководство описывает функцию и применение программного обеспечения *ibaAnalyzer*.

## 1.1 Целевая аудитория

Данное руководство предназначено для квалифицированных специалистов по работе с электрическими и электронными модулями, которые обладают необходимыми знаниями в области коммуникационных и измерительных технологий. Такими специалистами считаются лица, которые на основании своей профессиональной подготовки, специальных знаний и опыта, а также знаний соответствующих предписаний могут оценить возможные последствия и риски.

Данная документация предназначена для лиц, занимающихся анализом и обработкой измеренных и технологических данных. В связи с тем, что данные предоставляются в сочетании с другими iba-продуктами, для работы с *ibaAnalyzer* требуются или будут полезны следующие знания:

- Операционная система Windows
- ibaPDA (появление и структура измеренных данных)

## 1.2 Условные обозначения

В данном руководстве используются следующие условные обозначения:

Действие	Условное обозначение
Команды меню	Меню <i>Функциональная схема</i>
Вызов команды меню	"Шаг 1 - шаг 2 - шаг 3 - шаг x" Пример: Выбрать меню <i>Функциональная схема - Добавить - Новый функциональный блок</i>
Клавиши клавиатуры	<Название клавиши> Пример: <Alt>; <F1>
Одновременное нажатие клавиш	<Название клавиши> + <Название клавиши> Пример: <Alt> + <Strg>
Графические клавиши (кнопки)	<Название клавиши> Пример: <ОК>; <Отмена>
Имя файла, путь	"Имя файла", "Путь" Пример: „Test.doc“

## 1.3 Используемые символы

В данной документации используются символы техники безопасности, которые имеют следующее значение:

---

### Опасно!



**Несоблюдение данного предписания по технике безопасности грозит летальным исходом или тяжкими телесными повреждениями!**

- Соблюдайте технику безопасности

---

### Внимание!



**Несоблюдение данного предписания по технике безопасности может привести к летальному исходу или тяжким телесным повреждениям!**

- Соблюдайте технику безопасности.

---

### Осторожно!



**Несоблюдение данного предписания по технике безопасности может привести к травмам или причинить материальный ущерб!**

- Соблюдайте технику безопасности.

---

### Важно



Особые указания, например, исключения из правил и т.д.

---

### Совет



Советы, наглядные примеры и маленькие хитрости, позволяющие облегчить работу.

---

### Дополнительная документация



Ссылка на дополнительную документацию или специальную литературу.

## 1.4 Структура документации

В данной документации представлено обширное описание программного обеспечения *ibaAnalyzer*. Оно служит руководством пользователя для настройки ПО, а также как справочный документ.

Дополнительно к данной документации для получения последней информации об установленной версии программы Вы можете обратиться к журналу версий в главном меню *Журнал версий* (Файл *versions.htm*). В данном файле наряду с перечисленными устраненными программными ошибками есть краткая информация о расширениях и улучшениях ПО.

Кроме того, при каждом обновлении ПО, которое содержит существенно новые свойства, выходит специальная документация «NewFeatures...», которая предлагает подробное описание новых функций.

Версия программного обеспечения, к которой относится соответствующий текст данной документации, приведен в таблице ревизий на странице 2. Документация к *ibaAnalyzer* (PDF и печатное издание) поделена на четыре отдельные части. Каждая часть имеет свою собственную нумерацию глав и страниц и актуализируется независимо.

Часть	Заголовок	Содержание
Часть 1	Введение и установка	Общие указания, лицензии и аддоны Установка и запуск программы Пользовательский интерфейс
Часть 2	Работа с <i>ibaAnalyzer</i>	Работы с файлом измерений и анализ, функции изображения, макроконфигурации, редактор фильтров, настройки, печать, экспорт, интерфейс к <i>ibaHD-Server</i> , <i>ibaCapture</i> и <i>Reportgenerator</i> (генератор отчетов).
Часть 3	Редактор выражений	Каталог всех функций вычисления в редакторе выражений, вкл. объяснение.
Часть 4	Примеры применения	<i>* в подготовке</i>

## 2 Функция и управление

Редактор выражений является вспомогательным средством для ввода (математических) формул или выражений, которые подробно описаны в следующих разделах. Как правило, данные выражения могут быть вручную введены в строки таблицы сигналов, во вкладке «Определения сигналов».

Чтобы облегчить данный ввод и предоставить подробный список возможных операций и их синтаксиса, есть редактор выражений, который доступен в каждой строке, в которой может быть введен сигнал.

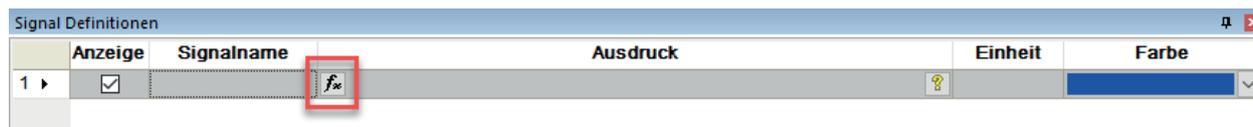


Рис. 1: Символ для запуска редактора выражений.

### Примечание



Символьная кнопка  на панели инструментов открывает не редактор выражений, а диалоговое окно для логических определений сигналов, см. *Логические определения сигналов* в части 2 руководства.

### 2.1 Структура

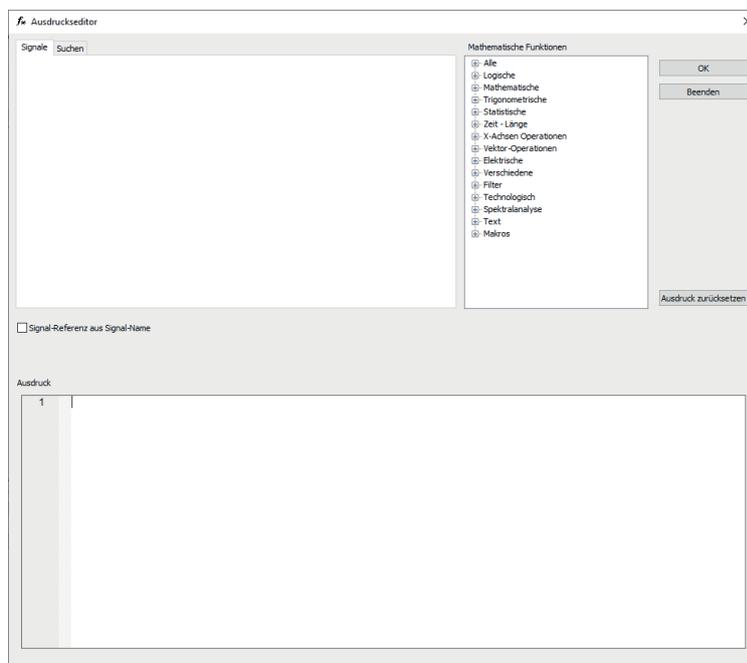


Рис. 2: Редактор выражений

Редактор выражений состоит из трех разделов.

В левой части находится дерево сигналов аналогичное дереву сигналов в окне дерева сигналов. В отличие от окна дерева сигналов здесь отображаются наряду с исходными сигналами все выражения, которые уже были созданы при помощи редактора выражений. Из данного дерева сигналов выбирают желаемые сигналы или выражения, с которыми должно выполняться вычисление.

В правой части диалогового окна в изображении дерева функций находится подборка доступных математических операций и других функций, сгруппированных по темам.

Под этими двумя полями выбора находится ввод команды, в которую вносится желаемое выражение в несколько строк. Вверху на сером участке появляется краткое указание по синтаксису операции, если она выделена в дереве функций, в т.ч. всплывающая подсказка, если функция выделена в вводе команды.

Кнопка <Сбросить выражение> удаляет все записи из командной строки.

Поле выбора «Референс сигнала из имени сигнала» Вы можете активировать, если Вы хотите использовать в выражениях имена сигналов вместо обычных обозначений сигналов, состоящие из [номер модуля:номер сигнала].

---

#### Примечание



При использовании имен сигналов в качестве референса нужно удостовериться, что имена сигналов уникальны.

---

## 2.2 Принцип работы редактора выражений

Редактор выражений позволяет применять в командную строку операции и операнды, сигналы и выражения двойным щелчком мыши или при помощи Drag & Drop. Данный порядок действий рекомендуется во избежание опечаток и ускорения работы.

Основное правило: В месте, в котором стоит курсор в строке ввода, вставляется операция или операнд, по которым в дереве функций или в дереве сигналов выполняется двойной щелчок.

Чтобы не потерять обзорность в комплексных выражениях, при помощи комбинации клавиш <Strg>+<B> можно «перепрыгивать» между связанными парами скобок.

---

#### Примечание



Функция для применения сигналов и выражений двойным щелчком по командной строке доступна в редакторе выражений и не может использоваться в обычном дереве сигналов в окне дерева сигналов.

---

Опытным пользователям как в таблице сигналов, так и в командной строке редактора выражений доступна функция *Intellisense* (помощь при вводе). При ручном вводе автоматически открывается окно с возможными дополнениями Вашего ввода. Оно включает как

функции и их параметры, так и сигналы или виртуальные выражения, содержащиеся в файле измерений.

При помощи стрелок Вы можете выбрать предложенный вариант в окне Intellisense и применить при помощи клавиши ввода. Если Вы продолжите писать, будут предлагаться соответствующие варианты дополнения команды. Если данная функция будет использоваться в редакторе выражений, все нужные скобки будут добавляться автоматически.

## 2.3 Диагностика / обнаружение синтаксических ошибок

Если закрыть редактор выражений нажатием на ОК, то в соответствующей строке определения сигнала появится только что созданное выражение.



Рис. 3: Диагностика редактора выражений

В качестве имени сигнала автоматически вносится само выражение, его можно изменить вручную на любой текст. В комплексных выражениях, имеющих каскадную взаимосвязь, рекомендуется выбирать по-возможности короткие и однозначные обозначения, чтобы сохранять обзорность выражения.

В случае ошибочного ввода при помощи редактора выражений появляется предупреждение, которое позволяет скорректировать выражение. При щелчке на <Да> курсор автоматически переходит в место предполагаемой ошибки.

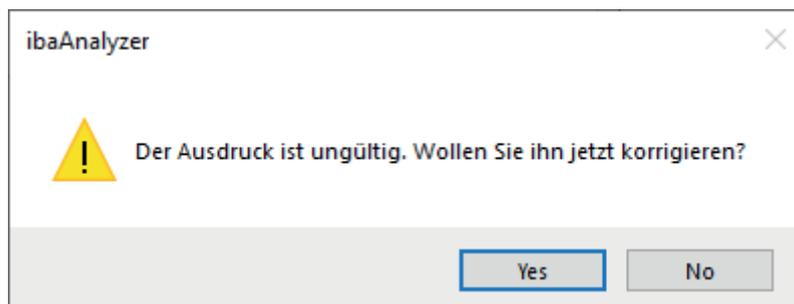


Рис. 4: Редактор выражения, диагностика, указание на ошибку

### Совет



Функция поиска возможных ошибок может быть запущена также вручную при помощи комбинаций клавиш <Strg>+<E>.

Если сообщение об ошибке игнорируется, или ошибка допускается при вводе в строке определения сигналов, *ibaAnalyzer* сигнализирует об этом красным цветом.



Рис. 5: Диагностика редактора выражения, идентификация ошибка

Таким образом могут быть обнаружены формальные или синтаксические ошибки, с которыми вычисление невозможно. Для получения более точной информации о причине

ошибки, можно открыть диагностику щелчком мыши по желтому символу знака вопроса в соответствующей строке определения сигнала.

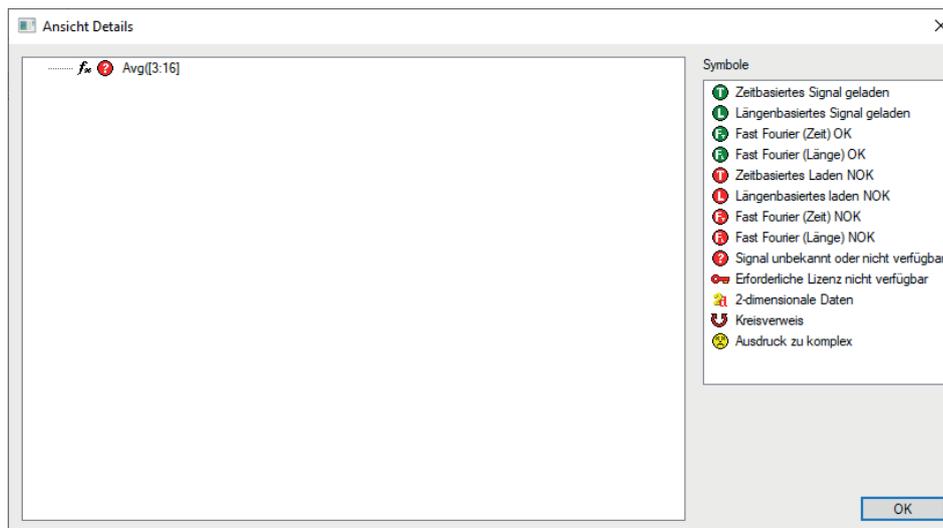


Рис. 6: Диагностика редактора выражений, окно диагностики, ошибка

## 3 Логические функции

### 3.1 Функции сравнения

например, ('Expression1') < ('Expression2')

>	больше
>=	больше/равно
<	меньше
<=	меньше/равно
<>	неравно
=	равно

Table 1: Функции сравнения

#### Описание

Операции сравнения >, >=, <, <=, <> и = позволяют сравнивать значения двух выражений (операндов) друг с другом. Операции возвращают как результат соответственно булево значение ИСТИНА или ЛОЖЬ. В качестве операндов могут быть внесены оригинальные сигналы, расчетные выражения или просто постоянные значения. Результат может изображаться и обрабатываться в качестве нового выражения как сигнал. Таким образом легко образуются новые сигналы, которые в свою очередь могут снова использоваться как условия для других функций.

#### Примечание



Если точка пересечения двух кривых находится между двумя точками измерения, то результат сравнения последних двух измерений сохраняется до следующей точки измерения. Т.е. любые изменения с ИСТИНЫ на ЛОЖЬ (или наоборот) всегда вводятся во вкладку точек измерений. Соединительная линия между двумя точками измерения при изображении аналоговых значений является только графическим приближением.

### 3.2 Булевы функции

например, ('Expression1') И ('Expression2')

AND	Логическое И
OR	Логическое ИЛИ
XOR	Логическое исключаящее ИЛИ
NOT	Логическое НЕ, отрицание

Table 2: Булевы функции

## Описание

При помощи булевых функций И, ИЛИ, НЕ и ИСКЛЮЧАЮЩЕЕ ИЛИ могут быть связаны друг с другом бинарные выражения, например, цифровые сигналы. В соответствии с правилами булевых функций, функции возвращают как результат соответственно значение ИСТИНА и ЛОЖЬ. Цифровые сигналы, вычисляемые (бинарные) выражения и числовые значения 0 или 1 могут вводиться как параметры.

Результат может изображаться и обрабатываться в качестве нового выражения как сигнал. Таким образом легко образуются новые сигналы, которые в свою очередь могут снова использоваться как условия для других функций.

A	B	A AND B (И)	A OR B (ИЛИ)	A XOR B (исключающее И)	NOT A (НЕ)
0	0	0	0	0	1
1	0	0	1	1	0
0	1	0	1	1	
1	1	1	1	0	

Table 3: Логические функции, таблица истинности

## 3.3 Булевы функции (побитовые)

например, ('Expression1') bw\_NOT ('Expression2')

bw_AND	Побитовое И
bw_OR	Побитовое ИЛИ
bw_XOR	Побитовое исключаяющее ИЛИ
bw_NOT	Побитовое НЕ

Table 4: Булевы функции (побитовые)

### Описание

Данные функции служат для побитовой связи двух аналоговых значений на базе булевой алгебры. Функции возвращают как результат сигнал типа 32Bit-Integer. В качестве аргументов ожидаются значения типа 32Bit-Integer.

Если аргументы не являются целочисленными значениями, то десятичная часть отсекается, прежде чем операция будет выполнена. Если аргументы слишком большие, что их абсолютное значение не подходит под тип 32Bit-Integer, операция выполняется только на 32 младших битах.

При соединении двух аналоговых значений с функцией bw логически связываются отдельные биты обоих значений. Результатом снова является однотипное аналоговое значение с битовой маской в соответствии с логической связью.

## Пример

Для 2 аналоговых значений  $W1 = 15$  и  $W2 = 2$  получаются следующие результаты:

	Десятичное значение	Биты	Шестнадцатеричные	Значение результата
Выходное значение W1	15	...1111	0x0000000F	
Выходное значение W2	2	...0010	0x00000002	
W1 bw_AND W2		...0010	0x00000002	2
W1 bw_OR W2		...1111	0x0000000F	15
W1 bw_XOR W2		...1101	0x0000000D	13
bw_NOT (W1)		...0000	0xFFFFFFFF	-16

Table 5: Таблица истинности для побитовой связи

## 3.4 Ветвление

### 3.4.1 If (Если)

```
If('Condition', 'IF-True', 'IF-False')
```

#### Аргументы

'Condition'	Условие в виде действия с булевыми результатами ИСТИНА или ЛОЖЬ
'IF-True'	Действие выполняется, если 'Condition' ИСТИНА
'IF-False'	Действие выполняется, если 'Condition' ЛОЖЬ

#### Описание

Функция If (ЕСЛИ) служит для условного выполнения дополнительных вычислений. В зависимости от булева результата условия ('Condition'), которое само может быть действием, при результате ИСТИНА выполняется действие 'IF-True', при результате ЛОЖЬ соответственно действие 'IF-False'.

Таким образом можно проводить различные вычисления с управлением процессом. Функция может использоваться для иерархического построения с реализацией дополнительных разветвлений.

#### Совет



Если для 'Condition' введено только одно значение, то в качестве условия осуществляется запрос, больше (ИСТИНА) или меньше (ЛОЖЬ) ли значение, чем 0,5.

### 3.4.2 Switch

```
Switch('Selector_Expression', 'Case_1_Expression', 'Value_1_Expression',
      'Case_2_Expression', 'Value_2_Expression',
      ...
      'Case_n_Expression', 'Value_n_Expression',
      'Default_Value_Expr')
```

#### Аргументы

'Selector_Expression'	Выражение, которое проверяется на различные условия
'Case_n_Expression'	Выражение, которое сравнивается при помощи 'Selector_Expression'
'Value_n_Expression'	Результат, если 'Selector_Expression' и 'Case_n_Expression' совпадают
'Default_Value_Expr'	Результат, если ни один из 'Case_n_Expression' не совпадает с 'Selector_Expression'

#### Описание

Данное указание сравнивает входящее 'Selector\_Expression' с любым множеством 'Case\_n\_Expression', опираясь на оператор SQL CASE. Требуется как минимум 3 аргумента. При четном количестве аргументов последний интерпретируется как 'Default\_Value\_Expr', который привлекается, если ни один из 'Case\_n\_Expression' не подходит к 'Selector\_Expression'.

Если 'Selector\_Expression' и 'Case\_n\_Expression' подходят друг к другу, возвращается соответствующее 'Value\_n\_Expression'. Если несколько 'Case\_n\_Expression' подходят к входному сигналу, то выбирается автоматически первый.

В качестве 'Selector\_Expression' допустимы следующие сигналы:

- Числовая постоянная
- Текстовая постоянная
- Равноудаленно или неравноудаленно собранный канал
- Текстовый канал

Как правило, типы сравниваемых значений подходят друг к другу, в противном случае соответствующий случай не выбирается.

## 3.5 Распознавание фронтов

### 3.5.1 OneShot

OneShot('Expression')

#### Описание

Данная функция возвращает результат ИСТИНА, если текущее измеренное значение 'Expression' не равно предыдущему. Она возвращает результат ЛОЖЬ, если текущее измеренное значение равно предыдущему.

#### Совет



Функция работает также с неравноудаленными измеренными значениями.

### 3.5.2 SetReset

SetReset('Set', 'Reset', 'SetDominant=1')

#### Аргументы

'Set'	Передний фронт устанавливает функцию на ИСТИНА	
'Reset'	Передний фронт устанавливает функцию на ЛОЖЬ	
'SetDominant'	Опциональный параметр (по умолчанию = 1), который управляет тем, какой входной аргумент является доминантным, если оба аргумента получают одновременно положительный фронт.	
	'SetDominant' = 1	Set имеет преимущество по отношению к Reset
	'SetDominant' = 0	Reset имеет преимущество по отношению к Set

#### Описание

Данная функция используется для управления цифровым результатом (ИСТИНА/ЛОЖЬ) при помощи передних фронтов (переход с 0 на 1) аргументов 'Set' и 'Reset'.

Передний фронт операнда 'Set' возвращает статическое ИСТИНА как результат. Передний фронт операнда 'Reset' сбрасывает результат на ЛОЖЬ. Аргумент 'SetDominant' является опциональным и определяет приоритетность 'Set' или 'Reset'.

---

**Совет**

При аналоговом сигнале превышение значения 0,5 соответствует переднему фронту.

---

## 3.6 Функции таймера (IEC 61131-3)

**TOF**

`TOF('in', 'pt')`

**Описание**

Задержка выключения. Выключение выхода осуществляется через 'pt' секунд после выключения входа 'in'.

**TON**

`TON('in', 'pt')`

**Описание**

Задержка включения. Включение выхода осуществляется 'pt' секунд после включения входа 'in'.

**TP**

`TP('in', 'pt')`

**Описание**

Функция импульса. Выход включается на 'PT' секунд после переднего фронта во входе 'IN'.

---

**Совет**

Следующий передний фронт во время выходного импульса не расширит выходной импульс и не перезапустит импульс.

---

## 3.7 IsData / Coalesce

### IsData

```
IsData('Expression', 'End')
```

#### Аргументы

'Expression'	Входной сигнал
'End'	Длина выходного сигнала

#### Описание

Результат данной операции ИСТИНА, если для 'Expression' доступны измеренные значения. Результат ЛОЖЬ, если отсутствуют измеренные значения или сигналы пустые. Данная функция может, например, использоваться как условие для других вычислений.

В качестве опции может быть указан параметр 'End'. При помощи данного параметра сигнал результата функции может быть сокращен или удлинен таким образом, что он будет совпадать с другими сигналами и может использоваться для дополнительных связей. Если Вы не указываете параметр 'End', то длина сигнала результата соответствует длине входного сигнала (вкл. недопустимые семплы).

### Coalesce

```
Coalesce('Candidate1', 'Candidate2', ...)
```

#### Описание

По соответствующему запросу SQL функция *Coalesce* возвращает первый из аргументов, который содержит данные.

Это может использоваться, например, чтобы создать защиту от отсутствующих сигналов в dat-файле.

## 4 Математические функции

### 4.1 Основные арифметические операции

#### 4.1.1 Основные арифметические операции +, -, \*, /

например, ('Expression1') + ('Expression2')

##### Описание

Все сигналы и выражения могут обрабатываться при помощи основных арифметических действий (сложение, вычитание, умножение и деление). Если в качестве операндов используются цифровые сигналы или выражения с основными арифметическими действиями, то *ibaAnalyzer* переводит значения ИСТИНА как 1.0 и ЛОЖЬ как 0.0. Результатом основного арифметического действия является аналоговое выражение.

#### 4.1.2 Abs

Abs('Expression')

##### Описание

Функция абсолютного значения возвращает абсолютное значение (= |значение|) аргумента 'Expression'.

---

##### Совет



Интерполированные значения при смене знака между двумя точками измерения могут отличаться по величине.

---

#### 4.1.3 Mod

Mod('Divident', 'Divisor')

##### Описание

Данная функция возвращает как результат модуль 'Divident' и 'Divisor'. Функция использует С-функцию fmod, которая позволяет использовать значения с плавающей запятой для 'Divident' (делимое) и 'Divisor' (делитель).

r по модулю - это остаток от деления делимое / делитель, где обратное равенство:

делимое = делитель \* x + r, где x - целое число (Integer).

r по модулю имеет всегда один и тот же знак, что и 'Divident' (делимое), а абсолютное значение r всегда меньше абсолютного значения 'Divisor' (делитель).

Если 'Divident' < 'Divisor', то функция возвращает значение 'Divident' как результат. Нахождение остатка на математическом языке называется «делением по модулю».

#### 4.1.4 Ceiling / Floor / Round (ОКРВВЕРХ / ОКРВНИЗ / ОКРУГЛ)

##### Ceiling (ОКРВВЕРХ)

`Ceiling('Expression')`

##### Описание

Данная функция возвращает наименьшее целочисленное значение, которое больше или равно 'Expression'.

##### Floor (ОКРВНИЗ)

`Floor('Expression')`

##### Описание

Данная функция возвращает наибольшее целочисленное значение, которое меньше или равно 'Expression'.

##### Round (ОКРУГЛ)

`Round('Expression')`

##### Описание

Данная функция округляет 'Expression' до следующего целого числа (целочисленного значения) в большую или меньшую сторону.

## 4.2 Интегральное и дифференциальное исчисление

### 4.2.1 Int

```
Int('Expression', 'Reset')
```

#### Аргументы

'Expression'	Измеренное значение	
'Reset'	Оptionальный цифровой параметр, который может использоваться для сброса интеграла или подавления процесса интегрирования. 'Reset' может сам быть выражением.	
	'Reset' > 0	Интеграл сбрасывается.
	'Reset'= 0	Активируется интегрирование (уставка)

#### Описание

Данная функция возвращает как результат интеграл выражения 'Expression'. Параметр 'Reset' может использоваться для сброса интеграла или подавления процесса интегрирования, например, для многократного интегрирования одного и того же сигнала при периодических или реверсивных процессах. 'Reset' может сам быть выражением.

### 4.2.2 Diff / Dif

```
Diff('Expression', 'dy'=0)
```

#### Описание

Данная функция возвращает как результат производную (или дифференциал) выражения 'Expression'. Если установить параметр 'dy' на Истина(), то вместо дифференциала вычисляется только разница между измеренными значениями.

#### Пример

Если 'Expression' является сигналом измерения длины, то при помощи функции *Diff* из него может быть рассчитан график скорости.

## 4.3 Степени и корни

### 4.3.1 Pow

```
Pow('Expression1', 'Expression2')
```

#### Аргументы

'Expression1'	Базовое число
'Expression2'	Показатель степени

#### Описание

Данная функция возводит 'Expression1' (базовое число) в степень 'Expression2' (показатель степени):

#### Пример

Вычисление некоторых важных степеней

$$(2)^0 = \text{Pow}(2, 0) = 1$$

$$(2)^{-2} = \text{Pow}(2, -2) = 0,25$$

$$(-2)^2 = \text{Pow}(-2, 2) = 4$$

$$(10)^{(\lg 2)} = \text{Pow}(10, \lg 2) = 2$$

$$(0)^{-1} = \text{Pow}(0, -1) = +\infty \text{ (бесконечность)}$$

#### Совет



Возведение 0 в степень -1 не выдает сообщение об ошибке, но не выдает результат.

### 4.3.2 Sqrt

```
Sqrt('Expression')
```

#### Описание

Данная функция возвращает как результат квадратный корень 'Expression'.

#### Примечание



Отрицательные значения для 'Expression' не выводят сообщение об ошибке, но и не дают результат.

## 4.4 Функция e и логарифмы

### 4.4.1 Exp

Exp('Expression')

#### Описание

Данная функция вычисляет выражение  $(e)^{\text{'Expression'}}$

### 4.4.2 Log

Log('Expression')

#### Описание

Данная функция возвращает как результат натуральный логарифм выражения 'Expression'.

---

#### Примечание



Отрицательные значения для 'Expression' не выводят сообщение об ошибке, но и не дают результат.

### 4.4.3 Log10

Log10('Expression')

#### Описание

Данная функция возвращает как результат десятичный логарифм выражения 'Expression'.

---

#### Примечание



Отрицательные значения для 'Expression' не выводят сообщение об ошибке, но и не дают результат.

## 4.5 Пи

Pi()

#### Описание

Число Пи - это математическая константа ( $\pi = 3.1415927\dots$ ) для различных вычислений в системе. При помощи данной функции Вы можете  $\pi$  добавить число в Ваши вычисления.

## 4.6 Sum

Sum('Expression', 'Reset'=0)

### Описание

Данная операция суммирует все точки сигналов функции. Если образование суммы прерывается значением сброса, образование суммы начинается повторно.

### Пример

Образование суммы начинается с точки сигнала 10 + 9 + 8 + ...6. Посредством 'Reset' = ИСТИНА осуществляется прерывание, и функция сбрасывается на нуль. Затем образование суммы начинается повторно.

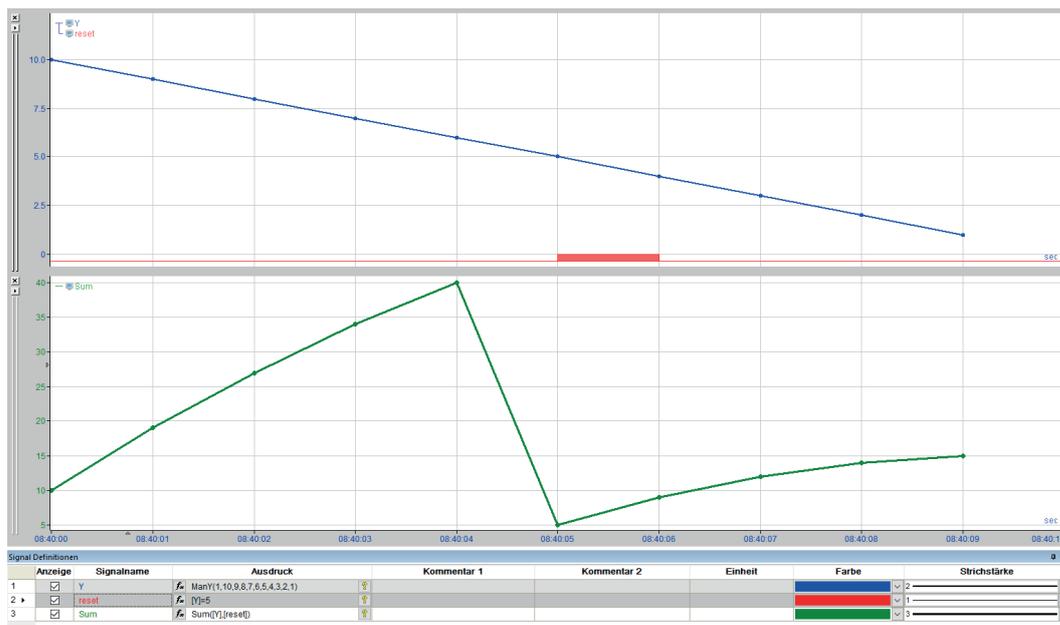


Рис. 7: Математические функции, сумма

## 4.7 Тригонометрические функции

например,  $\text{Sin}(\text{'Expression'})$

$\text{Sin}(\text{'Expression'})$	Данная функция возвращает как результат синус 'Expression' в радианах.
$\text{Cos}(\text{'Expression'})$	Данная функция возвращает как результат косинус 'Expression' в радианах.
$\text{Tan}(\text{'Expression'})$	Данная функция возвращает как результат тангенс 'Expression' в радианах.
$\text{Asin}(\text{'Expression'})$	Данная функция возвращает как результат арксинус 'Expression' в радианах.
$\text{Acos}(\text{'Expression'})$	Данная функция возвращает как результат арккосинус 'Expression' в радианах.
$\text{Atan}(\text{'Expression'})$	Данная функция возвращает как результат арктангенс 'Expression' в радианах.
$\text{Atan2}(\text{'X'}, \text{'Y'})$	Данная функция возвращает как результат арктангенс 'Y'/'X'.

Table 6: Тригонометрические функции

### Описание

Для самых разных вычислений, в которых требуются тригонометрические функции, например, расчет мощности в системах переменного тока, доступны стандартные функции и соответствующие обратные функции.

### Пример

Изображение тригонометрических функций:

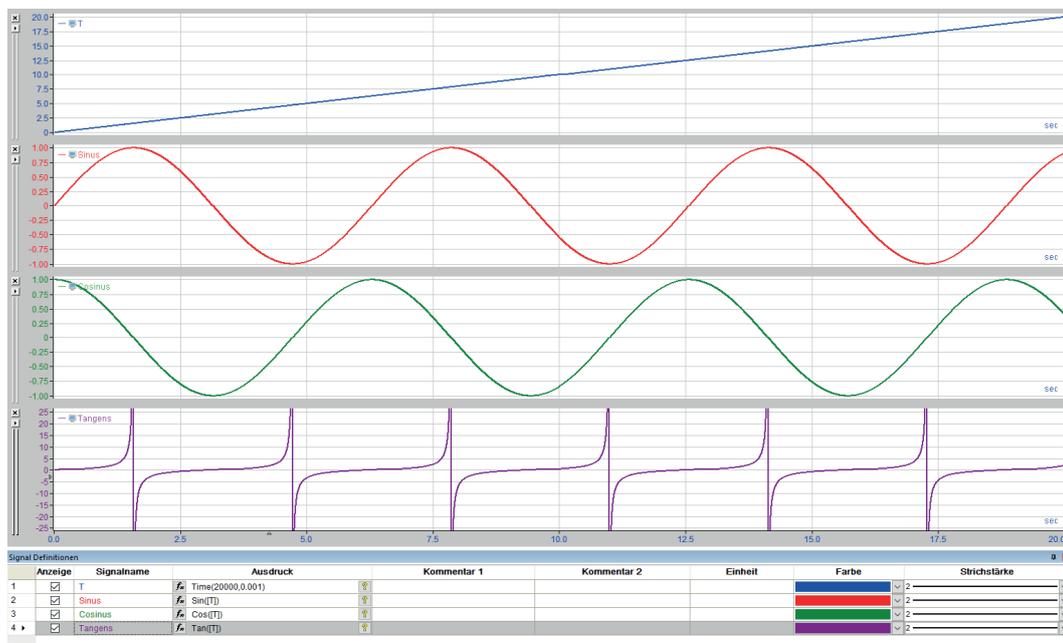


Рис. 8: Изображение различных тригонометрических функций

## 5 Статистические функции

*ibaAnalyzer* поддерживает вычисление различных статистических функций. Для всех функций обычно доступно четыре различных варианта:

- Стандартная функция вычисляет соответствующую величину всегда по всему сигналу.
- Суффикс *InTime* обозначает, что соответствующая величина образуется за интервалы указанной длины
- Суффикс *Valid* используется, чтобы вычислить соответствующую величину за интервалы, обозначенные бинарным сигналом
- Префикс *InTime* обозначает, что соответствующая величина образуется за скользящие интервалы указанной ширины

### 5.1 Среднее значение (Avg)

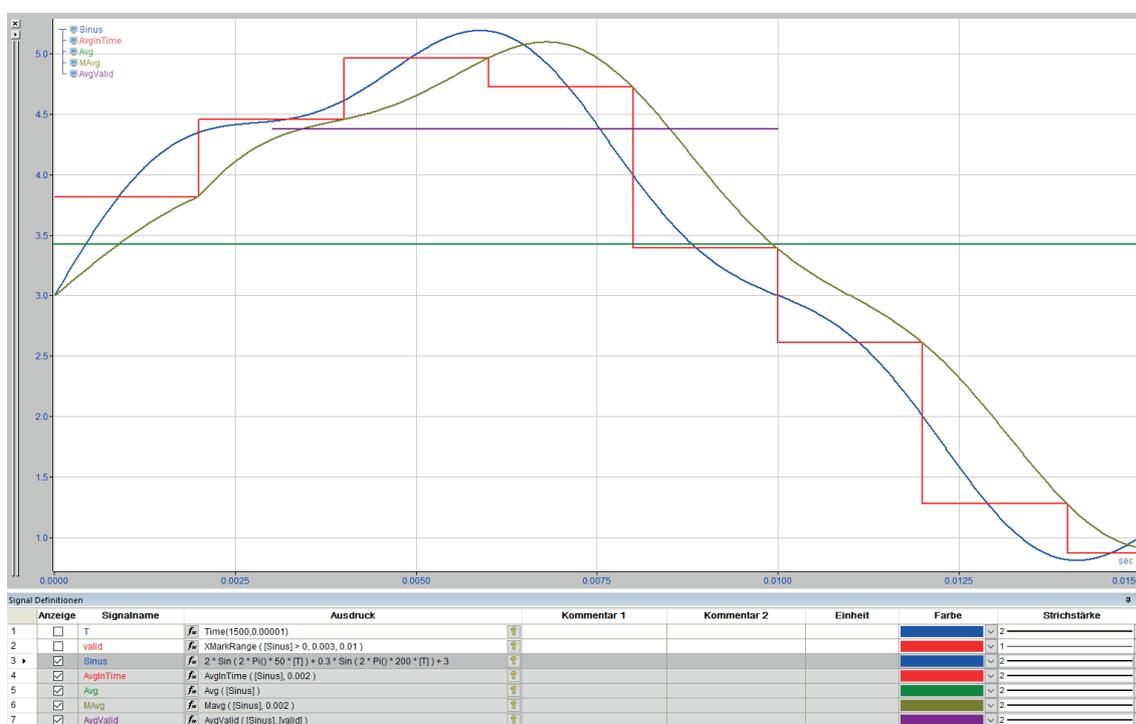


Рис. 9: Статистические функции, среднее значение

#### Avg

Avg ('Expression')

#### Описание

Данная функция возвращает как результат среднее значение 'Expression'. Оно отображается как постоянное значение (горизонтальная линия) в полосе сигналов.

#### AvgInTime

AvgInTime ('Expression', 'Interval')

**Аргументы**

'Expression'	Измеренное значение, для которого образуется среднее значение
'Interval'	Значение длины интервала

**Описание**

Данная функция возвращает как результат среднее значение каждого временного интервала длины 'interval' выражения 'Expression'.

**MAvg**

MAvg('Expression', 'Interval')

**Аргументы**

'Expression'	Измеренное значение, для которого образуется среднее значение
'Interval'	Длина интервала, за который вычисляется среднее значение.

**Описание**

Данная функция возвращает как результат скользящее арифметическое среднее значение 'Expression', вычисляемое за 'Интервал' в секундах.

**Совет**

При помощи данных функций можно также обрабатывать сигналы и выражения, но не на базе времени, а на базе длины, частоты или 1/длина. Вместо секунд необходимо указать диапазон оси X в м, Гц или 1/м в зависимости от соответствующей базы.

**AvgValid**

AvgValid('Expression', 'Valid')

**Аргументы**

'Expression'	Измеренное значение, для которого образуется среднее значение
'Valid'	Контрольный сигнал

**Описание**

Данная операция возвращает как результат среднее значение 'Expression' для интервала (время или длина), в котором соответствующий контрольный сигнал ИСТИНА.

## 5.2 Maxima (Max)

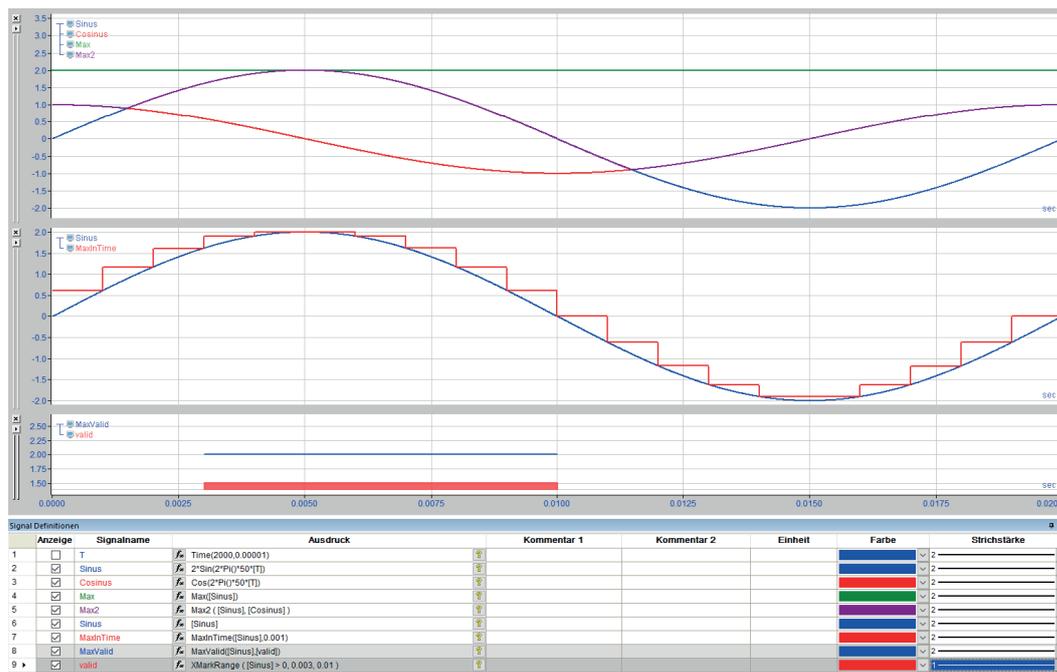


Рис. 10: Статистические функции, максимум

### Max

Max('Expression')

### Описание

Данная функция возвращает как результат максимальное значение 'Expression'. Оно отображается как постоянное значение (горизонтальная линия) в полосе сигналов.

### Max2

Max2('Expression1', 'Expression2')

### Описание

Данная функция возвращает как результат максимум из двух сигналов 'Expression1' и 'Expression2'. Оба сигнала сравниваются значение за значением, и соответственно большее значение передается как результат.

### MaxInTime

MaxInTime('Expression', 'Interval')

### Аргументы

'Expression'	Измеренное значение, для которого образуется максимальное значение
'Interval'	Длина интервала, за который вычисляется максимум.

**Описание**

Данная функция возвращает как результат максимальное значение 'Expression' в пределах каждого интервала длины 'Interval'. Могут обрабатываться сигналы и выражения на базе времени ('Interval' в секундах) или на базе длины ('Interval' в метрах)

**MaxValid**

```
MaxValid('Expression', 'Valid')
```

**Аргументы**

'Expression'	Измеренное значение, для которого образуется максимальное значение
'Valid'	Контрольный сигнал

**Описание**

Данная операция возвращает как результат максимум 'Expression' для интервала (время или длина), в котором соответствующий контрольный сигнал ИСТИНА.

**MMax**

```
MMax('Expression', 'Interval')
```

**Аргументы**

'Expression'	Измеренное значение, для которого образуется максимальное значение
'Interval'	Длина интервала, за который вычисляется максимум

**Описание**

Данная функция возвращает максимум 'Expression' в пределах скользящего интервала оси X длины 'Interval', продвигаясь соответственно на точку измерения.

### 5.3 Minima (Min)

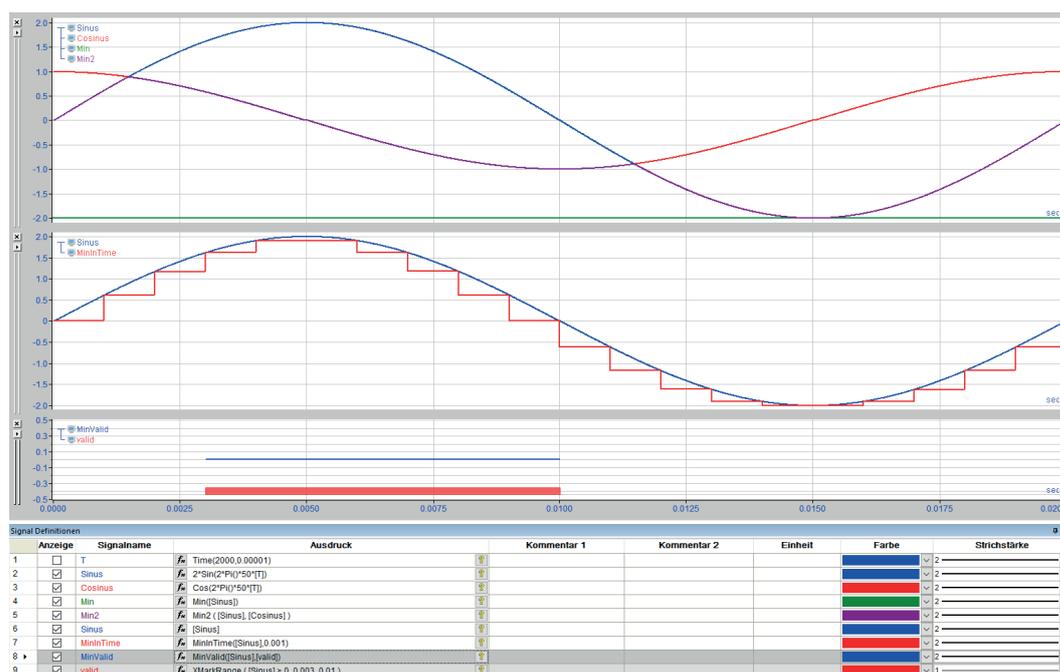


Рис. 11: Статистические функции, минимум

#### Min

Min ('Expression')

#### Описание

Данная функция выводит как результат минимальное значение сигнала 'Expression'. Оно отображается как постоянное значение (горизонтальная линия) в полосе сигналов.

#### Min2

Min2 ('Expression1', 'Expression2')

#### Описание

Данная функция возвращает как результат минимум из двух сигналов 'Expression1' и 'Expression2'. Оба сигнала сравниваются значение за значением, и соответственно меньшее значение передается как результат.

#### MinInTime

MinInTime ('Expression', 'Interval')

#### Аргументы

'Expression'	Измеренное значение, для которого образуется минимальное значение
'Interval'	Длина интервала, за который вычисляется минимум.

**Описание**

Данная функция возвращает как результат минимальное значение 'Expression' в пределах каждого интервала длины 'Interval'. Могут обрабатываться сигналы и выражения на базе времени ('Interval' в секундах) или на базе длины ('Interval' в метрах)

**MinValid**

```
MinValid('Expression', 'Valid')
```

**Аргументы**

'Expression'	Измеренное значение, для которого образуется минимальное значение
'Valid'	Контрольный сигнал

**Описание**

Данная операция возвращает как результат минимум 'Expression' для интервала (время или длина), в котором соответствующий контрольный сигнал ИСТИНА.

**MMin**

```
MMin('Expression', 'Interval')
```

**Аргументы**

'Expression'	Измеренное значение, для которого образуется минимальное значение
'Interval'	Длина интервала, на базе которого вычисляется минимум

**Описание**

Данная функция возвращает минимум выражения 'Expression' в пределах скользящего интервала оси X длины 'Interval', продвигаясь соответственно на одну точку измерения.

## 5.4 Стандартное отклонение (StdDev)

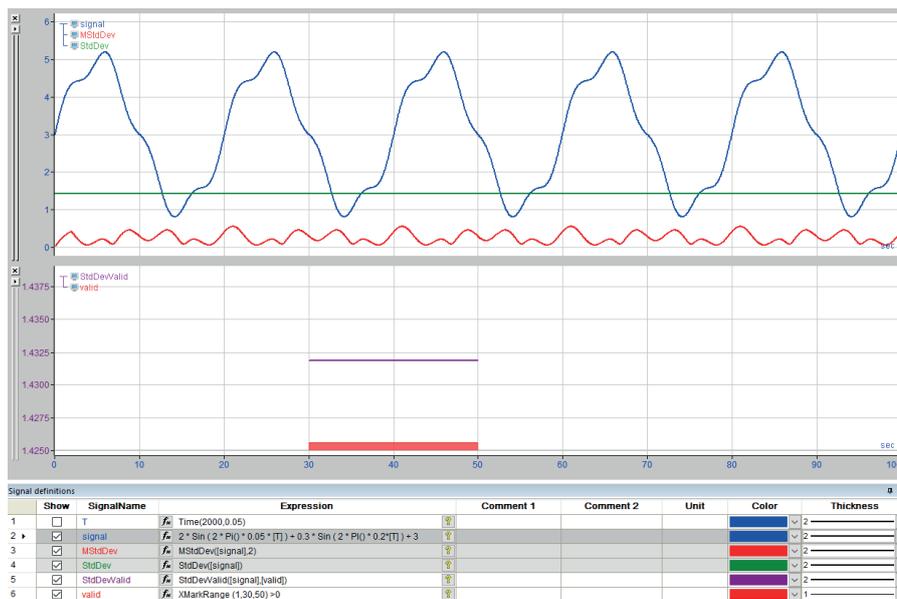


Рис. 12: Статистические функции, стандартное отклонение StdDev, MStdDev

### StdDev

StdDev ('Expression')

#### Описание

Данная функция возвращает как результат стандартное отклонение 'Expression'.

Вычисление стандартного отклонения осуществляется по формуле:

$$s_x = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

$s_x$  = Standardabweichung  
 $\bar{x}$  = Mittelwert  
 n = Anzahl Messungen

### MStdDev

MStdDev ('Expression', 'Interval')

#### Аргументы

'Expression'	Измеренное значение, для которого образуется стандартное отклонение
'Interval'	Длина интервала, за который вычисляется стандартное отклонение.

#### Описание

Данная функция возвращает как результат скользящее стандартное отклонение 'Expression' за каждый интервал времени длины 'Interval'. Могут обрабатываться сигналы и выражения на базе времени ('Interval' в секундах) или на базе длины ('Interval' в метрах)

**StdDevInTime**

```
StdDevInTime('Expression', 'Interval')
```

**Аргументы**

'Expression'	Измеренное значение, для которого образуется стандартное отклонение
'Interval'	Длина интервала, за который вычисляется стандартное отклонение.

**Описание**

Данная функция возвращает как результат стандартное отклонение 'Expression' через каждый интервал времени длины.

**Примечание**

Стандартное отклонение всегда указывается для предыдущего интервала.

**StdDevValid**

```
StdDevValid('Expression', 'Valid')
```

**Аргументы**

'Expression'	Измеренное значение, для которого образуется стандартное отклонение
'Valid'	Контрольный сигнал

**Описание**

Данная функция возвращает как результат стандартное отклонение 'Expression' для интервала (время или длина), в котором соответствующий контрольный сигнал ИСТИНА.

## 5.5 Процентиль (Percentile)

**Percentile**

```
Percentile('Expression', 'p'=0.5)
```

**Аргументы**

'Expression'	Измеренное значение, для которого вычисляется процентиль
'p'	Процентиль

**Описание**

Данная функция возвращает как результат 'p'-тый процентиль выражения 'Expression'.

'p'-тый процентиль обозначает наименьшее значение ряда измеренных значений, которое больше р% количества измеренных значений. Типичный процентиль - это 50%-процентиль, так называемая медиана. Медиана делит ряд измеренных значений на две равные части: 50% всех измеренных значений меньше значения медианы, другие 50% - больше или равны. Другие наиболее употребляемые процентиля - это 25% и 75%, которые вместе с медианой делят ряд измеренных значений на четыре группы, так называемые квартили. (< 25%, <50%, <75%, ≥75%).

Функция "Percentile" определяет значение процентиля из общего ряда точек измерений сигнала. Величина процентиля 'p' должна быть указана в десятичных значениях:

- 50 % --> p = 0.5 (значение по умолчанию)
- 75 % --> p = 0.75
- 95,9 % --> p = 0.959

Данная функция подходит, например, для оценки качества продукта, при которой определенное свойство продукта должно удовлетворять заданной классификации.

### PercentileValid

```
PercentileValid('Expression', 'Valid', 'p'=0.5)
```

#### Аргументы

'Expression'	Измеренное значение, для которого вычисляется процентиль
'Valid'	Значение интервала, за который образуется процентиль
'p'	Процентиль

#### Описание

Данная операция возвращает как результат процентиль выражения 'Expression' для каждого интервала (время или длина), в котором соответствующий контрольный сигнал 'Valid' является ИСТИНОЙ.

### PercentileInTime

```
PercentileInTime('Expression', 'Interval', 'p'=0.5)
```

#### Аргументы

'Expression'	Измеренное значение, для которого вычисляется процентиль
'Interval'	Значение интервала, за который образуется процентиль
'p'	Процентиль

#### Описание

Данная функция возвращает как результат процентиль выражения 'Expression' за каждый интервал времени длины 'Interval'.

### MPercentile

```
MPercentile('Expression', 'Interval', 'p'=0.5)
```

#### Аргументы

'Expression'	Измеренное значение, для которого вычисляется процентиль
'Interval'	Значение скользящего интервала, за который образуется процентиль
'p'	Процентиль

**Описание**

Данная функция возвращает как результат скользящий процентиль выражения 'Expression' за каждый интервал времени длины 'Interval'. Могут обрабатываться сигналы и выражения на базе времени ('Interval' в секундах) или на базе длины ('Interval' в метрах)

## 5.6 Корреляция и ковариация (Correl, CoVar)

### Correl

```
Correl('Expression1', 'Expression2')
```

#### Аргументы

'Expression1/2'	Измеренные значения, для которых вычисляется коэффициент корреляции
-----------------	---

#### Описание

Данная функция вычисляет коэффициент корреляции между 'Expression1' и 'Expression2'. Рассматривается вся длина записи. Функция возвращает постоянное значение.

### Mcorrel

```
Mcorrel('Expression1', 'Expression2', 'Interval')
```

#### Аргументы

'Expression1/2'	Измеренные значения, для которых вычисляется коэффициент корреляции
'Interval'	Значение интервала, на базе которого образуется коэффициент корреляции

#### Описание

Данная функция вычисляет коэффициент корреляции между 'Expression1' и 'Expression2' через скользящие интервалы величины 'Interval', измеренные в с, м, Гц или 1/м.

### CoVar

```
CoVar('Expression1', 'Expression2')
```

#### Аргументы

'Expression1/2'	Измеренные значения, для которых вычисляется коэффициент ковариация
-----------------	---

#### Описание

Данная функция вычисляет ковариацию между 'Expression1' и 'Expression2'. Рассматривается вся длина записи. Функция возвращает постоянное значение.

### MCoVar

```
MCoVar('Expression1', 'Expression2', 'Interval')
```

#### Аргументы

'Expression1/2'	Измеренные значения, для которых вычисляется коэффициент ковариация
'Interval'	Значение интервала, на базе которого образуется ковариация

**Описание**

Данная функция вычисляет ковариацию между 'Expression1' и 'Expression2' через скользящие интервалы длины 'Interval', измеренные в с, м, Гц или 1/м.

## 5.7 Куртозис

Вычисление куртозиса используется, например, для оценки и анализа вибраций. Он служит для определения количества отклонений в пределах сигнала вибрации.

С математической точки зрения, куртозис является показателем относительной «плоскостности» распределения (по сравнению с нормальным распределением, имеющим нулевой куртозис). Положительный куртозис отображает конусообразное распределение (так называемое островершинное распределение), а отрицательный куртозис отображает плоское распределение (плосковершинное распределение).



Рис. 13: Различные функции куртозиса

Данный статистический метод особенно подходит для анализа случайных или стохастических сигналов, например, в случае техобслуживания по состоянию (мониторинг состояния) при виброанализе.

Для описания графика сигнала используются методы плотности или частотности распределения вероятностей. Предполагается, что сигнал шума с амплитудным распределением Гаусса может быть измерен в исправных установках после отфильтровывания, например, частотных составляющих колебаний вибраций. В случае возникновения неисправностей отдельные импульсные сигналы смешиваются с данным сигналом, изменяя функцию распределения. Образование подходящих показателей, например, коэффициент амплитуды и коэффициент куртозиса, позволяет проводить оценку состояния установки.

При регулярных измерениях данные методы обеспечивают обзор состояния установки. Однако, недостаток заключается в том, что показатели после повышения снова снижаются. Причина в том, что количество импульсных сигналов возрастает при прогрессирующей неисправности. Это, в свою очередь влияет, на эффективное значение, но не на пиковое значение.

Изменения сигнала времени, вызванные ударными импульсами, влияют на изменения результирующей функции распределения. Неисправности очевидно дискретного характера могут вызвать резкое повышение коэффициента куртозиса. Таким образом его абсолютное значение позволяет делать выводы о неисправностях.

Вычисление куртозиса осуществляется по аналогичному образцу, что и вычисление стандартного отклонения, `StdDev`.

### Куртозис

`Kurtosis('Expression')`

#### Аргументы

'Expression'	Измеренное значение, для которого образуется куртозис
--------------	---

#### Описание

Данная операция возвращает в качестве результата куртозис выбранного сигнала времени (выражение).

### KurtosisInTime

`KurtosisInTime('Expression', 'Interval')`

#### Аргументы

'Expression'	Измеренное значение, для которого образуется куртозис
'Interval'	Длина интервала, по которому вычисляется куртозис.

#### Описание

При данной операции выбранное выражение делится на равные интервалы величины 'Interval'. Для данных интервалов будет затем осуществляться вычисление куртозиса.

### MKurtosis

`MKurtosis('Expression', 'Interval')`

#### Аргументы

'Expression'	Измеренное значение, для которого образуется куртозис
'Interval'	Длина интервала в секундах, по которому вычисляется куртозис.

#### Описание

При данной операции вычисляется куртозис 'выражения' в постоянном, но скользящем интервале оси X.

### KurtosisValid

`KurtosisValid('Expression', 'Valid')`

#### Аргументы

'Expression'	Измеренное значение, для которого образуется куртозис
'Valid'	Контрольный сигнал

**Описание**

Данная операция возвращает куртозис для диапазонов, в которых соответствующий контрольный сигнал ИСТИНА.

## 5.8 Асимметрия (Skewness)

Аналогично коэффициенту кurtозиса коэффициент асимметрии предназначен для оценки и анализа колебаний. Коэффициент асимметрии может быть применен, если, например, должны быть проверены свойства симметрии сигнала вибрации.

С математической точки зрения речь идет при этом об оценке асимметрии функции распределения. Распределение называется правоасимметричным (или левоасимметричным), если основная часть распределения сконцентрирована в левой (или в правой) части. Степень асимметрии определяется третьим моментом распределения.

Процесс расчета асимметрии аналогичен функциям кurtозиса и среднеквадратичного отклонения:

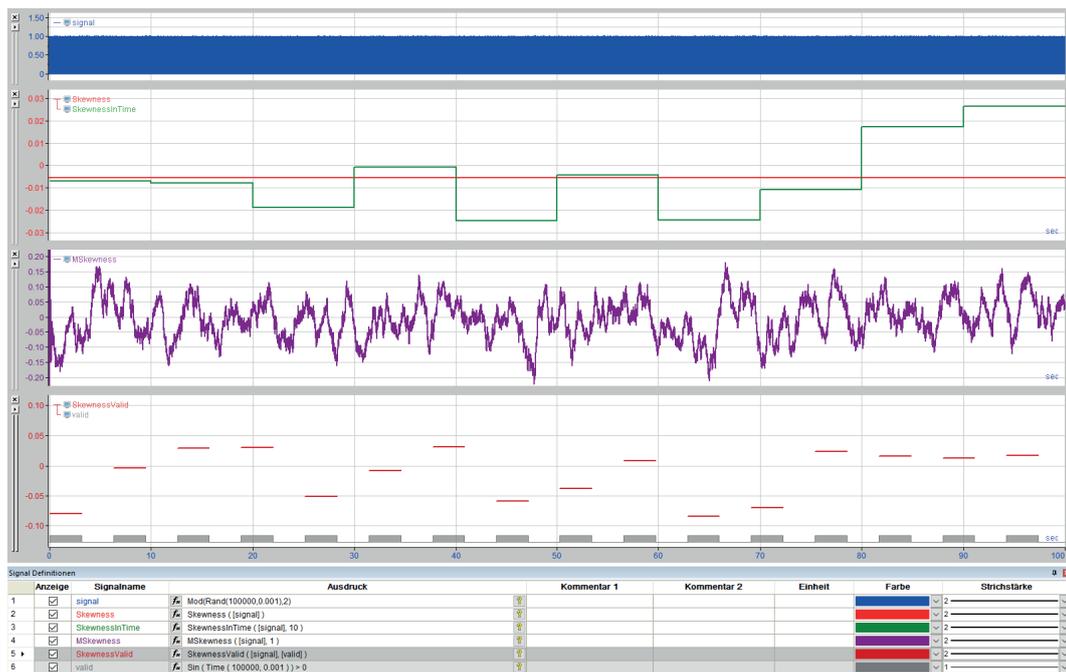


Рис. 14: Различные функции асимметрии

### Асимметрия

`Skewness ('Expression')`

### Описание

Данная операция возвращает в качестве результата асимметрию выбранного сигнала времени 'Expression'.

### SkewnessInTime

`SkewnessInTime ('Expression', 'Interval')`

### Аргументы

'Expression'	Измеренное значение, для которого образуется асимметрия
'Interval'	Длина интервала, за который вычисляется асимметрия.

**Описание**

При данной операции выбранное выражение делится на равные интервалы величины 'Interval'. Для данных интервалов будет затем осуществляться вычисление асимметрии.

**MSkewness**

```
MSkewness('Expression', 'Interval')
```

**Аргументы**

'Expression'	Измеренное значение, для которого образуется асимметрия
'Interval'	Длина скользящего интервала, за который вычисляется асимметрия.

**Описание**

При данной операции вычисляется асимметрия 'выражения' за постоянный, но скользящий интервал оси X.

**SkewnessValid**

```
SkewnessValid('Expression', 'Valid')
```

**Аргументы**

'Expression'	Измеренное значение, для которого образуется асимметрия
'Valid'	Контрольный сигнал

**Описание**

Данная операция возвращает асимметрию для диапазонов, в которых соответствующий контрольный сигнал ИСТИНА.

## 6 Счет и сортировка

### 6.1 Count

```
Count('Expression', 'Level'=0.5, 'Hysteresis'=0, 'EdgeType'=1, 'Reset=0')
```

#### Аргументы

'Expression'	Измеренное значение	
'Level'	Значение уровня	
'Hysteresis'	Значение полосы гистерезиса	
'EdgeType'	Указание, какие фронты должны считаться: передние, задние или передние и задние	
	'EdgeType' <0	только задние фронты (выход из полосы гистерезиса в отрицательном направлении)
	'EdgeType' >0	только передние фронты (выход из полосы гистерезиса в положительном направлении)
	'EdgeType' = 0	передние и задние фронты
'Reset'	Оptionальный цифровой параметр, который может использоваться для сброса показаний счетчика. 'Reset' может сам быть выражением.	
	'Reset' > 0	Счетчик сбрасывается
	'Reset' = 0	Значение счетчика сохраняется/счетчик считает дальше (параметр)

#### Примечание



Условие 'Reset' не должно относиться к самой функции *Count*.

#### Описание

Данная функция считает, сколько раз 'Expression' пересекает уровень 'Level'.

При помощи параметра 'Hysteresis' можно указать полосу допуска, расположенную в равных частях над и под 'Level'. Считаются только полные пересечения полосы допуска.

Параметр 'EdgeType' определяет, какие фронты будут считаться. Параметр 'Reset' служит для сброса значения счетчика на 0. 'Reset' может быть также сформулирован как выражение.

#### Пример

При указании для 'Level' значения 2,5 и для 'Hysteresis' 2,0 пересечения уровня будут считаться в возрастающем направлении только при 'Expression' > 3,5 и в нисходящем направлении только при 'Expression' < 1,5.

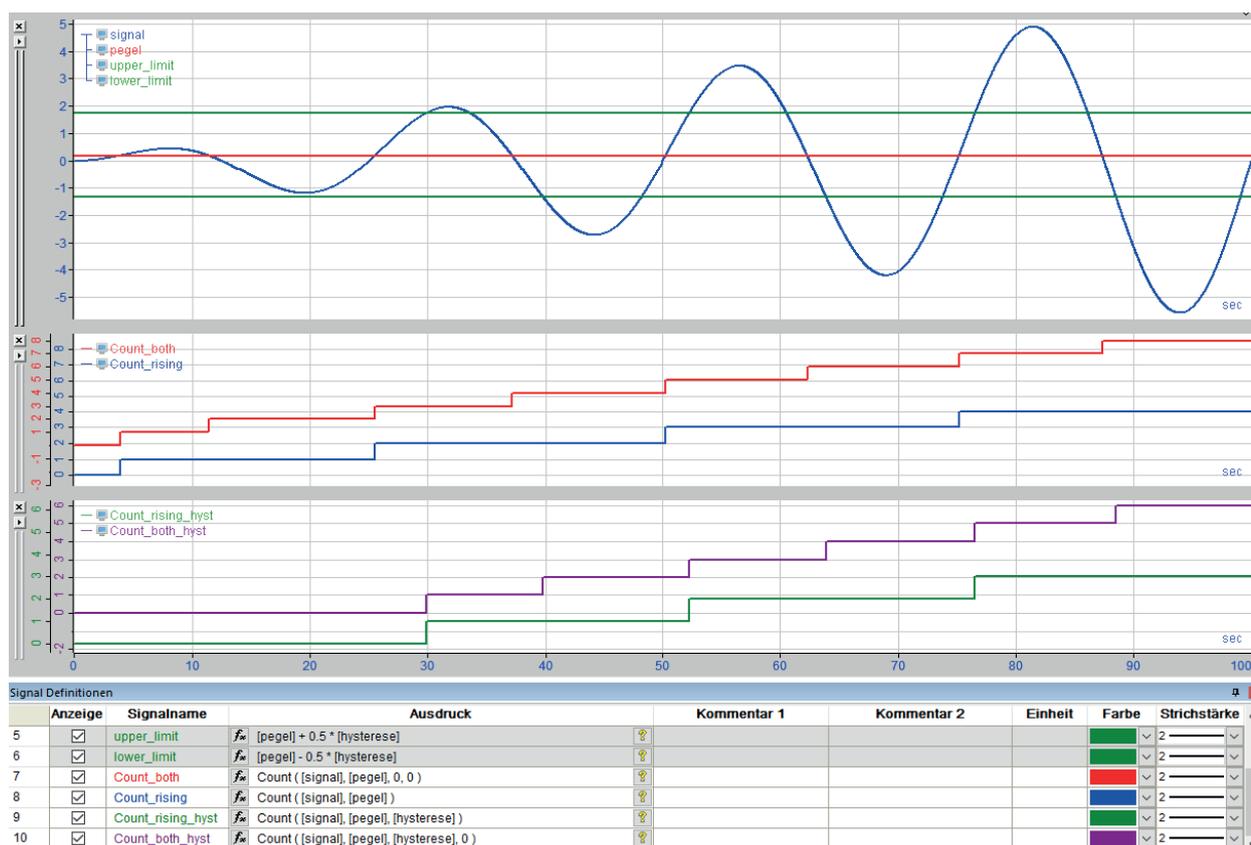


Рис. 15: Различные функции, Count

**Совет**



Функция *Count* может также использоваться для бинарных сигналов. Для этого указывают как уровень 0,5 и как гистерезис, например, 0,1. При этом учитывается и считается каждый переход с ЛОЖЬ на ИСТИНА и наоборот.

## 6.2 CountSamples

`CountSamples('Expression', 'Reset'=0)`

**Аргументы**

'Expression'	Измеренное значение, для которого определяется количество точек сигналов
'Reset'	<p>Опциональный цифровой параметр, который может использоваться для сброса или подавления процесса подсчета. 'Reset' может быть сам выражением</p> <p>'Reset' &gt; 0 Операция подсчета сбрасывается.</p> <p>'Reset' = 0 Операция подсчета активируется (уставка)</p>

**Описание**

Данная функция позволяет рассчитать количество отдельных точек сигналов, независимо от их равноудаленности друг от друга. Недопустимые сигналы при подсчете не учитываются. Если входной сигнал недопустим, то в качестве результата выводится постоянное значение 0.

**Совет**

Данная функция может быть использована, например, в сочетании с XMarkValid (см. функции XMark ➔ XMarkRange / XMarkValid, страница 57).

**6.3 Sort**

```
Sort('Expression', 'Descending'=0)
```

**Аргументы**

'Expression'	Измеренное значение, для которого осуществляется сортировка семплов
'Descending'	Опциональный цифровой параметр для изменения направления порядка сортировки

**Описание**

Данная функция сортирует все семплы графика ('Expression') в соответствии с их значениями в порядке возрастания слева направо.

Уставка: Сортировка в порядке возрастания ('Descending'=ЛОЖЬ). Если семплы нужно сортировать в порядке возрастания слева направо, то в качестве второго операнда нужно установить ИСТИНА.

## 7 Функции времени/длины

### 7.1 Преобразование и передискретизация

#### 7.1.1 ConvertBase

```
ConvertBase('Expression', 'From', 'To')
```

##### Аргументы

'Expression'	Измеренное значение, для которого изменяется база
'From'/'To'	Настройка базы, с которой/на которую нужно заменить. 0 = время 1 = длина 2 = частота 3 = обратная длина

##### Описание

Данная операция изменяет базу выражения с одной на другую. Физического пересчета или масштабирования не осуществляется.

Функция служит для изменения опорных величин сигнала. Это может потребоваться, если для дополнительных вычислений используются, например, референсные значения на базе длины, но существующий сигнал представлен только на базе времени

#### 7.1.2 Resample (Neuabtasten)

```
Resample('Expression', 'Basis', 'interpolate'=1)
```

##### Аргументы

'Expression'	Измеренное значение, подлежащее повторной дискретизации
'Basis'	Новая частота дискретизации результата
'interpolate'	Опциональный параметр для преотращения автоматической интерполяции для новых измеренных значений

##### Описание

Данное действие возвращает как результат график сигнала 'Expression' с новым опорным временем. При этом из оригинального графика моментальные значения переносятся с корректным временем в соответствии с новым опорным временем, чтобы длина нового графика была практически такой же. Функция может также использоваться для сигналов на базе длины. Вместо периода времени необходимо ввести значение пути в м.

**Совет**

График можно графически сгладить, если в функции передискретизации использовать большее опорное время, т.к. меньше точек соединяются друг с другом. Значения не усредняются.

### 7.1.3 SampleAndHold

```
SampleAndHold('Expression', 'Sample', 'Initial'=0)
```

#### Аргументы

'Expression'	Измеренное значение
'Sample'	Параметр, который определяет, вытекает ли функция из измеренного значения (1) или сохраняет последнее измеренное значение (0). 'Sample' может быть сам условием или задаваться другой функцией.
'Initial'	Опциональный параметр (уставка=0), который задает исходное значение функции, если в начале измерения 'Sample' неактивен.

#### Описание

Данная функция является функцией выборки/хранения. Вывод вытекает из 'Expression', если 'Sample' = TRUE. Он остается без изменений, если 'Sample' = FALSE. Опциональный параметр 'Initial' позволяет указывать исходное значение выхода, если при вызове функция стоит на «Сохранить».

### 7.1.4 SampleOnce

```
SampleOnce('Expression', 'Sample')
```

#### Аргументы

'Expression'	Измеренное значение
'Sample'	Цирковой сигнал, передние фронты которого задают точки выборки.

#### Описание

Данная функция считывает входной сигнал 'Expression' в отдельных точках, задаваемых передним фронтом цифрового сигнала 'Sample'. Результат содержит точку измерения в каждом участке переднего фронта и неверен на участках между ними.

#### Пример

Данная функция может быть использована для изображения сигнала датчика фаз (кифазора) в диапазоне времени. Каждый раз, когда датчик фаз переходит в 'ИСТИНА', считывается исходный сигнал. Посредством наложения обоих изображений соответствующим образом отображаются точки времени датчика фаз. На примере ниже можно увидеть смещение фазы на 180° при пересечении резонанса.

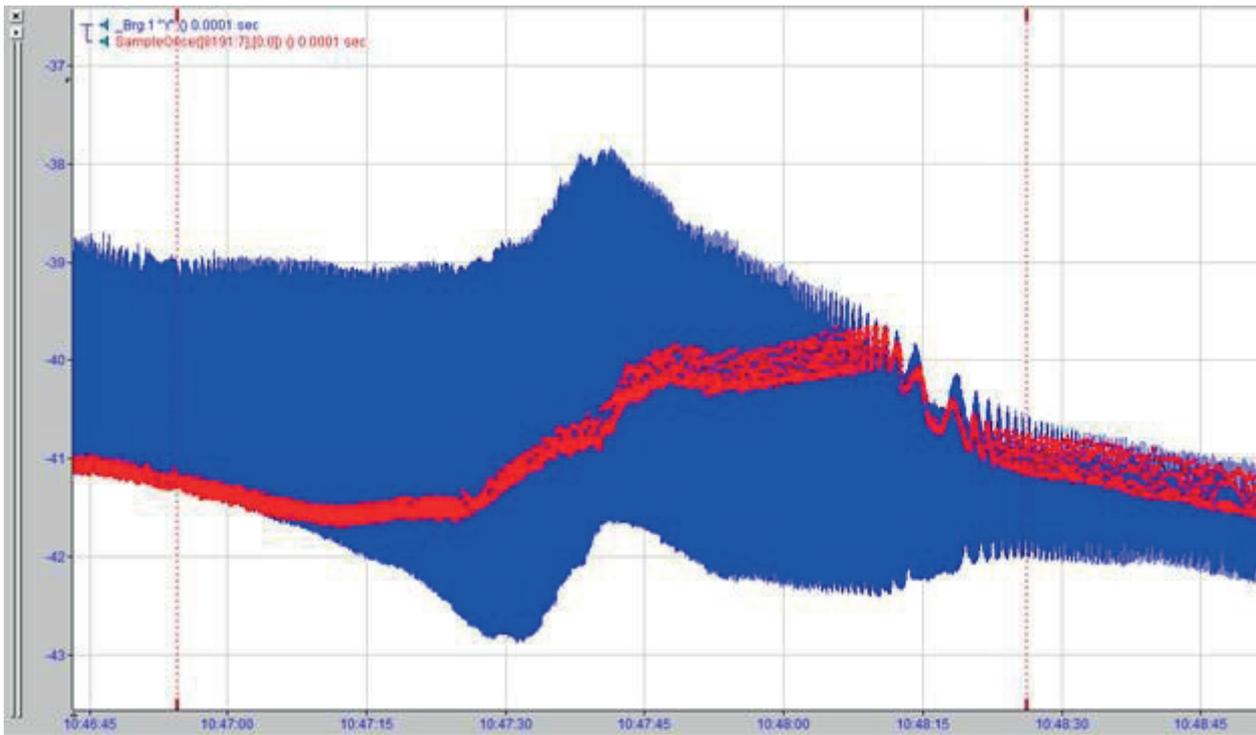


Рис. 16: Функция SampleOnce применимо к вибросигналу

## 7.2 Время

### 7.2.1 Время

```
Time('Count', 'Basis')
```

#### Аргументы

'Count'	Количество создаваемых точек измерения
'Basis'	Частота дискретизации результата

Данная функция возвращает как результат линейный, пропорциональный времени сигнал с количеством значений 'Count' с интервалом 'Basis'. Время указывается в секундах. Значения времени наносятся при этом как на оси X, так и на оси Y.

#### Примечание



Для использования функции времени файл измерений загружать не требуется.

### 7.2.2 AbsoluteTime

```
AbsoluteTime('Time', 'DoSync'=0)
```

#### Аргументы

'Time'	Относительное время, подлежащее преобразованию
'DoSync'	Опциональный параметр, который задает настройку абсолютного времени по времени начала текущей отображаемой оси времени.

#### Описание

Данная функция преобразует относительную информацию о времени 'Time' (например, генерированную при помощи First, XLast или Xvalues) в абсолютное время. При этом в зависимости от того, постоянно ли входное время или нет, вектор будет возвращаться с постоянными или изменяющимися записями. Опциональный, бинарный параметр 'DoSync' задает настройку результата по времени начала текущей отображенной оси времени.

Результатом является вектор со следующими записями, которые могут быть считаны при помощи GetRows:

- Индекс 0: миллисекунды
- Индекс 1: секунды
- Индекс 2: минуты
- Индекс 3: часы
- Индекс 4: день месяца
- Индекс 5: месяц

- Индекс 6: год
- Индекс 7: день года
- Индекс 8: День недели (1=понедельник, 2=вторник, ... , 7=воскресенье)

## 7.3 Пересчет с базы времени на базу длины

### TimeToLength

`TimeToLength('Expression', 'Speed', 'Precision')`

#### Аргументы

'Expression'	Выражение, которое должно быть пересчитано на длину
'Speed'	Сигнал скорости
'Precision'	Опциональный параметр, который задает частоту дискретизации результата в м.

#### Описание

Данная функция преобразует измеренное значение на базе времени в значение на базе длины, со скоростью объекта измерения 'Speed' в качестве вектора скорости [м/с].

Каждое измеренное значение, для которого есть подходящее измеренное значение скорости, может быть пересчитано при помощи данной функции в изображение на базе длины. Это значит, что может быть изображена не только взаимосвязь между измеренным значением и временем, но также и между измеренным значением и пройденным путем. На примере металлической полосы на стане при помощи данной функции можно определить распределение измеренных значения по длине полосы. При условии, если процесс был организован таким образом, что начало и конец измерения точно совпадают с началом и концом полосы, то при помощи данной функции можно вычислить и общую длину полосы. Наибольшее определенное значение длины вносится как значение конца полосы на оси X (автомасштабирование).

'Precision' - это опциональные данные в [м]. Если не указано, то точки для графика на базе длины вычисляются в соответствии с количеством точек измерения исходного сигнала и вносятся в полосу сигналов. Если указана точность, например, 0,1, то каждые 0,1 м вычисляется новое значение на базе длины и вносится как точка графика.

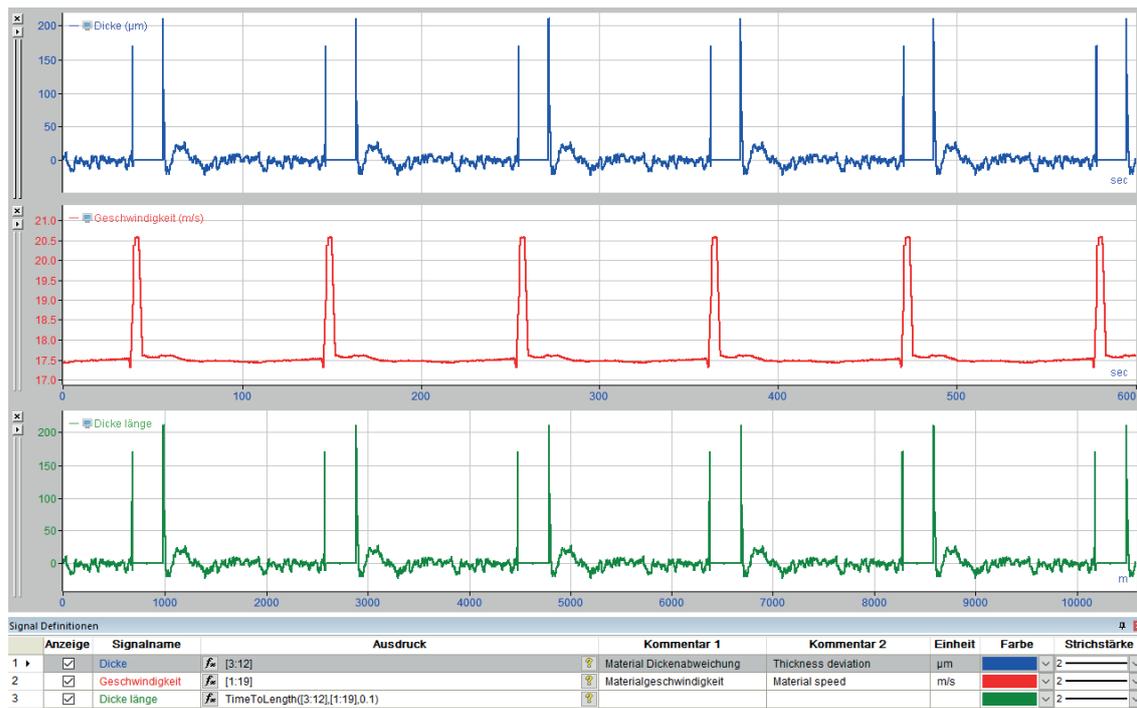


Рис. 17: Функции время/длина TimeToLength

### TimeToLengthL

TimeToLengthL('Expression', 'Length', 'Precision')

#### Аргументы

'Expression'	Выражение, которое должно быть пересчитано на длину
'Length'	Сигнал длины
'Precision'	Оptionальный параметр, который задает частоту дискретизации результата в м.

#### Описание

Данная функция преобразует измеренное значение на базе времени 'Expression' в значение на базе длины, с измеренным значением 'Length' в качестве положения [м].

Пояснения в *TimeToLength* используются аналогично, только вместо скорости используется подходящее измеренное значение длины и положения.

## 8 Действия с осью X

### 8.1 Смещение вдоль оси X

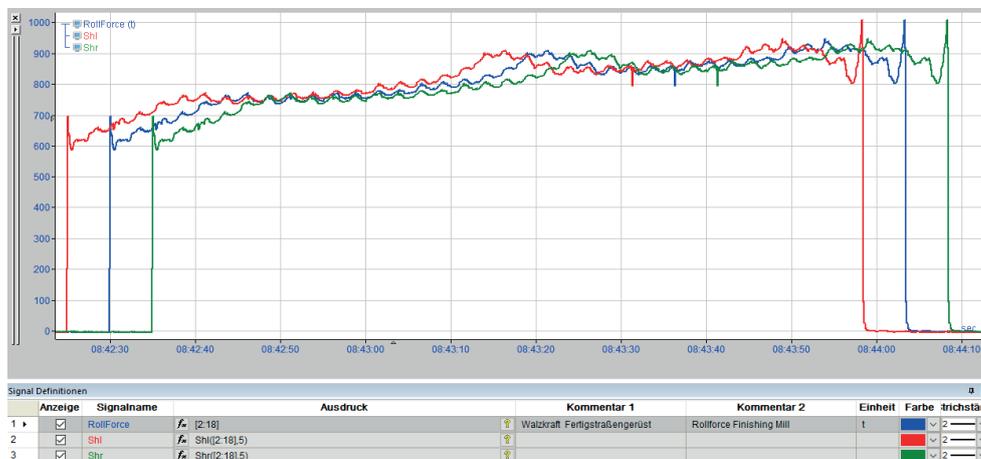


Рис. 18: Функции времени-длины: смещение вправо/влево

#### Shl

`Shl ('Expression', 'Distance')`

#### Аргументы

'Expression'	Выражение, которое должно быть смещено
'Distance'	Смещение в секундах или в метрах при сигналах на базе длины

Данная операция возвращает как результат график сигнала, смещенный на величину длины 'Distance' на оси X влево по отношению к оригинальному сигналу. В противном случае измеренные значения остаются без изменений. Функция может использоваться как для сигналов на базе времени ('смещение' в секундах), так и для сигналов на базе длины ('смещение' в метрах).

#### Shr

`Shr ('Expression', 'Distance')`

#### Аргументы

'Expression'	Выражение, которое должно быть смещено
'Distance'	Смещение в секундах или в метрах при сигналах на базе длины

Данная операция возвращает как результат график сигнала, смещенный на величину длины 'Distance' на оси X вправо по отношению к оригинальному сигналу. В противном случае измеренные значения остаются без изменений. Функция может использоваться как для сигналов на базе времени ('смещение' в секундах), так и для сигналов на базе длины ('смещение' в метрах).

## 8.2 XCutRange / XCutValid

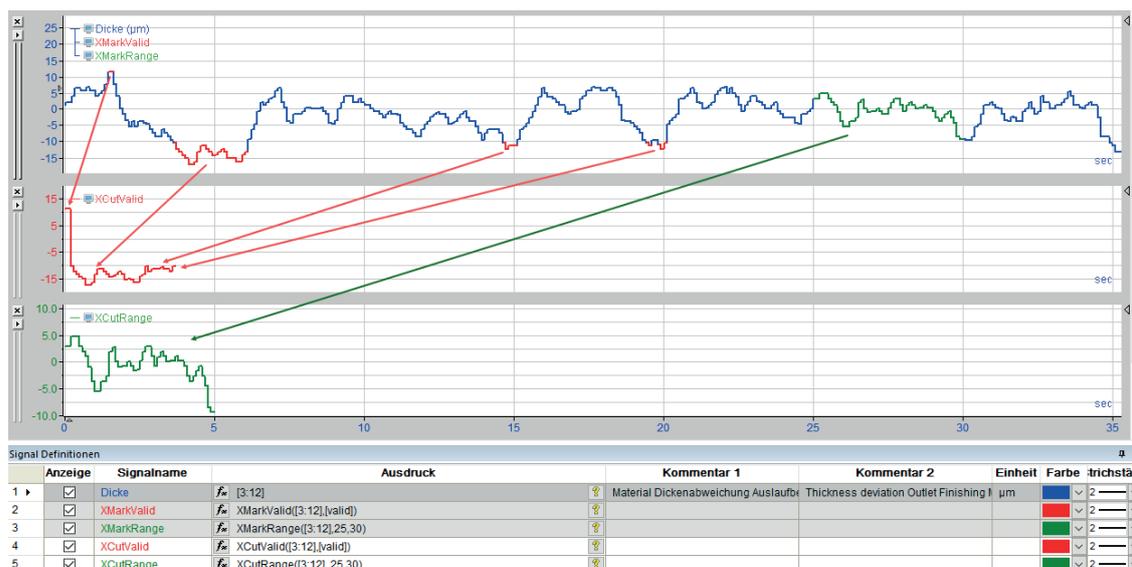


Рис. 19: Операции на оси X XCutRange и XCutValid

### XCutRange

XCutRange('Expression', 'Start', 'End')

#### Аргументы

'Expression'	Выражение, из которого должна быть вырезана часть
'Start'	Начало выбранного диапазона в секундах или метрах
'End'	Конец выбранного диапазона в секундах или метрах

#### Описание

При помощи данной функции может быть вырезан фрагмент графика. Функция может быть применена как к полосам на базе времени, так и на базе длины. Параметры 'Start' и 'End', указанные в [с] или [м], задают начало и конец вырезаемого участка.

Вырезанная часть перемещается в начало собственной полосы сигнала. Т.к. ось X (время или длина) остаются без изменений, корректной базы времени или длины измеренных значений больше нет.

### XCutValid

XCutValid('Expression', 'Valid')

#### Аргументы

'Expression'	Выражение, из которого должна быть вырезана часть
'Valid'	Бинарный сигнал, который описывает выбранный участок

#### Описание

При помощи данной функции вырезаются все точки измерений графика сигналов 'Expression' в зависимости от условия 'Valid', если данное условие возвращает значение ИСТИНА. Функция может быть применена как к сигналам на базе времени, так и на базе длины. Параметр 'Valid' является булевым выражением. Это может быть цифровой вход-

ной сигнал, результат операции сравнения или любое другое бинарное выражение. Точки измерений, для которых условие ЛОЖЬ, игнорируются.

Вырезанные части располагаются друг за другом в начале новой полосы сигналов.

### 8.3 XMarkRange / XMarkValid

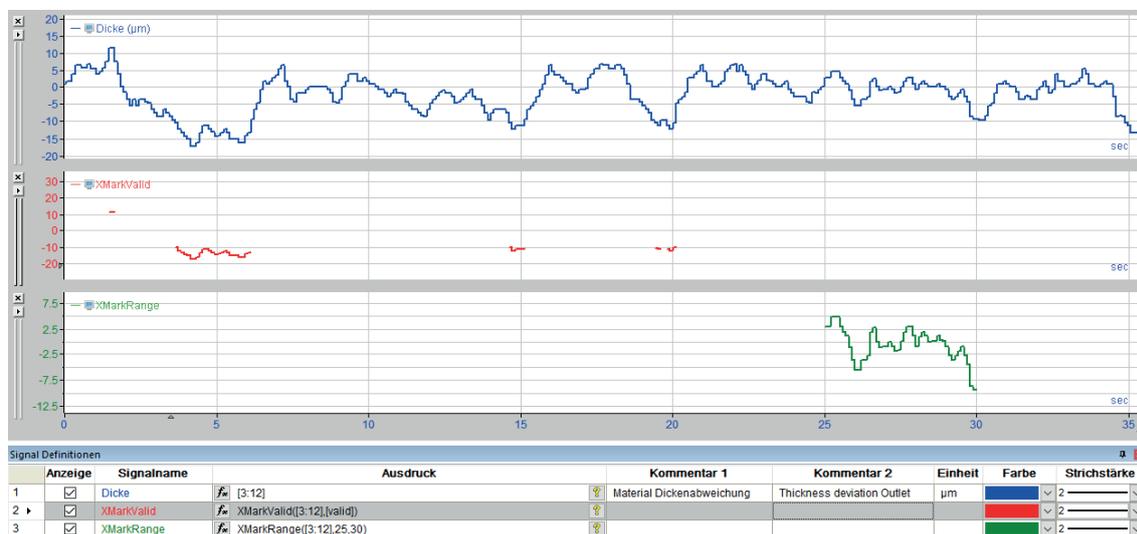


Рис. 20: Ось X, операции X XCutRange и XCutValid

#### XMarkRange

XMarkRange('Expression', 'Start', 'End')

#### Аргументы

'Expression'	Выражение, из которого должна быть выбрана часть
'Start'	Начало выбранного участка в секундах или метрах
'End'	Конец выбранного участка в секундах или метрах

#### Описание

При помощи данной функции может быть вырезан участок характеристики аналогично функции XCutRange. Функция может быть применена как к полосе сигналов на базе времени, так и на базе длины. Параметры 'Start' и 'End', указанные в [с] или [м], задают начало и конец вырезаемого участка. Вырезанная часть отображается в собственной полосе сигналов, но остается в исходном месте на оси времени или пути, а точки измерения, находящиеся за пределами указанного диапазона, игнорируются.

#### XMarkValid

XMarkValid('Expression', 'Valid')

#### Аргументы

'Expression'	Выражение, из которого должна быть выбрана часть
'Valid'	Бинарный сигнал, который описывает выбранный диапазон

### Описание

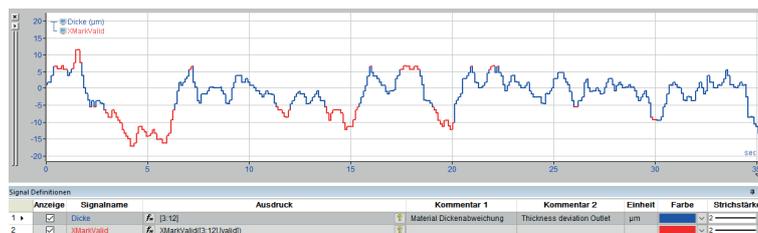
При помощи данной функции - аналогично *XCutValid* - вырезаются все точки измерений графика сигналов 'Expression' в зависимости от условия 'Valid', если данное условие возвращает значение ИСТИНА. Функция может быть применена как к полосам на базе времени, так и на базе длины. Параметр 'Valid' является булевым выражением. Это может быть цифровой входной сигнал, результат операции сравнения или любое другое бинарное выражение. Точки измерений, для которых условие ЛОЖЬ, игнорируются. Вырезанные части изображаются в новой полосе сигналов, сохраняя свои позиции X.

### Совет



Функция *XMarkValid* подходит для выделения цветом на графике сигналов, например, превышений предельных значений, изобразив результирующий сигнал в той же полосе и на той же оси Y, что и исходный сигнал. По различным цветам можно легко распознать диапазоны нарушения предельных значений.

Пример: Значения в пределах допуска = синие, значения за пределами допусками = красный



## 8.4 XMirror / XStretch / XStretchScale

### XMirror

XMirror('Expression')

#### Описание

Данная функция позволяет зеркалить весь график (переставить начало и конец). Зеркалирование осуществляется при этом по вертикальной центральной оси всего графика сигналов. Функция может быть применена как к полосам сигналов на базе времени, так и на базе длины.

Это позволяет лучше сравнивать графики измерений реверсивных процессов (изменение направления). Таким образом, например, в прокатном производстве можно переставить головную часть и хвостовую часть в (четных) проходах, чтобы графически нейтрализовать изменение направления. Чтобы иметь возможность сравнивать несколько проходов друг с другом, соответствующие измеренные значения необходимо предварительно вырезать из оригинального сигнала при помощи функции XCutValid, чтобы иметь возможность зеркалировать их по-отдельности и накладывать друг на друга.

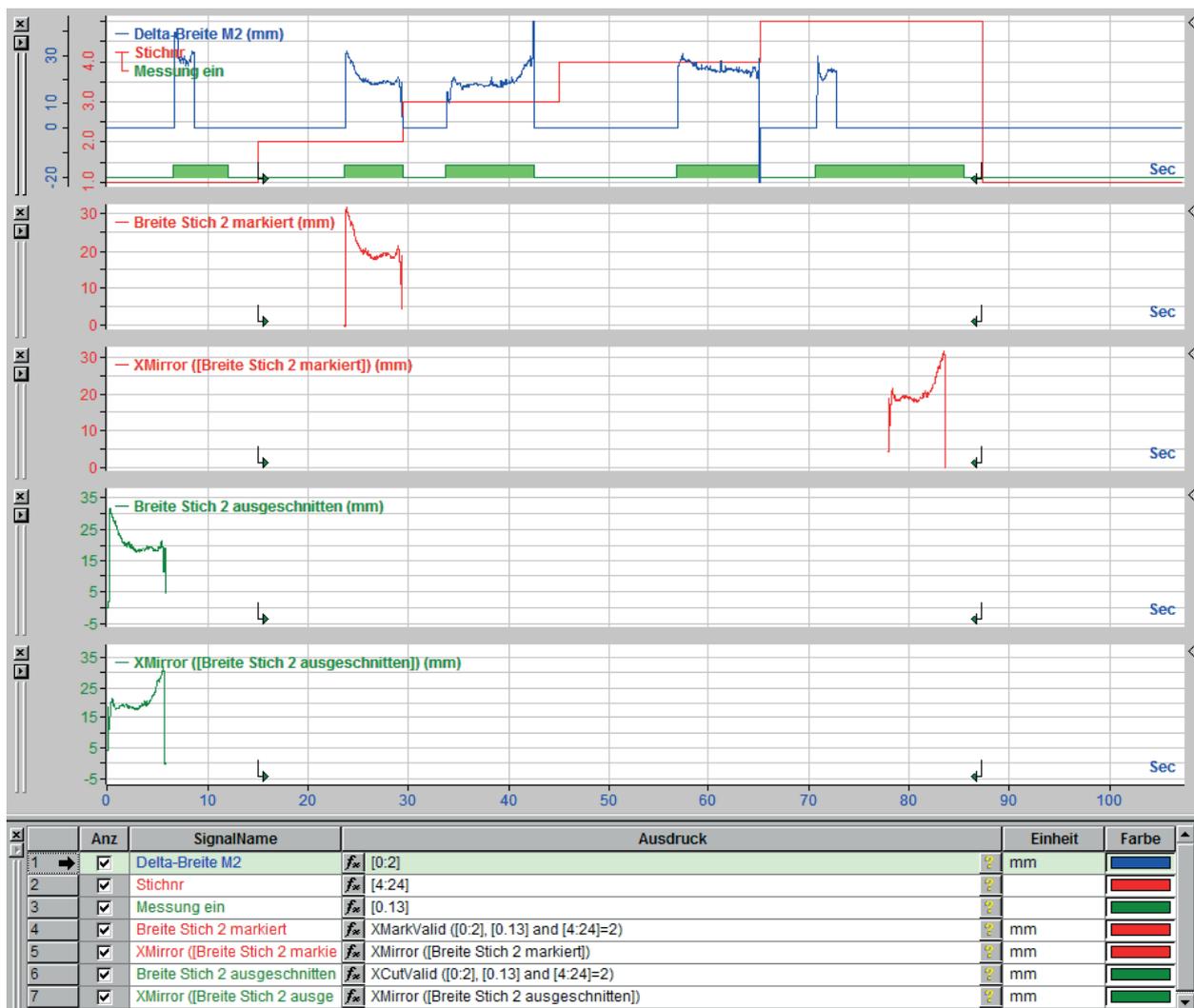


Рис. 21: Операции на оси X XMirror

На изображении выше представлен различный результат зеркалирования, в зависимости от того, был ли зеркалируемый фрагмент предварительно вырезан при помощи *XMarkValid* (красного цвета) или при помощи *XCutValid* (зеленого цвета).

### XStretch

`XStretch('Expression', 'ReferenceExpression')`

#### Описание

При помощи данной функции кривая сигнала может быть графически растянута на ту же конечную длину другого сигнала. Функция может быть применена как к полосам сигналов на базе времени, так и на базе длины.

Таким образом измеренные значения, например, проката черновой группы клетей можно сопоставить с измеренными значениями чистовой группы клетей, или сравнить отдельные проходы реверсивного прокатного стана друг с другом.

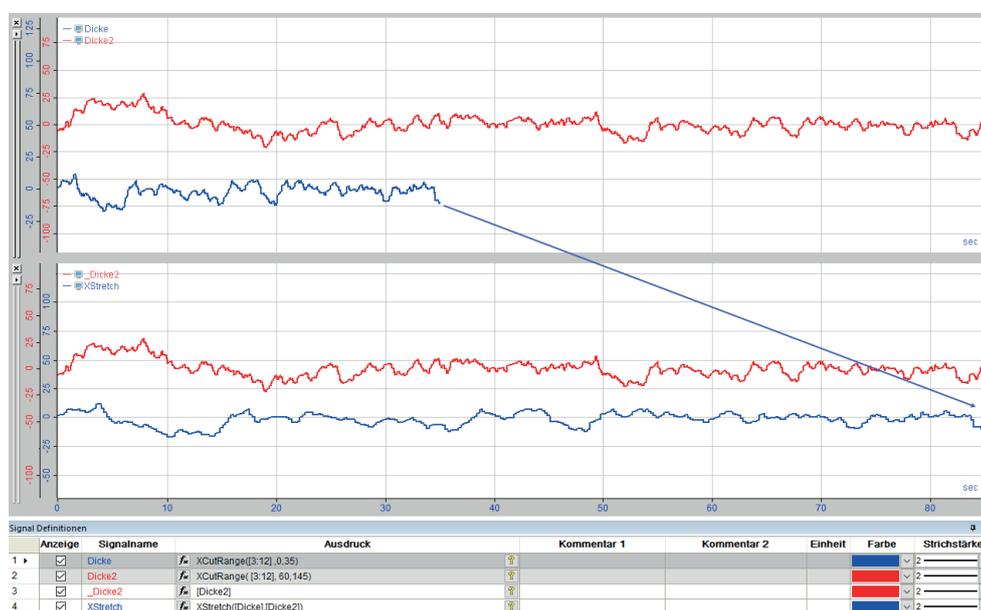


Рис. 22: Операции на оси X XStretch

На рисунке выше график усилия прокатки первого прохода (синий график) растягивается на конечную длину в соответствии с девятым проходом.

### XStretchScale

`XStretchScale('Expression', 'Scale')`

#### Описание

Данная функция позволяет растянуть кривую сигнала на заданный коэффициент. Коэффициент масштабирования может применяться и в случае смещения графика сигнала.

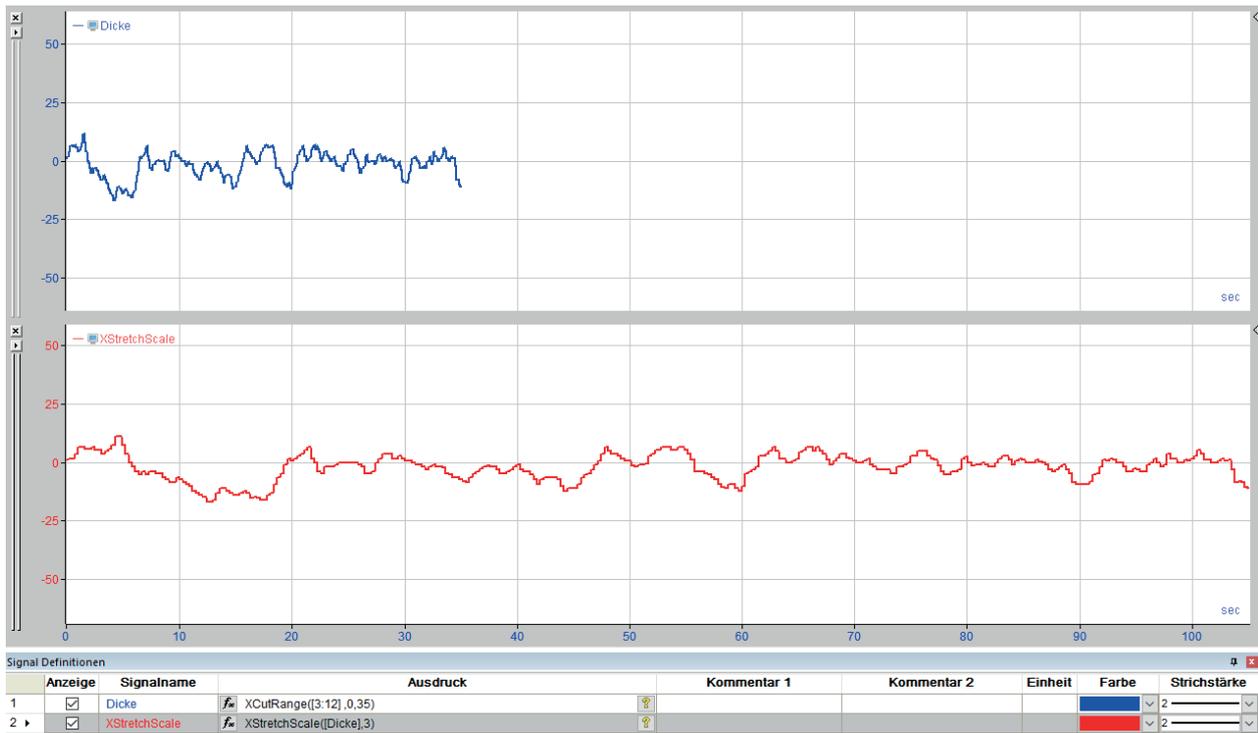


Рис. 23: Операции на оси, X XStretchScale

## 8.5 XFirst / XLast / XNow

### XFirst

`XFirst('Expression', 'Skip=0', 'SkipInitialEdge=FALSE')`

#### Аргументы

'Expression'	(Булев) входной сигнал
'Skip'	Опция для пропуска первого переднего фронта
'SkipInitialEdge'	Решает, будет ли первый семпл считаться передним фронтом, если он ИСТИНА

#### Описание

Данная функция возвращает как результат значение на оси X (время [с] или путь [м]), для которого выражение 'Expression' в первый раз ИСТИНА. Соответственно 'Expression' должно быть булевой величиной. Это может быть цифровой входной сигнал, результат операции сравнения или любое другое бинарное выражение.

### XLast

`XLast('Expression', 'Skip=0', 'SkipFinalEdge=FALSE')`

#### Аргументы

'Expression'	(Булев) входной сигнал
'Skip'	Опция для пропуска последних задних фронтов
'SkipFinalEdge'	Решает, будет ли последний семпл считаться задним фронтом, если он ИСТИНА

#### Описание

Данная функция возвращает как результат значение на оси X (время [с] или путь [м]), для которого выражение 'Expression' в последний раз ИСТИНА. Соответственно 'Expression' должно быть булевой величиной. Это может быть цифровой входной сигнал, результат операции сравнения или любое другое бинарное выражение.

### XNow

`XNow()`

#### Описание

Данная функция возвращает относительное время с момента последнего старта *ibaAnalyzer*.

## 8.6 XSize / XSumValid

### XSize

XSize ('Expression')

#### Описание

Данная функция возвращает как результат общую длину 'Expression' в единицах измерения оси X (время в [с] или путь в [м]). Результат постоянно 0, если входной сигнал некорректен.

### XSumValid

XSumValid ('Expression')

#### Описание

При помощи данной функции определяется время или длина, в течение которых условие 'Expression' ИСТИНА. Все точки измерения, в которых условие не выполняется (ЛОЖЬ) не учитываются при вычислении. Соответственно 'Expression' должно быть булевой величиной. Это может быть цифровой входной сигнал, результат операции сравнения или любое другое бинарное выражение.

Результат постоянно 0, если входной сигнал некорректен.

## 8.7 XValues / YValues

### XValues

XValues ('Expression')

#### Описание

Данная функция возвращает как результат значения X всех точек измерения выражения. Особенностью данной функции является то, что они работают также с сигналами или выражениями не только на базе времени, но и на базе длины (м), частоты (Гц) или обратной длины (1/м).

При обычном непрерывном графике сигналов времени или пути результатом является возрастающая прямая с базовыми единицами измерения (значения времени или пути) на оси Y в с или м. Функция работает также с неравноудаленными измеренными значениями.

### YValues

YValues ('Expression', 'TimeBase=1')

#### Аргументы

'Expression'	Входной сигнал
'TimeBase'	Опорное время выходного сигнала

### Описание

Данная функция возвращает как результат значения X всех измеренных значений выражения. Независимо от того, была ли осуществлена выборка входного сигнала равноудаленно или нет, результат является равноудаленным сигналом с опорным временем 'TimeBase'.

Указание опорного времени является опцией, и как значение по умолчанию используется 'TimeBase'=1.

## 8.8 VarDelay

```
VarDelay('Expression', 'Delay')
```

### Аргументы

'Expression'	Входной сигнал
'Delay'	Время задержки в секундах

### Описание

Данная операция возвращает как результат выражение 'Expression' с задержкой на постоянную времени 'Delay'.

## 8.9 XY

```
XY('Expression1', 'Expression2', 'Precision')
```

### Аргументы

'Expression1'	Входной сигнал, который содержит значения X нового сигнала
'Expression2'	Входной сигнал, который содержит значения Y нового сигнала
'Precision'	Опциональный параметр для указания частоты дискретизации выходного сигнала

### Описание

Данная функция используется, если результат должен использоваться из изображения X-Y для других операций. После успешного выбора функции присваиваются сигналы оси X и Y.

Обратите внимание, что в результирующей функции интервал между точками сигналов не равно интервалам исходных сигналов. Также и интервал между самими точками сигналов другой. При помощи параметра 'Precision' может быть настроен постоянный интервал между точками сигналов. Если параметр не указан, для всех последующих операций используется наименьший интервал точек сигналов как постоянное значение.

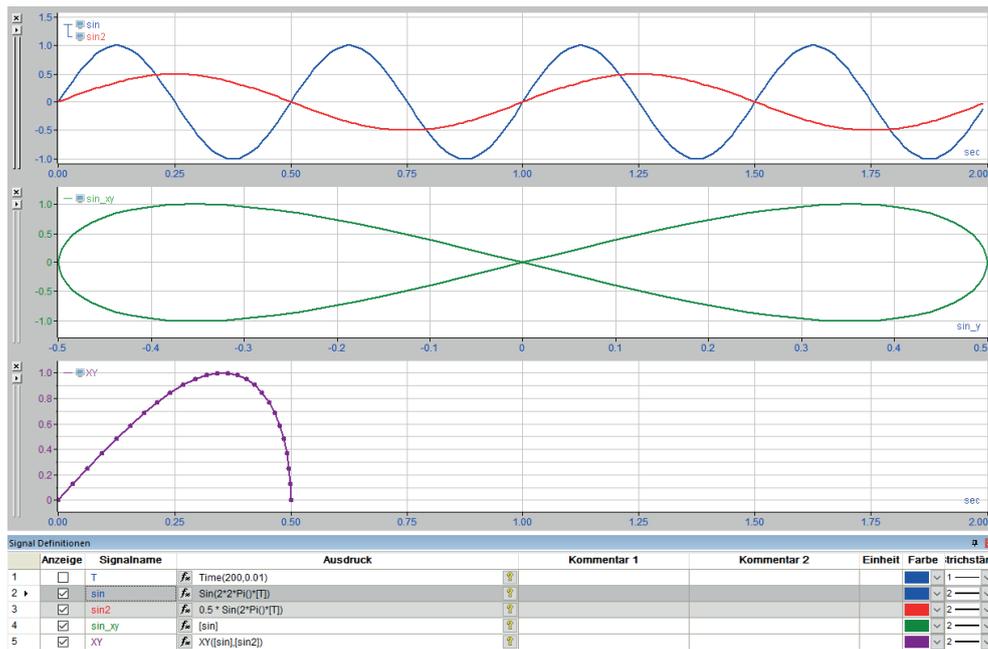


Рис. 24: Операции на оси X, XY

## 8.10 XMarker1 / XMarker2

XMarker1 () или XMarker2 ()

### Описание

Данная функция возвращает как результат позицию маркера X1 или X2 на оси X.

## 8.11 XBase / XOffset

### XBase

XBase ('Expression')

#### Описание

При помощи данной функции Вы можете определить опорное время записи или интервалы на базе длины или частоты между семплами. Функция возвращает при равноудаленно считанном сигнале расстояние между двумя точками измерения в единицах оси X.

Пр сигнале, семплы которого имеют не равное расстояние друг от друга, расстояние выводится в единицах оси X, которое было бы получено при передискретизации на равноудаленные семплы. По умолчанию это наименьший интервал между двумя семплами сигнала.

### XOffset

XOffset ('Expression')

#### Описание

Данная функция возвращает временной интервал первой точки измерения сигнала в начале файла измерения в секундах. Результат отрицательный, если сигнал начинается раньше и положительный, если он начинается позднее.

Если несколько файлов измерений открыты одновременно и активирована опция «Синхронизировать файлы измерений со временем записи», то смещение определяется не обязательно относительно начала файла измерений выбранного сигнала, а относительно начала файла измерений, имеющего самый ранний момент старта.

## 8.12 FillGaps

FillGaps ('Expression')

#### Описание

При помощи функции *FillGaps* пробелы в сигнале 'Expression' могут быть заполнены линейными интерполированными записями.

Данная функция особенно полезна, если при запросах тренда из базы данных из-за нулевых записей появляются пробелы.

## 8.13 XAlignFft

```
XAlignFft('Expression_fixed', 'Expression_shift', 'Start', 'End', 'MinScale', 'MaxScale', 'scale', 'type')
```

### Аргументы

'Expression_fixed'	Выражение, служащее референсным выражением
'Expression_shift'	Выражение, которое выравнивается по референсу
'Start'	
'End'	
'MinScale'	Наименьший коэффициент масштабирования x, подлежащий проверке.
'MaxScale'	Наибольший коэффициент масштабирования x, подлежащий проверке.
'scale'	
'type'	

### Описание

При помощи данной функции сигналы на базе длины с одним и тем же физическим значением, измеряемые в различных местах технологического процесса, могут быть выравнены относительно друг друга.

Некоторые параметры описаны ниже более подробно.

#### ■ 'Expression\_fixed'

Измерение толщины, рассматриваемое в ходе алгоритма как «постоянное», т.е. не масштабируемое или смещаемое. Это должно быть измерение толщины, содержащее профиль другого измерения. (В сравнении горячей/холодной прокатки - горячий прокат)

#### ■ 'Expression\_shift'

На данное измерение толщины ссылается затем результат выравнивания. Данное измерение должно быть масштабировано и смещено со результирующими значениями.

#### ■ 'Start'

Интервал с начала до конца указывает на участок оси X, в котором может быть смещено измерение 'Expression\_fixed'. Нулевая точка при этом является нулевой точкой выражения. Допустимы также отрицательные значения. Если измерение 'Expression\_shift' на левой стороне в *ibaAnalyzer* 10 единиц оси по сравнению с 'Expression\_fixed' могут выходить за пределы, то старт должен быть = -10.

#### ■ 'End'

Указывает конец описанного интервала. Рекомендуется выбрать данный конец в зависимости от длины 'Expression\_fixed'. Например, конец =  $XSize([Expression\_fixed])$  или конец =  $1.2 * Xsize([Expression\_fixed])$ , если допустим выступ в 20 процентов.

#### ■ 'scale'

При помощи данного параметра можно управлять отношением между точностью и скоростью. Чем меньше значение, тем медленнее и достовернее работает алгоритм. Чем выше

значение, тем больше ускоряется алгоритм за счет эвристики. При высоких значениях для 'scale' это может привести к некорректному результату. Для оптимального результата рекомендуется передавать разрешение измеренных данных. Если точки измерения имеют, например, интервал 10 см, то масштаб должен быть = 0,1. Если вычисление результата слишком длительное, значение можно скорректировать в большую сторону.

## 9 Операции с вектором

Операции с вектором расширяют возможности анализа для двумерных сигналов.

Векторы, обозначаемые в более ранних описаниях как массивы, могут образовываться различными способами:

- Посредством группировки нескольких сигналов в *ibaPDA* и обозначения группы как «Вектор»
- Посредством объединения нескольких сигналов в логических определениях сигналов в *ibaAnalyzer*
- Как результат различных функций вычислений, например , функции FFT

Векторы могут изображаться в *ibaAnalyzer* в горизонтальной проекции и в 3D-представлении. Векторные операции в редакторе выражений служат для использования векторных данных для дальнейших вычислений.

### Дополнительная документация



Подробные указания к данным изображениям и их настройкам см. в руководстве *ibaAnalyzer*, Часть 2, Типы изображения.

### 9.1 GetFirstIndex / GetLastIndex

`GetFirstIndex('Condition')` или `GetLastIndex('Condition')`

#### Описание

Данная функция возвращает индекс первого или последнего канала в векторе, для которого условие 'Condition' истина. Сам вектор должен при этом быть операндом условия 'Condition'. Если 'Condition' выдает ЛОЖЬ для всех каналов вектора, то функция возвращает -1 как результат.

### 9.2 GetRows

`GetRows('Vector', 'StartIndex', 'Counter', 'Step')`

#### Аргументы

'Vector'	Вектор, из которого должны считываться отдельные записи
'StartIndex'	Первый индекс, из которого должны считываться записи
'Counter'	Количество записей
'Step'	Размер шага

### Описание

Данная функция считывает ряды значений из вектора. При этом, исходя из 'StartIndex' (наименьший возможный индекс - 0) в с шагом размером 'Step' считывается общее число 'Counter' записей.

## 9.3 GetZoneCenters

`GetZoneCenters ('Vector')`

### Описание

Данная функция определяет положение центра зоны на оси Y для каждой зоны вектора. Единственным аргументом функции является вектор. Результатом снова является вектор с количеством значений в соответствии с количеством зон.

### Пример

Функция *GetZoneCenters* особенно полезна, если она применяется к результату функции *FftInTime*. Функция *FftInTime* возвращает как результат вектор с *n* "зонами", которые соответствуют частотным полосам (бинам). При помощи функции *GetZoneCenters* могут быть определены центральные частоты отдельных полос спектра и тем самым - вектор частоты. Таким образом возможно дифференцировать или интегрировать частотные домены, умножив или разделив соответственно результаты функции *FftInTime* и *GetZoneCenters*.

## 9.4 GetZoneOffset

`GetZoneOffset ('Vector')`

### Описание

Данная функция определяет смещение первой зоны, т.е. позицию центра зон первой зоны относительно нулевой линии оси Y. Единственным аргументом функции является вектор. Результатом является постоянное значение.

## 9.5 GetZoneWidths

`GetZoneWidths ('Vector')`

### Описание

Данная функция определяет ширину каждой зоны вектора в единицах оси Y. Единственным аргументом функции является вектор. Результатом снова является вектор с количеством значений в соответствии с количеством зон.

## 9.6 MakeVector

`MakeVector (r_0, r_1, ..., r_n)`

### Описание

Данная функция создает вектор с рядами значений от *r\_0* до *r\_n*. Аргументы могут быть постоянными значениями или сигналам и выражениями. Это сопоставимо с созданием вектора в диалоговом окне *Логических определений сигналов*.

**Пример**

Функция *MakeVector* служит, в основном, для того, чтобы позволить макросам выводить многомерные сигналы как результат. В редакторе макросов могут быть заявлены частичные результаты различных вычислений в пределах макросов как промежуточные значения. Как финальный результат макроса может быть задан вектор, аргументы которого являются промежуточные значения. Вектор используется как контейнер для результатов макроса, чтобы упростить интерфейсы макроса.

## 9.7 SetZoneWidths

```
SetZoneWidths('Vector', 'Widths', 'Offset')
```

**Аргументы**

'Vector'	Вектор со (измеренными) значениями вектора результатов
'Widths'	Вектор, который содержит в качестве значений ширину зон
'Offset'	Расстояние центра первой зоны от нулевой линии

**Описание**

Данная функция создает вектор с заданной шириной зон. При этом значения вектора результатов берутся из вектора 'Vector' и ширина зон - из вектора 'Widths'. Т.к. вектор с шириной зон может использовать, со своей стороны, выражения как аргументы, то при помощи данной функции создаются векторы с различной шириной зон в зависимости от загруженных данных. Выражения для определения ширины зон должны быть при этом постоянными в течение времени и не изменяться для уже загруженных данных. Если это не так, то значения ширины усредняются за весь период времени.

**Пример**

Функция *SetZoneWidths (MakeVector(1,2,3,2,1), MakeVector(2,4,10,4,2), -10)* создает аналогичный вектор, как и были созданы с логическими определениями сигналов, см. Руководство *ibaAnalyzer* Часть 2, раздел *Логические определения сигналов*.

## 9.8 VectorAvg

```
VectorAvg('Vector')
```

**Описание**

Данная функция вычисляет для каждого семпла среднее значение поперечного профиля, т. е. среднее значение всех путей векторов за момент времени или для каждой позиции на оси X. Как результат функция возвращает одномерный сигнал, который показывает график среднего значения поперечного профиля за время/длину векторного сигнала с таким же количеством семплов.

## 9.9 VectorKurtosis

`VectorKurtosis('Vector')`

### Описание

Данная функция вычисляет для каждого семпла куртозис (эксцесс) поперечного профиля. Как результат функция возвращает одномерный сигнал, который показывает график куртозиса поперечного профиля за время/длину векторного сигнала с таким же количеством семплов.

### Примечание



Обратите внимание, должно быть доступно как минимум 4 точки сигнала, т.е. вектор должен иметь как минимум 4 входа.

## 9.10 VectorMarkRange

`VectorMarkRange('Vector', 'Start', 'End')`

### Аргументы

'Vector'	Вектор с входными сигналами
'Start'	Начало выбираемого диапазона
'End'	Конец выбираемого диапазона

### Описание

Данная функция возвращает как результат часть вектора 'Vector' с шириной зоны с 'Start' (нижняя грань) до 'End' (верхняя грань).

Позиции указываются в единицах оси Y. Позиции могут быть как постоянными значениями, так и сигналами или выражениями и быть тем самым независимыми от загруженных данных.

Выражения для определения позиций должны быть при этом постоянными в течение времени и не изменяться для уже загруженных данных. Если это не так, то значения положения усредняются за весь период времени.

## 9.11 VectorMin / VectorMax

### VectorMax

`VectorMax('Vector')`

### Описание

Данная функция вычисляет для каждого семпла максимум поперечного профиля, т.е. максимальное значение всех векторов за момент времени или для каждой позиции оси X. Как результат функция возвращает одномерный сигнал, который показывает график максимума поперечного профиля за время/длину векторного сигнала с таким же количеством семплов.

**VectorMin**

```
VectorMin('Vector')
```

**Описание**

Данная функция вычисляет для каждого семпла минимум поперечного профиля, т. е. минимальное значение всех векторов за момент времени или для каждой позиции оси X. Как результат функция возвращает одномерный сигнал, который показывает график минимума поперечного профиля за время/длину векторного сигнала с таким же количеством семплов.

**9.12 VectorPercentile**

```
VectorPercentile('Vector', 'Percentile'=0.5)
```

**Аргументы**

'Vector'	Вектор с входными сигналами
'Percentile'	Процентиль между 0 и 1

**Описание**

Данная функция вычисляет для каждого семпла процентиль поперечного профиля.

Второй аргумент наряду с 'Vector' является величиной 'Percentile', которая должна вычисляться. Значение по умолчанию: 0,5 (медиана). Как результат функция возвращает одномерный сигнал, который показывает график максимума процентилей поперечного профиля за время/длину векторного сигнала с таким же количеством семплов.

**9.13 VectorSkewness**

```
VectorSkewness('Vector')
```

**Описание**

Данная функция вычисляет для каждого семпла асимметрию поперечного профиля. Как результат функция возвращает одномерный сигнал, который показывает график асимметрии поперечного профиля за время/длину векторного сигнала с таким же количеством семплов. Обратите внимание, что для вычисления должно быть доступно как минимум 4 точки сигналов.

**9.14 VectorStdDev**

```
VectorStdDev('Vector')
```

**Описание**

Данная функция вычисляет для каждого семпла стандартное отклонение поперечного профиля. Как результат функция возвращает одномерный сигнал, который показывает график стандартного отклонения за время/длину векторного сигнала с таким же количеством семплов.

## 9.15 VectorSum

```
VectorSum('Vector')
```

### Описание

Данная функция вычисляет для каждого семпла сумму всех значений поперечного профиля. Как результат функция возвращает одномерный сигнал, который показывает график суммы значений в поперечном профиле за время/длину векторного сигнала с таким же количеством семплов.

### Пример

Если разделить выражение *VectorSum* на количество векторных путей, то получится тот же результат, что и с функцией *VectorAvg*.

## 9.16 VectorToSignal / SignalToVector

### VectorToSignal

```
VectorToSignal('Vector', 'XBase')
```

### Аргументы

'Vector'	Вектор с (максимально постоянными) входными сигналами
'XBase'	Частота дискретизации выходного сигнала

### Описание

Данная функция создает вдоль поперечного профиля из элементов вектора одномерный сигнал. Каждый семпл результирующего сигнала соответствует элементу вектора. Результат соответствует поперечному профилю.

Параметр 'XBase' является опциональным. Если 'XBase' не указан, то используются ширина зон и смещение вектора. Результирующий сигнал может таким образом содержать и неравноудаленные семплы.

### Пример

В сочетании с функциями *YatX* и позицией маркера функция *VectorToSignal* может использоваться для того, чтобы выводить в любом месте в векторе поперечный профиль.

```
VectorToSignal(YatX([Vektor], XMarker1()))
```

### SignalToVector

```
SignalToVector('Signal')
```

### Описание

В отличие от функции *VectorToSignal* функция *SignalToVector* создает вектор с постоянными записями из сигнала 'Signal'. Ширина зоны и смещение задаются при этом частотой дискретизации входного сигнала. Обратите внимание, что данная функция в отличие от *VectorToSignal* не имеет опциональный аргумент, чтобы определять ширину зон. Для этого может использоваться функция *SetZoneWidth*.

**Примечание**

Входной сигнал должен иметь меньше 1000 семплов, в противном случае сигнал не обрабатывается и выделяется как слишком сложный.

## 9.17 Traverse / TraverseW

### Traverse

```
Traverse('Signal', 'Position', 'N'=40, 'Min', 'Max', 'Avg'=1)
```

#### Аргументы

'Signal'	Сигнал, измеренный перпендикулярно движущимся устройством
'Position'	Позиция движущегося перпендикулярно измерительного устройства вдоль поперечного профиля
'N'	Количество зон результирующего вектора
'Min'	Опциональный порог для минимума отображаемого диапазона
'Max'	Опциональный порог для максимума отображаемого диапазона
'Avg'	Опциональный, бинарный параметр для усреднения нескольких семплов в пределах пересечения зон, по умолчанию образуется среднее значение

#### Описание

Данная функция преобразует сигналы, поступающие от движущегося перпендикулярно измерительного устройства

### TraverseW

```
TraverseW('Signal', 'Position', 'Widths', 'Offset'=0, 'Avg'=1)
```

#### Аргументы

'Signal'	Сигнал, измеренный перпендикулярно движущимся измерительным устройством
'Position'	Позиция движущегося перпендикулярно измерительного устройства вдоль поперечного профиля
'Widths'	Ширина результирующих зон
'Offset'	Опциональное смещение первой зоны
'Avg'	Опциональный, бинарный параметр для усреднения нескольких семплов в пределах пересечения зон, по умолчанию образуется среднее значение

#### Описание

Данная функция функционирует аналогично *Traverse*, с тем отличием, что размеры результирующего вектора устанавливаются непосредственно за ширину зон 'Widths' и опциональный параметр смещения.

## 9.18 VectorPolynomial / VectorLSQPolyCoef

### VectorPolynomial

`VectorPolynomial('Coefs', 'Vector')`

#### Аргументы

'Coefs'	Коэффициенты интерполяционного многочлена; вычисляет с <i>VectorLSQPolyCoef</i>
'Vector'	Опорные точки для обработки интерполяционного многочлена

#### Описание

Данная функция может использоваться для изображения интерполяционного многочлена, который описывается коэффициентами 'Coefs' как результат функции *VectorLSQPolyCoef*.

Опорные точки для обработки многочлена задаются точками дискретизации вектора 'Vector'. Если заданы смещение зон и/или ширина зон, то они тоже используются, в противном случае знаки привлекаются как значения Y.

#### Примечание



Обратите внимание, что записи 'Vector' здесь роли не играют.

### VectorLSQPolyCoef

`VectorLSQPolyCoef('Vector', 'Degree')`

#### Аргументы

'Vector'	Вектор, для записей которого вычисляются многочлены среднеквадратичной аппроксимации
'Degree'	Степень интерполяционного многочлена

#### Описание

Данная функция - это расширение функции *LSQPolyCoef* на векторы. При этом вычисляются коэффициенты интерполяционного полинома степени 'Degree' при помощи метода наименьших квадратов для каждого поперечного профиля. В качестве базы для этого берутся экземпляры вектора, за исключением случаев, когда при создании вектора было задано смещение зон и/или ширина зон. В данном случае соответствующие значения используются как база.

## 10 Текстовые функции

### 10.1 InfofieldText / ChannelInfoFieldText / ModuleInfoFieldText

Данная функция позволяет предоставлять информацию из инфо-поля файла измерений, канала или модуля как текстовый канал.

#### Аргументы

'Index'	Индекс файла, в т.ч. канала или модуля
'InfoField'	Инфополе, подлежащее считыванию Должно быть указано в кавычках!
'Start'	Первый знак содержания поля, который должен быть считан (опция) Если значение не указывается, то считывается всё содержание
'End'	Последний знак содержимого поля, который должен быть считан (опция) Если значение не указывается, то считывание осуществляется со 'Start' до последнего знака.

#### Примечание



В функции *ModuleInfoFieldText* необходимо указать два знака. Индекс файла измерения как первый аргумент и индекс модуля - как второй. Все остальные аргументы остаются такими же.

#### InfoFieldText

```
InfofieldText('FileIndex', 'InfoField', 'Start', 'End')
```

#### Описание

Данная функция выводит содержимое инфо-поля как текстовый канал.

#### Совет



Если щелкнуть дважды по желаемому инфо-полю, то *ibaAnalyzer* вставит соответствующую функцию автоматически как новый сигнал в таблицу сигналов. Затем нужно только скорректировать, если требуется, имя сигнала и начало/конец.

Если Вы хотите считывать содержимое инфо-поля как числовое значение, используйте функцию *InfoField*.

**ChannelInfoFieldText**

```
ChannelInfoFieldText('ChannelIndex', 'InfoField', 'Start', 'End')
```

**Описание**

Данная функция выводит содержимое инфо-поля канала текста.

**ModuleInfoFieldText**

```
ModuleInfoFieldText('FileIndex', 'ModuleIndex', 'InfoField', 'Start', 'End')
```

**Аргументы**

'FileIndex'	Индекс файла, к которому относится модуль
'ModuleIndex'	Индекс модуля
'InfoField'	Инфо-поле, которое должно считываться из модуля
'Start'	Индекс старта инфо-строки
'End'	Индекс конца инфо-строки

**Описание**

Данная функция работает как функции *InfoFieldText* и *ChannelInfoFieldText*, но она относится к инфо-полям модуля, а не файла измерений или сигнала. Функция возвращает как результат текстовый канал с содержимым указанного инфо-поля.

**10.2 TextCompare / CompareText**

```
TextCompare('Text1', 'Text2', 'CaseSensitive=True')
```

**Аргументы**

'Text1/2'	Две сравниваемые строки
'CaseSensitive'	Опциональный параметр, при помощи которого Вы можете задать, будут ли учитываться при сравнении верхний и нижний регистр.

**Описание**

Данная функция позволяет выполнять лексикографическое сравнение текстовой информации друг с другом. Функция работает как с содержанием текстовых каналов, так и с чистым текстом, который - сопровождаемый кавычками - вносится непосредственно в определение сигнала.

Сравнение и результаты:

- Результат -1, если информация первого текста должна располагаться лексикографически перед информацией второго текста.
- Результат 0, если оба текста имеют одинаковую информацию.
- Результат +1, если информация первого текста лексикографически должна располагаться после информации второго текста.

**Пример**

Влияние параметра 'CaseSensitive' показывает следующая таблица:

Text1	Text2	Результат		Примечание
		TextCompare	TextCompare	
1234 abcd	1234 abcd	0	0	1 = 2
1234 abcd	1234 bcde	-1	-1	1 < 2 "a" стоит перед "b"
1234 Abcd	1234 abcd	0	1	1 = 2 (без учета верхне- го/нижнего реги- стра) 1 > 2 (с учетом верхне- го/нижнего реги- стра) "A" стоит после "a"
12340 abcd	1234 abcd	1	1	1 > 2 "0" стоит после " "
1234 0abcd	1234 abcd	-1	-1	1 < 2 "0" стоит перед "a"
12034 abcd	1234 abcd	-1	-1	1 < 2 "0" стоит перед "3"
1234 abcd	1y34 abcd	-1	-1	1 < 2 "2" стоит перед "y"
1z34 abcd	1Y34 abcd	1	1	1 > 1 "z" стоит после "Y"

Table 7: Результаты функции TextCompare (примеры)

## 10.3 ToText / FromText

### ToText

```
ToText ('Expression', 'Format'='%g', 'datatype'=0)
```

#### Аргументы

'Expression'	Выражение, содержимое которого должно быть преобразовано в текстовый канал
'Format'	Опциональный параметр для строки формата
'datatype'	Опциональный параметр, который задает формат с плавающей запятой - 'datatype'=0 значение по умолчанию - 'datatype'=1 вывод форматируется как 16-разрядное целое со знаком - 'datatype'=2 вывод форматируется как 16-разрядное целое без знака - 'datatype'=3 вывод форматируется как 32-разрядное целое со знаком - 'datatype'=4 вывод форматируется как 16-разрядное целое без знака

#### Описание

Данная функция преобразует числовое значение сигнала в текстовый канал. При этом при равноудаленных семплах входного сигнала 'Expression' и сохраняемом значении Y только значение первой точки сигнала вносится и отображается как семпл в текстовом канале. Если значение Y изменяется, для каждого нового значения Y вносится и отображается один семпл в текстовом канале.

Если входной сигнал 'Expression' не содержит равноудаленные семплы, то для каждого входного семпла вносится и отображается один семпл в текстовом канале.

Опциональный параметр 'Format' необходимо задавать в соответствии с синтаксисом C printf. Вы можете указать только один параметр (%), который должен соответствовать 32-разрядному значению с плавающей запятой IEEE. Значение по умолчанию - %g. Данное значение используется также, если Вы не указываете опциональный параметр.

Примеры:

%g = преобразование значения с плавающей точкой в текст

%.4f = текст/число с 4 знаками после запятой, и т.д.

#### Пример

Функция *ToText* может быть, например, полезна, если визуализируются тренды, содержащие большое количество данных. Без постоянного переключения представлений маркеров и сигналов могут быть просто отображены числовые значения.

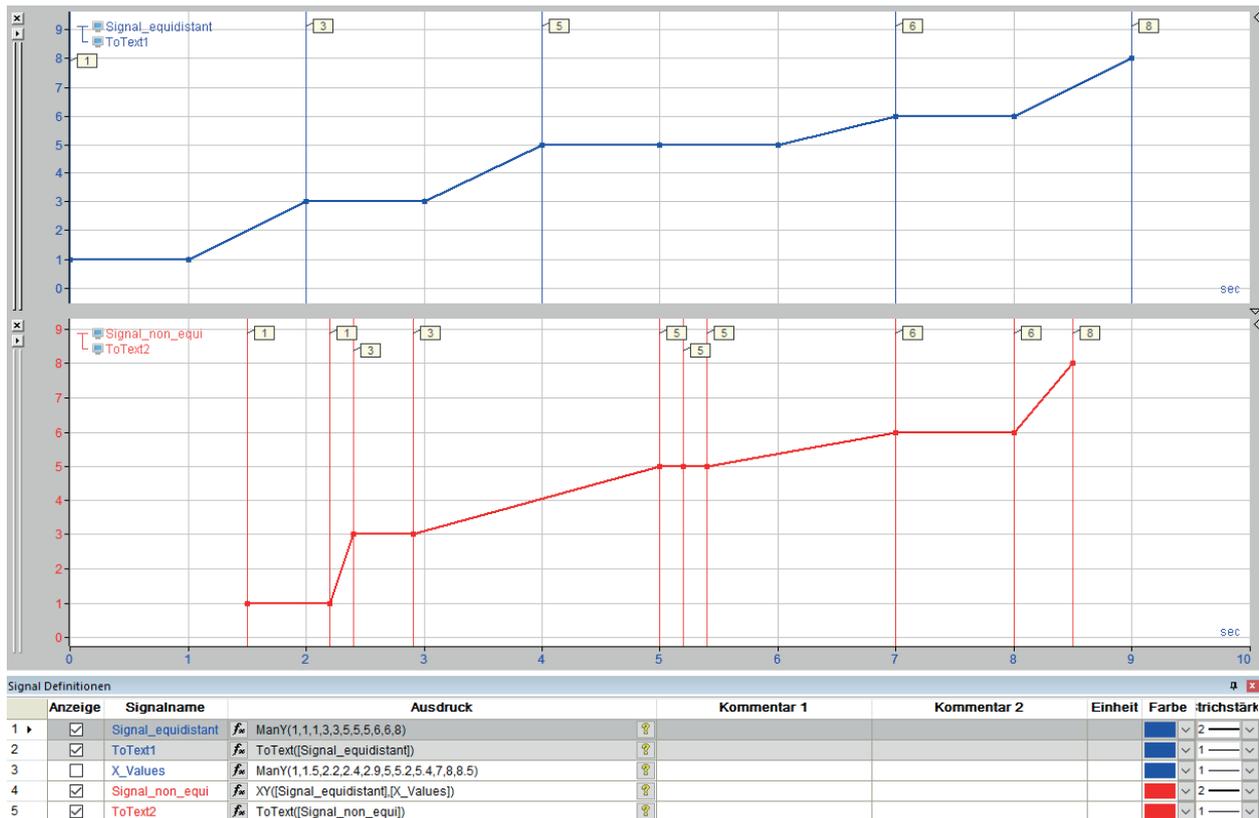


Рис. 25: Применение функции ToText

### FromText

FromText('TextChannel', 'Start', 'End')

### Аргументы

'TextChannel'	Текстовый канал, подлежащий преобразованию
'Start'	Первый знак содержания поля, который должен быть считан (опция) Если значение не указывается, то считывается всё содержание
'End'	Последний знак содержимого поля, который должен быть считан (опция) Если значение не указывается, то считывание осуществляется со 'Start' до последнего знака.

### Описание

Данная функция конвертирует содержимое текстового канала ,TextChannel' в числовое значение. Параметры ,Start' и ,End' могут использоваться в качестве опции как знаки, чтобы не преобразовывать всю строку. По умолчанию используются ,Start'=0 и ,End'=длина строки.

## 10.4 TrimText

```
TrimText('Text', 'RemoveOption=0')
```

### Аргументы

'Text'	Текстовый канал или выражение, из которого должны быть удалены пробелы.
'RemoveOption'	Параметры для настройки режима работы: 0 (по умолчанию): Удалить пробелы перед и после текста 1: Удалить только пробелы перед текстом 2: Удалить пробелы после текста 3: Удалить все пробелы, также в тексте

### Описание

При помощи данной функции Вы можете удалить из текста пробелы. Функция может применяться как к текстовым каналам, уже содержащимся в файле измерений, так и к результатам функций *InfofieldText* и *ToText*.

## 10.5 ConcatText

```
ConcatText('Text1', 'Text2', ...)
```

### Описание

Данная функция может использоваться для объединения несколько текстовых каналов. При этом допустимы также числовые сигналы. Они автоматически преобразуются в текст.

Если позиции X отдельных каналов не совпадают, для каждой существующей позиции X создается новая запись и отсутствующая запись заменяется левой соседней, если есть.

### Пример

Для последовательно расположенных сигналов доступен идентификатор файла в виде канала технотроки. Для конечных продуктов есть счетчик продуктов. Как результат должны быть вместе расположены обе части информации.

## 10.6 CharValue

```
CharValue('Text', 'CharNumber'=0)
```

### Аргументы

'Text'	Текст ввода
'CharNumber'	Положение, начиная с нуля, определяемого знака.

### Описание

Данная функция возвращает как результат значение ASCII знака в положении 'CharNumber' в тексте 'Text'. По умолчанию используется первый знак, имеющий как 'CharNumber' нуль.

## 10.7 CountText / TextLength

```
CountText('Text', 'CountOnlyDifferent'=0, 'Reset'=0)
```

**Аргументы**

'Text'	Текст ввода
	Опциональный параметр для подсчета текстов, которые отличаются от предыдущего фрагмента текста.
'Reset'	Опциональный параметр: для сброса счетчика

**Описание**

Данная функция возвращает как результат количество текстов в пределах текстового канала 'Text'. Если параметр 'CountOnlyDifferent' установлен на Истина, считаются только тексты, которые отклоняются от предыдущего фрагмента текста. При помощи цифрового сигнала 'Reset' счетчик может быть сброшен на нуль.

```
TextLength ('Text')
```

**Описание**

Данная функция возвращает как результат количество знаков в 'Text'.

**10.8 DeleteText / InsertText / ReplaceText**

```
DeleteText ('Text', 'StartPos', 'Length')
```

**Аргументы**

'Text'	Текст ввода
'StartPos'	Начинающийся с нуля указатель первого удаленного знака
'Length'	Количество подлежащих удалению знаков

**Описание**

Данная функция удаляет количество знаков 'Length' из текста 'Text'. Первый удаляемый знак, находится в положении 'StartPos' в тексте.

```
InsertText ('Text1', 'Text2', 'Pos')
```

**Аргументы**

'Text1'	Исходный текст
'Text2'	Текст, вставляемый в исходный текст
'Pos'	Позиция знаков, начинающаяся с нуля, в которой вставляется 'Text2'

**Описание**

Данная функция вставляет 'Text2' в 'Text1' в положении знаков, начинающегося с нуля 'Pos'. Если 'Pos' меньше или равно нулю, 'Text2' стоит перед 'Text1'. Если 'Pos' больше или равна длине 'Text1', 'Text2' прикрепляется 'Text1'.

```
ReplaceText ('Text', 'SearchText', 'ReplaceText')
```

**Аргументы**

'Text'	Текст ввода
--------	-------------

'SearchText'	Текст, который должен быть заменен текстом 'Text'
'ReplaceText'	Если обнаруживается 'SearchText', он заменяется текстом 'ReplaceText'

**Описание**

Данная функция заменяет все совпадения с 'SearchText' в пределах 'Text' текстом 'ReplaceText'.

**10.9 MidText / FindText**

```
MidText ('Text', 'StartPos', 'Length')
```

**Аргументы**

'Text'	Текст ввода
'StartPos'	Индекс на базе нуля первого извлекаемого знака из текста
'Length'	Количество знаков, которые должны быть извлечены из текста

**Описание**

Данная функция возвращает как результат количество знаков ('Length') из текста 'Text'. Результирующая последовательность знаков начинается со знака в положении на базе нуля 'StartPos'.

```
FindText ('Text', 'SearchText', 'CaseSensitive'=0, 'N'=1)
```

**Аргументы**

'Text'	Текст ввода
'SearchText'	Текст, который должен быть идентифицирован в тексте ввода
'CaseSensitive'	Опциональный параметр для активации поиска текста с учетом регистра
'N'	Параметр для поиска не первого, а n-ого совпадения текста поиска 'SearchText'.

**Описание**

Данная функция возвращает как результат индекс, начинающийся с нуля, положения 'N'-ого совпадения текста поиска 'SearchText' в тексте 'Text'. Если параметр 'CaseSensitive' установлен на Истина, то выполняется поиск с учетом регистра. Если 'SearchText' не найден, то результат -1.

# 11 Различные функции

## 11.1 Debounce

```
Debounce('Expression', 'Interval')
```

### Аргументы

'Expression'	Входной сигнал, в котором должен быть устранен дребезг
'Interval'	Время нечувствительности

### Описание

Данная функция возвращает как результат график сигнала с отфильтрованным дребезгом выражения 'Expression' с 'Interval' как время нечувствительности в [с]. В случае сигналов на базе длины 'Interval' интерпретируется как путь в [м].

Функция работает аналогично реле времени с фиксированной задержкой, с разницей лишь в том, что переход сигнала с ИСТИНА на ЛОЖЬ (задний фронт) изображается с корректным временем, т.е. без задержек, если в пределах настроенного времени не происходит опять переход с ЛОЖЬ на ИСТИНА (передний фронт).

Таким образом возможно выравнять неустойчивые сигналы, например, фотоячеек или конечных выключателей. Это особенно важно, если данные сигналы используются как условия в таких операциях, как *XMarkValid* или *XCutValid*, т.к. при каждом пропуске прерывается вычисление операции, и тем самым могут потеряться результативные значения. Отличие четко видно на следующем изображении.

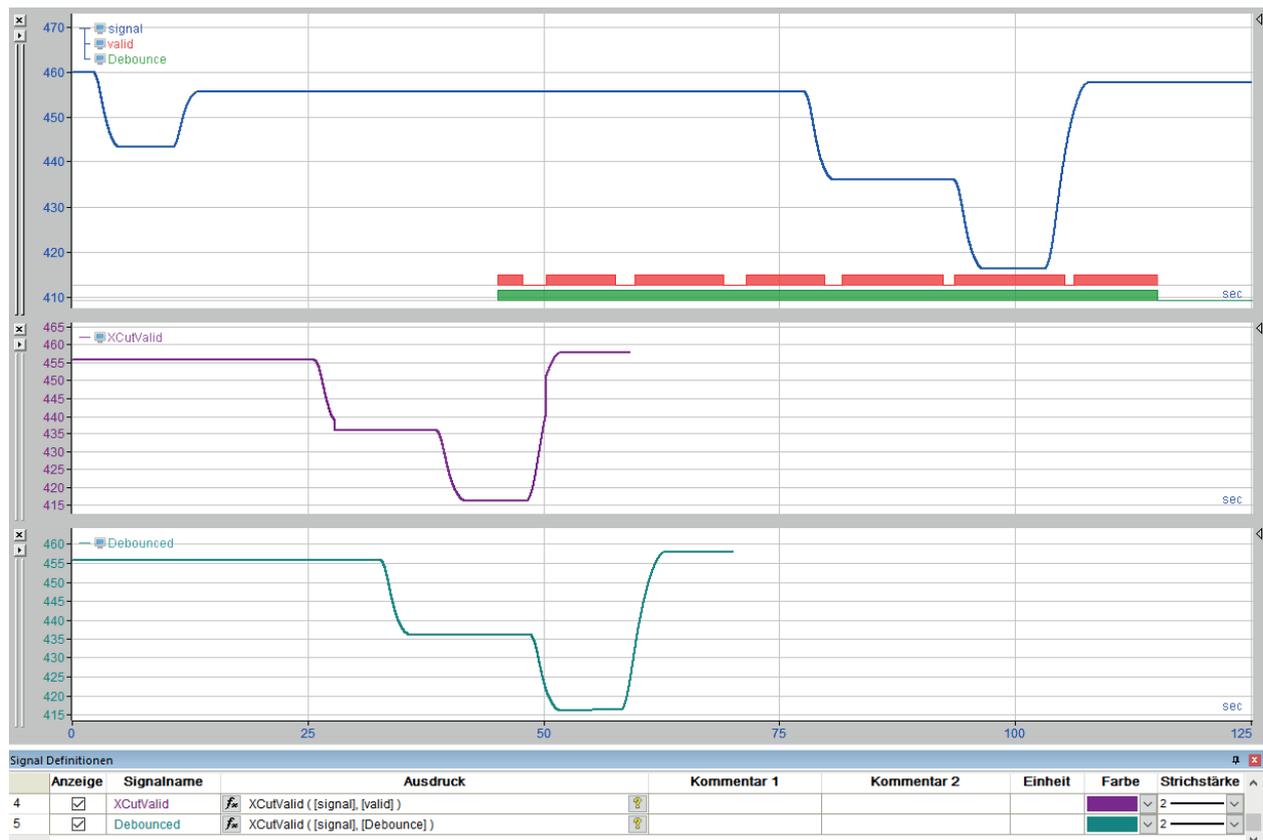


Рис. 26: Различные функции, фильтр дребезга

## 11.2 Envelope

Envelope('Expression', 'Interval')

### Аргументы

'Expression'	Выражение, для которого образуется огибающая кривая
'Interval'	Оси X, интервал

### Описание

Данная функция вычисляет верхнюю огибающую кривую. Огибающая кривая образуется при соединении максимумов. Качество огибающей кривой зависит от параметра 'Interval'. Без указания данного параметра учитываются только наибольшие максимумы за период записи. При помощи параметра 'Interval' задается длина интервала в единицах оси X (с, м, Гц, 1/м). Тогда учитываются пики в пределах данного интервала, и огибающая кривая прилегает сильнее к графику сигналов.

---

**Совет**

Чтобы получить огибающую кривую в нижней части графика сигналов, Вы можете использовать ту же функцию в виде

```
-Envelope ('Expression', 'Interval')
```

. Таким образом соединяются друг с другом минимумы.

---

## 11.3 False / True

False() bzw. True()

### Описание

Данные операнды возвращают постоянное значение 0 или 1.

В булевых операциях (И, ИЛИ и т.д.) значение интерпретируется как логический 0 (ложь) или логическая 1 (истина). В арифметических операциях и в сочетании с аналоговыми значениями значение интерпретируется как 0,0 или 1,1 («постоянный нуль» или «постоянная единица»).

## 11.4 GetBit / GetBitMask

### Getbit

```
Getbit('Expression', 'Bitnumber')
```

#### Описание

Данная функция возвращает как результат булево значение бита 'BitNo' выражения 'Expression' после округления 'Expression' до ближайшего целого значения. Округление осуществляется соответственно с шагом 0,5. (2,48 --> 2; 2,50 -->3). Допустимая последовательность битов: от 0 (LSB) до 15 (MSB).

#### Важно



Целые значения с 64 битами не могут обрабатываться при помощи данной функции, т.к. они не поддерживаются программой *ibaPDA* и тем самым не могут входить в файл измерений.

#### Пример

В таблице ниже в качестве примера представлены самые младшие байты целого значения с битами от 0 до 7. Для указания значений от 0 до 8 отдельные биты отмечены знаком X. (X = ИСТИНА)

Номер бита	7	6	5	4	3	2	1	0
0								
1								X
2							X	
3							X	X
4						X		
5						X		X
6						X	X	
7						X	X	X
8					X			

Table 8: Примеры значений битов (X=ИСТИНА)

#### Совет



Если нужно разложить на отдельные биты несколько целых значений 8-, 16- или 32-разрядные, то можно существенно облегчить себе работу, щелкнув в дереве сигналов правой кнопкой мыши по желаемому сигналу и выбрав в контекстном меню «Показывать биты». Все биты сразу же отобразятся в виде отдельных цифровых сигналов, без необходимости программировать функцию *Getbit*. Применяется аналогичный механизм, как и в *Getbit*.

## GetBitMask

```
GetBitMask('Expression', 'Bitnumber')
```

### Описание

Данная функция интерпретирует 'Expression' как битовую маску значения float и выдает как результат значение бита 'BitNumber'. Допустимый диапазон: от 0 (LSB) до 31 (MSB)

Данная функция была специально разработана для работы с данными SimadynD в специальном случае применения, когда в пакете записываются до 32 цифровых значений как переменные с плавающей запятой. Функция *GetBitMask* обрабатывает только валентность указанного бита 'BitNo', независимо от того, является ли он частью мантиссы или порядка. В отличие от функции *GetBit* округление до целого значения не проводится.

---

### Совет



Если нужно разложить на отдельные биты несколько 32-разрядных значений с плавающей запятой, то можно существенно облегчить себе работу, щелкнув в дереве сигналов правой кнопкой мыши по желаемому сигналу и выбрав в контекстном меню «Показывать биты». Все биты сразу же отобразятся в виде отдельных цифровых сигналов. Применяется аналогичный механизм, как и в *GetBitMask*.

---

## 11.5 HighPrecision

```
HighPrecision('Expression')
```

### Описание

При помощи данной функции выражение 'Expression' обозначается как величина с удвоенной точностью. Вычисления, выполняемые тогда с 'Expression', осуществляются с удвоенной точностью, даже если исходное выражение имеет только простую точность.

Удвоенная точность имеет преимущество в том, что вычисления могут выполняться точнее, но и недостаток в том, что задействуется вдвое больше памяти. Поэтому *ibaAnalyzer* решает автоматически на базе входных аргументов для какого-либо вычисления, какая точность будет использоваться.

## 11.6 InfoField / ChannelInfoField / ModuleInfoField

Данная функция позволяет считывать информацию из инфо-поля файла измерений, канала или модуля.

### Примечание



Функции *InfoField*, *ChannelInfoField* und *ModuleInfoField* ожидают числовое значение. Если требуется считать текст, нужно использовать функции *InfoFieldText*, *ChannelInfoFieldText* и *ModuleInfoFieldText*, см. ↗ [InfofieldText / ChannelInfoFieldText / ModuleInfoFieldText](#), страница 77.

### Аргументы

'Index'	Индекс файла, в т.ч. канала или модуля
'InfoField'	Инфополе, подлежащее считыванию Должно быть указано в кавычках!
'Start'	Первый знак содержания поля, который должен быть считан (опция) Если значение не указывается, то считывается всё содержание
'End'	Последний знак содержимого поля, который должен быть считан (опция) Если значение не указывается, то считывание осуществляется до последнего знака.

### Примечание



В функции *ModuleInfoField* необходимо указать два знака. Индекс файла измерения как первый аргумент и индекс модуля - как второй. Все остальные аргументы остаются такими же.

### InfoField

```
Infofield('FileIndex' , "'InfoField'", 'Start', 'End')
```

### Совет



Если щелкнуть дважды по желаемому инфо-полю, то *ibaAnalyzer* вставит соответствующую функцию автоматически как новый сигнал в таблицу сигналов. Затем нужно только скорректировать, если требуется, имя сигнала и начало/конец. Данный метод работает также в поле ввода редактора выражений. Функция тогда вставляется в месте курсора.

Если Вы хотите считывать содержимое инфо-поля как текстовый канал, используйте функцию *ChannelInfofieldText*.

**ChannelInfoField**

```
ChannelInfofield('ChannelIndex','InfoField','Start','End')
```

**ModuleInfoField**

```
ModuleInfoField('FileIndex','ModuleIndex','InfoField','Start','End')
```

---

**Примечание**

В данной функции необходимо указать два знака. Индекс файла измерения как первый аргумент и индекс модуля - как второй.

---

## 11.7 LimitAlarm

```
LimitAlarm('Expression', 'Limit', 'DeadBand', 'Time')
```

### Аргументы

'Expression'	Измеренное значение
'Limit'	Предельное значение, с которого функция возвращает ИСТИНА
'DeadBand'	Указание зоны нечувствительности ниже верхнего предельного значения, в пределах которого функция не сбрасывается на ЛОЖЬ.
'Time'	Время, в течение которого измеренное значение должно быть выше предельного значения, пока функция станет ИСТИНОЙ

### Описание

Данная функция контролирует измеренное значение ('Expression') и устанавливает результат на ИСТИНА, если измеренное значение дольше указанного времени ('Time') выше предельного значения ('Limit'). Результат функции будет снова ЛОЖЬ, если измеренное значение не достигает предельное значение на указанное в зоне нечувствительности ('DeadBand') значение.

### Совет



Функция *LimitAlarm* может также использоваться для нижнего предельного значения. Для этого должны зеркалироваться только измеренное значение и предельное значение, т.е. умножаться на (-1).

Например, `LimitAlarm([0:1] *(-1), 9 *(-1), 0.5, 0.4)`

## 11.8 ManY

```
ManY('Xbase', 'y0', 'y1', ....)
```

### Аргументы

'XBase'	Частота дискретизации выходного сигнала
'y0', 'y1', ...	Значения Y выходного сигнала

### Описание

При помощи данной функции можно вручную создавать график сигналов с "измеренными значениями" 'y0'...'y99', которые имеют соответственно интервал времени или пути 'Xbase'. 'Xbase' указывается в [с] для значений на базе времени и в [м] для значений на базе длины. Количество точек ограничено до 100.

Вы можете указать, например, референсные графики, с которыми будут сравниваться реально измеренные сигналы. Или будет в анализ будут добавляться данные, которые не являются измеренными значениями. Кроме того, при помощи данной функции могут быть созданы вручную текстовые каналы и заполнены различными значениями.

### Совет



Если Вы зададите параметры от y0 до макс. Y99 в кавычках, введенные знаки будут применяться не как числовые значения, а как знаки ASCII.

## 11.9 Rand (СЛЧИС)

`Rand('Count', 'Xbase')`

### Описание

Данная функция создает сигнал, состоящий из случайных чисел в диапазоне от 0 до 32767 для 'Count' точек в 'Xbase' [с] (на базе времени) или [м] (на базе длины). На рисунке представлены три сигнала длительностью 100 с, но состоящие из различного количества точек. опорное время 'Xbase' составляет 1 с, 100 мс и 10 мс.

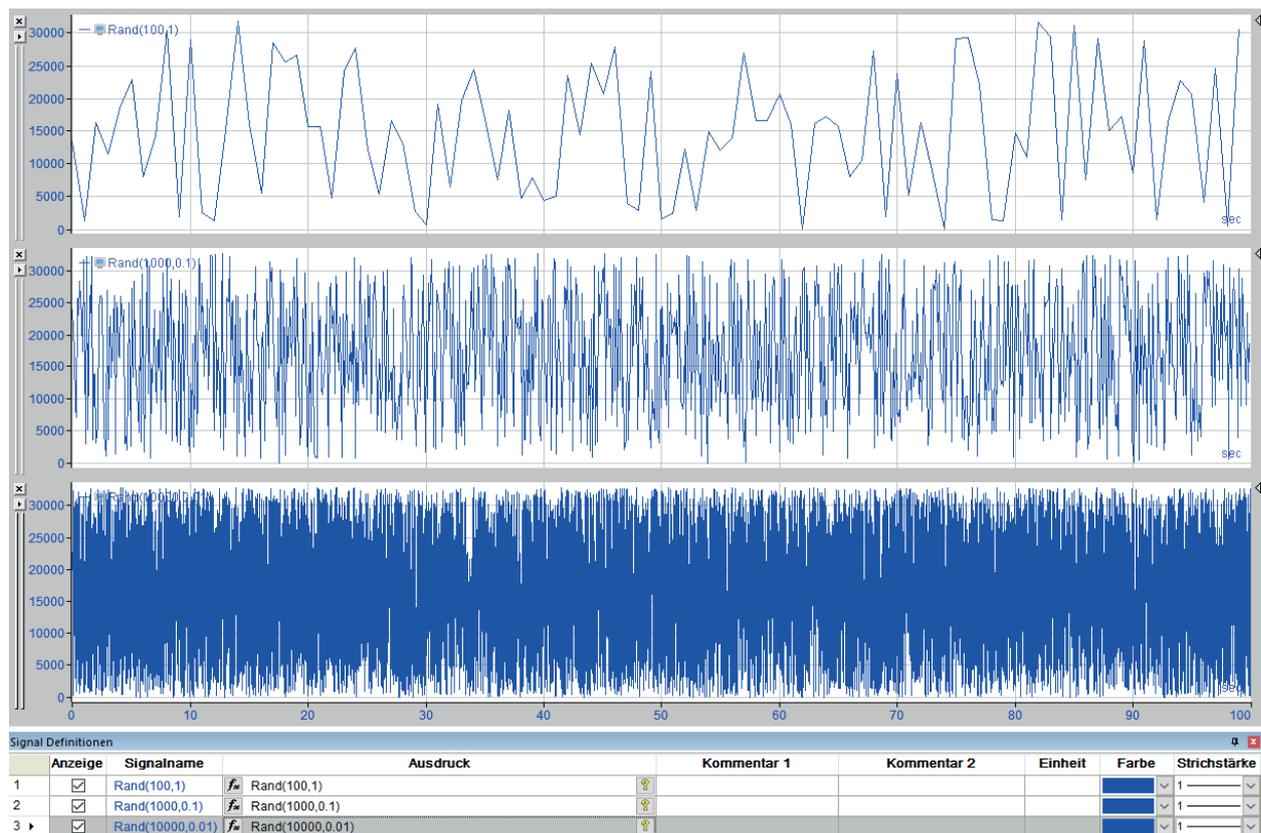


Рис. 27: Различные функции Rand

## 11.10 Sign

`Sign('Expression')`

### Описание

Данная функция возвращает как результат знак выражения 'Expression':

'Expression' > 0 --> +1

'Expression' = 0 --> 0

'Expression' < 0 --> -1

## 11.11 Технострока

```
Technostring('Index', 'Begin', 'End')
```

### Аргументы

'Index'	Индекс файла измерений
'Begin'	Начало считываемого диапазона
'End'	Конец считываемого диапазона

### Описание

Данная функция извлекает строку из индекса файла измерений 'Index' между 'Begin' и 'End'. Индекс старта по умолчанию - 0. Таким образом информация из техностроки интерпретируется как сигнал (только числовые знаки).

Обрабатывается информация техностроки, которая отображается в ветви информации в окне дерева сигналов. Для этого информация технострок *ibaPDA* должна быть сохранена в файле измерений.

'Begin' и 'End' соответствуют положению знаков в техностроке, которые ограничивают желаемый диапазон, обрабатываемый как сигнал. Могут обрабатываться только числовые знаки. Ведущие нули игнорируются.

Указание 'Index' требуется только тогда, когда одновременно открываются несколько файлов измерения. Файл в самом верху в окне дерева сигналов имеет индекс 0. Все остальные файлы тогда сверху вниз 1, 2 и т.д. Если открыт только один файл, то индекс должен быть всегда = 0.

## 11.12 WindowAlarm

```
WindowAlarm('Expression', 'Limit1', 'DeadBand1', 'Limit2', 'DeadBand2', 'Time')
```

### Аргументы

'Expression'	Измеренное значение
'Limit1'	Верхнее предельное значение, с которого функция возвращает ИСТИНА
'DeadBand1'	Указание зоны нечувствительности ниже верхнего предельного значения ('Limit1'), в пределах которого функция не сбрасывается на ЛОЖЬ.
'Limit2'	Нижнее предельное значение, с которого функция возвращает ИСТИНА
'DeadBand2'	Указание зоны нечувствительности выше нижнего предельного значения ('Limit1'), в пределах которого функция не сбрасывается на ЛОЖЬ.
'Time'	Время, в течение которого измеренное значение должно быть больше верхнего предельного значения и меньше нижнего предельного значения

### Описание

Данная функция контролирует измеренное значение ('Expression') и устанавливает результат на ИСТИНА, если измеренное значение больше указанного времени ('Time') находится за пределами диапазона между верхним предельным значением ('Limit1') и нижним предельным значением ('Limit2'). Результат функции становится снова ЛОЖЬ, если измеренное значение не достигает верхнее предельное значение на указанное в зоне нечув-

ствительности1 ('DeadBand1') значение, или превышает нижнее предельное значение на указанное в зоне нечувствительности2 ('DeadBand2') значение.

## 11.13 YatX / SetYatX

### YatX

```
YatX('Expression', 'X', 'Continuous'=0)
```

#### Аргументы

'Expression'	Входной сигнал
'X'	Позиция, в котором должно считываться значение
'Continuous'	Опциональный параметр для допуска переменных значений 'X'

#### Описание

Данная функция возвращает как результат значение Y выражения 'Expression' в положении 'X' на оси X. Функция может быть применена как к сигналам на базе времени, так и на базе длины.

В стандартном режиме, т.е. если параметр 'Continuous' не указан, или ЛОЖЬ или 0, то функция ожидает постоянное значение X и возвращает постоянное значение Y как результат.

Параметр 'X' может быть также переменной, т.е. он сам может быть функцией. В данном случае нужно активировать непрерывный режим, установив параметр 'Continuous' на ИСТИНА или 1. Функция определяет тогда для каждого значения 'X' подходящее значение Y.

### SetYatX

```
SetYatX('Expression', 'Value', 'XPos')
```

#### Аргументы

'Expression'	Выражение, которое должно быть изменено
'Value'	Значение, которое должно быть добавлено в 'XPos'
'XPos'	Положение X, в котором должно быть добавлено значение 'Value'

#### Описание

Функция *SetYatX* позволяет создавать копию сигнала, в которой было изменено значение. Она возвращает как результат копию сигнала 'Expression', при котором в точке 'Xpos' было добавлено значение 'Value'.

В зависимости от наличия равноудаленно выборочного сигнала функция ведет себя по-разному. В случае равноудаленных сигналов различают следующие случаи:

- Если 'XPos' меньше смещения сигнала, сигнал возвращается без изменений.
- Если 'XPos' соответствует размеру сигнала (см. XSize) плюс величине дискретизации, сигнал удлиняется на семпл со значением 'Value'.
- Во всех других случаях вставляется новой значение в точке 'Xpos' или в следующей меньшей точке семпла.

Для неравноудаленно считанных сигналов функция заменяет значение в точке 'XPos', если есть, или вставляет новый семпл.

#### Примечание



Функция может также использоваться для добавления текста.

## 11.14 PulseFreq

```
PulseFreq('Expression', 'Omega'=0, 'EdgeType'=2, 'MinFreq'=0.05)
```

#### Аргументы

'Expression'	Сигнал счетчика импульсов	
'Omega'	Частота фильтра	
'EdgeType'	Считаемый тип фронта	
	'EdgeType' = -1	Только задние фронты
	'EdgeType' = 0	Передний и задний фронты
	'EdgeType' = 1	Только передние фронты
	'EdgeType' = 2	'Expression' является счетчиком импульсов
'MinFreq'	Наименьшая изображаемая частота	

#### Описание

Данная функция вычисляет частоту импульсов или счетчиков импульсов 'Expression'. Единица измерения результата - импульсы/с или Гц.

Фильтр нижних частот с предельной угловой скоростью 'Omega' применяется к результату. Если 'Omega' 0, то фильтр нижних частот деактивирован. 'EdgeType' определяет, какие фронты импульсов должны учитываться. Как вычисленная частота возвращается нуль, если за 1000 считывания не появляется ни одного импульса.

Данная функция создана специально для анализа инкрементального датчика WAGO 750-631. Данная функция может использоваться для вычисления скорости по состоянию счетчика импульсов. Состояние счетчика импульсов дифференцируется с учетом возможного переполнения. Т.к. результат данной дифференциации может быть с частотами помех или шумами, к нему применяется фильтр нижних частот. Настраиваемая частота фильтров должна быть немного выше максимальной частоты импульсов.

## 12 Функции фильтра

### 12.1 LP

`Lp('Expression', 'Omega')`

#### Аргументы

'Expression'	Измеренное значение
'Omega'	Угловая частота для фильтра нижних частот

#### Описание

Данная функция является цифровым фильтром нижних частот первого порядка с угловой частотой 'Omega'. Применимо к сигналу 'Expression' она возвращает как результат сигнал, который содержит только переменные составляющие с частотами меньше 'Omega'.

#### Примечание



Цифровые фильтры, созданные при помощи редактора фильтров, могут быть сохранены в системе и доступны также в редакторе выражений как функции фильтров.

### 12.2 PreWhiten

`PreWhiten('Expression', 'Order')`

#### Аргументы

'Expression'	Измеренное значение, подлежащее фильтрации
'Order'	Порядок фильтра FIR

#### Описание

Данная функция применяет фильтр FIR с коэффициентами, которые задаются при помощи уравнения Юля Уокера. Речь идет о фильтре высоких частот, который оставляет только белый шум и импульсные компоненты сигнала.

## 13 Технологические функции

### 13.1 ChebyCoef

```
ChebyCoef('Vector','begin_seg','end_seg','Order','CoverFactor'=1)
```

#### Аргументы

'Vector'	Измеренные значения, подлежащие аппроксимации
'begin_seg'	Первый используемый сегмент вектора
'end_seg'	Последний используемый сегмент вектора
'Order'	Порядок многочлена Чебышева
	Опциональный аргумент для определения коэффициента наложения

#### Описание

Функция *ChebyCoef* вычисляет коэффициент многочлена Chebyshev порядка 'Order' по поперечному профилю вектора 'Vector'. При этом учитываются только записи вектора между сегментами 'begin\_seg' и 'end\_seg'. Опциональный коэффициент наложения задает поведение на краях.

#### Пример

Многочлен Чебышева, названный по имени русского математика, зарекомендовал себя подходящим средством для математического описания профиля зазора валков. Релевантными для аппроксимации зазора валков являются порядки от 0 до 6 многочлена, для которых функция возвращает соответствующие коэффициенты.

На практике коэффициенты могут быть выведены из измеренных значений ролика планшетности. При этом измеренные значения планшетности отдельных зон могут быть собраны в трехмерном сигнале 'Vector'. Каждое поле массива соответствует сегменту поперечного профиля.

### 13.2 CubicSpline

```
CubicSpline('Expression','X','Y')
```

#### Аргументы

'Expression'	Вспомогательный сигнал для определения частоты дискретизации события
'X'	Координаты X (опорные точки), которые задают сплайн
'Y'	Координаты Y, которые задают сплайн

#### Описание

Данная функция возвращает как результат кубический сплайн, настроенный в опорных точках 'X' с соответствующими значениями 'Y'. Частота дискретизации и точки обработки результата задаются посредством 'Expression'.

Координаты X не должны быть уникальными и сортироваться. Если есть несколько пар значений с теми же координатами X, то для вычисления сплайна используется только последняя пара значений. Оставшиеся пары значений автоматически сортируются по координатам X.

## Пример

Для ряда точек функция возвращает как результат сглаженный сигнал вдоль вычисленного сплайна. Функция может использоваться для интерполирования кривой замещения для сигнала с несколькими значениями:

График показывает только 17 значений за период в 5000 с (значения Y, зеленый график). Подходящие для этого координаты X - в данном случае только 17 значений - вносятся как синяя кривая. Кривая замещения должна получить как сглаженный сигнал значительно более высокое разрешение (больше значений). Поэтому функции *CubicSpline* будет передана как параметр 'Ausdruck' линейная функция с 5000 значений с интервалом в 1 с.

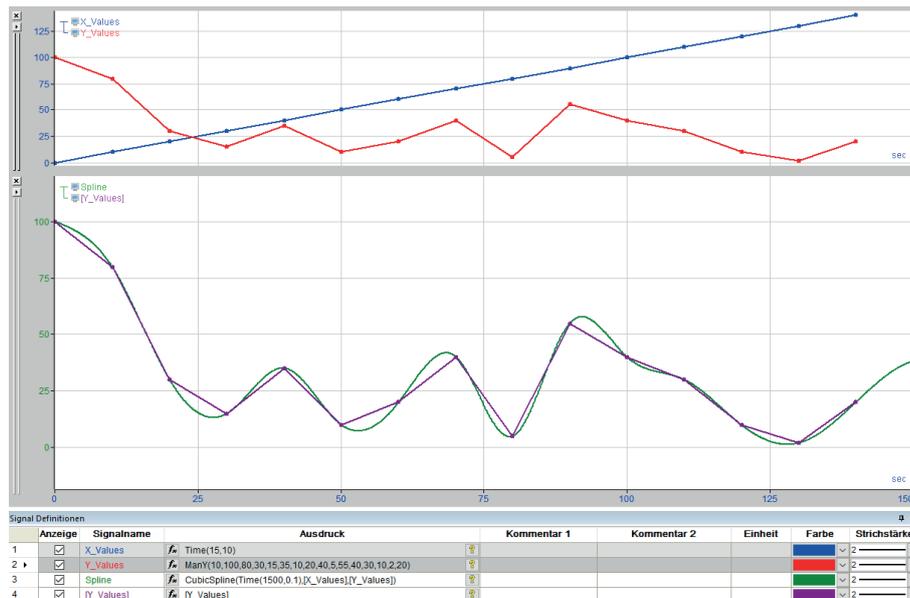


Рис. 28: Применение функции CubicSpline

В сильно увеличенном представлении можно распознать вычисленные значения кривой замещения. Исходные координаты X/Y образуют узлы сплайна (фиолетовые).

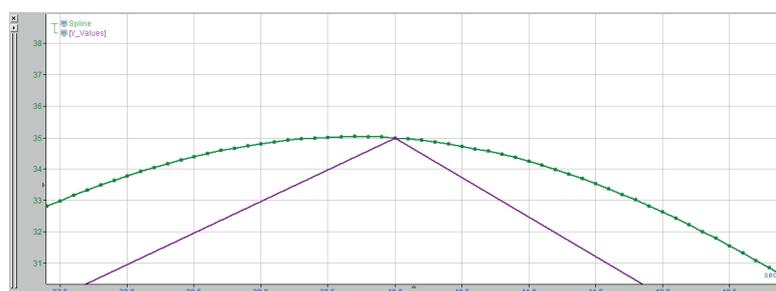


Рис. 29: Узлы сплайна и кривая результатов

## 13.3 LSQPolyCoef

```
LSQPolyCoef('X','Y','degree')
```

**Аргументы**

'X'	Координаты X (опорные точки), которые задают интерполяционный многочлен
'Y'	Координаты Y, которые задают интерполяционный многочлен
'degree'	Степень многочлена (0 = среднее значение, 1 = линейная, 2 = квадратичная, 3 = кубическая и т.д.)

**Описание**

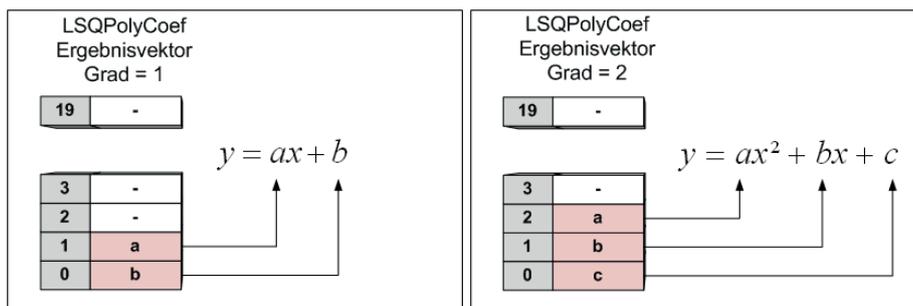
Данная функция вычисляет коэффициенты интерполяционного многочлена степени 'degree' для пар значений 'X' и 'Y' по методу наименьших квадратов.

Результатом функции является вектор (многомерный сигнал, массив), содержащий коэффициенты. Поле массива с индексом 0 содержит постоянную составляющую или смещение многочлена. Коэффициенты записываются с возрастающей степенью в соответствии с полями массива с возрастающим индексом.

Определение полинома может осуществляться при помощи функции 'Polynomial'.

**Пример**

Квадратичная аппроксимация возвращает полином формы  $y = ax^2 + bx + c$ , т. е. Как результат получают вектор с 3 коэффициентами.



Как правило, функция основывается на отношении X-Y, т. е. операнды X и Y могут быть двумя разными сигналами измерений. Если необходимо вычислить кривую регрессии для сигнала за время, то нужно привести значения времени в форму сигнала, например, при помощи функции XValues ([сигнал]). Данный сигнал времени, значения Y которого идентичны времени вдоль оси X, может использоваться тогда как операнд 'X' в функции LSQPolyCoef.

## 13.4 Полиномиальная

`Polynomial('Coefs', 'X')`

### Аргументы

'Coefs'	Вектор с коэффициентами, например, как результат функции <code>LSQExponentialCoef</code>
'X'	Координаты X (опорные точки), в которых должен быть определен многочлен.

### Описание

Данная функция вычисляет для каждого значения 'X' значение многочлена на базе вектора коэффициента *Coefs*'. Она требуется для изображения прямых регрессии или кривых замещения, коэффициенты которых были рассчитаны ранее при помощи функции *LSQPolyCoef*.

## 13.5 LSQExponentialCoef

`LSQExponentialCoef('X', 'Y')`

### Аргументы

'X'	Координаты X (точки выборки), которые задают интерполирующую экспоненциальную функцию.
'Y'	Значения Y, которые задают интерполирующую экспоненциальную функцию.

### Описание

Данная функция вычисляет коэффициенты интерполирующего полинома  $a \cdot \exp(b \cdot x)$  для заданной координатами 'X' функции и соответствующих значений функции 'Y' по методу наименьших квадратов. Результатом функции является вектор, содержащий коэффициенты.

## 13.6 Экспоненциальная

`Exponential('Coefs', 'X')`

### Аргументы

'Coefs'	Вектор с коэффициентами, например, результат <code>LSQExponentialCoef</code>
'X'	Координаты X (точки выборки), в которых определяется экспоненциальная функция

### Описание

Данная функция вычисляет значение экспоненциальной функции  $a \cdot \exp(b \cdot X)$  для каждого семпла 'X' с коэффициентами a и b, которые указаны как векторы 'Coefs'. Функция может использоваться для визуализации событий *LSQExponentialCoef*.

## 14 Спектральный анализ (операции БПФ)

*ibaAnalyzer* имеет возможность спектрального анализа методом быстрого преобразования Фурье (FFT - БПФ). При помощи доступных операций FFT сигнал на базе времени или длины может отображаться не только в режиме FFT, но и быть доступно как расчетное выражение и использоваться для последующих анализов.

Для большинства описанных здесь функций есть возможность отображать либо тренд амплитуд либо мощности. В именах функций они обозначаются как окончания 'Ampl' или 'Power'.

### 14.1 FftInTimeAmpl / FftInTimePower

например, `FftInTimeAmpl('Expression','Samples','#Freq','min_Freq'=0,'max_Freq','Window'=0,'Overlap'=0,'DC-Suppression'=0)`

#### Аргументы

'Expression'	Выражение, для которого вычисляется БПФ
'Time'	Определение используемых интервалов времени или длины. Здесь осуществляется округление до интервала, который содержит значения $2^N$
'#Freq'	Количество отображенных частот
'min_Freq'	Минимальная частота
'max_Freq'	Максимальная частота
'Window'	Тип окна: 0 = прямоугольник 1 = Бартлетт 2 = Блэкман 3 = Хэмминг 4 = Хэннинг 5 = Блэкман-Харрис 6 = Плоская вершина
'Overlap'	Коэффициент наложения
'DC-Suppression'	Подавление постоянной составляющей

#### Описание

Данные функции вычисляют амплитуды или мощность образов Фурье выражения 'Expression' для участков с измеренными значениями  $2^N$ . При этом N определяется посредством округления продукта 'Time' x частоту дискретизации до степени двух.

Результатом является вектор, который содержит '#Freq' равномерно распределенных частот между 'min\_Freq' и 'max\_Freq' для каждого участка. Типом окна, используемым для вычисления, можно управлять через параметр 'Window'.

Коэффициент наложения задает наложение сегментов времени и может быть между 0 (без наложения) и 1 (полное наложение). В качестве опции возможно активировать подавление постоянной составляющей при помощи параметра 'DC-Suppression'.

### Пример

При помощи функции *FftInTime* можно изобразить колебания частоты в течение времени. Для этого может быть отображен результирующий вектор в двумерном представлении.

## 14.2 FftOrderAnalysisAmpl / FftOrderAnalysisPower

например, `FftOrderAnalysisAmpl('Expression', 'Samples', 'Freq', 'min_Order'=0, 'max_Order', 'Order_division', 'Window'=0, 'Overlap'=0, 'DC-Suppression'=0)`

### Аргументы

'Expression'	Выражение, для которого выполняется порядковый анализ
'Time'	Определение используемых интервалов времени или длины
'Freq'	Основная частота для порядкового анализа (частота вращения)
'min_Order'	Минимальный отображаемый порядок
'max_Order'	Максимальный отображаемый порядок
'Order_division'	Ширина раstra между целочисленными порядками
'Window'	Тип окна 0 = прямоугольник 1 = Бартлетт 2 = Блэкман 3 = Хэмминг 4 = Хэннинг 5 = Блэкман-Харрис 6 = Плоская вершина
'Overlap'	Коэффициент наложения
'DC-Suppression'	Подавление постоянной составляющей

### Описание

Данная функция вычисляет порядки (т.е. кратные величины основной частоты 'Freq') для сигнала и возвращает как результат вектор с порядками между 'min\_Order' и 'max\_Order'. Количество точек данных на каждый порядок задается параметром 'Order\_division'.

Типом окна, используемым для вычисления, можно управлять через параметр 'Window'. Коэффициент наложения задает наложение сегментов времени и может быть между 0 (без наложения) и 1 (полное наложение). В качестве опции возможно активировать подавление постоянной составляющей при помощи параметра 'DC-Suppression'.

В отличие от функции *FftInTime* на оси Y больше не отображаются время/частота, а отображаются частоты вращения и их кратные, т.е. порядки. Ось частоты, к тому же, искажается в соответствии с мгновенной частотой, таким образом, что порядки больше не отображаются как график, а отображаются как прямые линии. В зависимости от функ-

ции вычисляются либо тренд амплитуды (*FftOrderAnalysisAmpl*) либо тренд мощности (*FftOrderAnalysisPower*).

**Примечание**



Функция не возвращает результаты, если количество точек сигналов в обороте больше чем в два раза выбранного параметра 'Time'.

**Пример**

При помощи функции *FftOrderAnalysis* Вы можете выполнять вычисление порядкового анализа. Частоты, которые соответствуют числу оборотов двигателя или его кратным, называются порядками. Первый порядок соответствует частоте оборотов двигателя, второй порядок соответствует частоте первого порядка, помноженной на коэффициент 2 и т.д. При порядковом анализе вычисляется уровень или график уровня данных порядков.

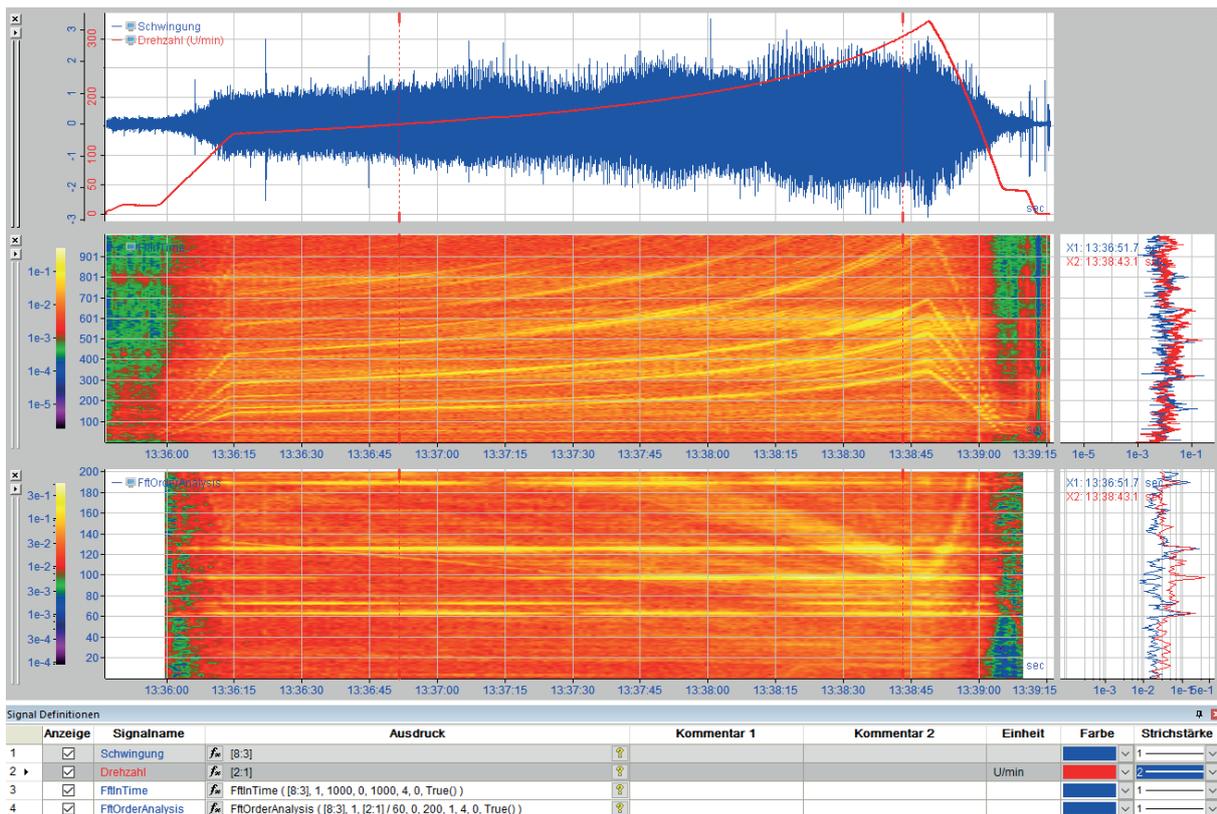


Рис. 30: Применение FftInTime и FftOrderAnalysis к сигналу колебания

Между отдельными точками сигнала осуществляется интерполяция для вычисления *FftOrderAnalysis*.

### 14.3 FftPeaksInTimeAmpl / FftPeaksInTimePower

например, `FftPeaksInTimeAmpl('Expression', 'Samples', '#Peaks', 'min_Freq'=0, 'max_Freq', 'Window'=0, 'Overlap'=0, 'DC-Suppression'=0, 'Zero-Padding'=0)`

#### Аргументы

'Expression'	Выражение, для которого выполняется анализ частотных пиков
'Time'	Определение используемых интервалов времени или длины. Здесь осуществляется округление до интервала, который содержит значения $2^N$
'#Peaks'	Количество отображенных пиков
'min_Freq'	Опциональный параметр для минимально учитываемой частоты, т.е. пики при более низких частотах не отображаются
'max_Freq'	Опциональный параметр для максимально учитываемой частоты, т.е. пики при более высоких частотах не отображаются
'Window'	Тип окна: 0 = прямоугольник 1 = Бартлетт 2 = Блэкман 3 = Хэмминг 4 = Хэннинг 5 = Блэкман-Харрис 6 = Плоская вершина
'Overlap'	Коэффициент наложения
'DC-Suppression'	Подавление постоянной составляющей
'Zero-Padding'	Дополнение нулями

#### Описание

Данная функция служит для вычисления частотных пиков за скользящие интервалы времени, заданные параметром 'Time'. При этом вычисляются '#Peaks', наивысшие пики между частотами 'min\_Freq' и 'max\_Freq'. Временной график частоты и пары пиковых значений возвращаются как вектор.

При этом записи сортируются по следующему образцу:

- Индекс 0: Частота с наивысшим пиком
- Index 1: Амплитуда/мощности наивысшего пика
- Индекс 2: Частота со вторым по высоте пиком
- Индекс 3: Амплитуда/мощности второго по высоте пика
- и т.д.

**Совет**

Для чтения желаемых значений из массива результатов Вы можете использовать функцию 'GetRows'.

Типом окна, используемым для вычисления, можно управлять через параметр 'Window'. Коэффициент наложения задает наложение сегментов времени и может быть между 0 (без наложения) и 1 (полное наложение). В качестве опции возможно активировать подавление постоянной составляющей при помощи параметра 'DC-Suppression'. Если параметр 'Zero\_Padding' 1 или Истина(), то перед вычислением БПФ последнее окно заполняется нулями.

## 14.4 FftAmpl / FftPower

z.B. `FftAmpl('Expression','Samples','Window'=0,'DC-Suppression'=0,'Zero-Padding'=0)`

### Аргументы

'Expression'	Выражение, для которого проводится вычисление образа Фурье
'Samples'	Количество учитываемых измеренных значений и следовательно неявное определение использованного интервала времени или длины, в зависимости от частоты дискретизации.
'Window'	Тип окна: 0 = прямоугольник 1 = Бартлетт 2 = Блэкман 3 = Хэмминг 4 = Хэннинг 5 = Блэкман-Харрис 6 = Плоская вершина
'DC-Suppression'	Подавление постоянной составляющей
'Zero-Padding'	Дополнение нулями

### Описание

Данная функция вычисляет амплитуду или мощность образов Фурье сигнала. Используемый раздел времени определяется округлением использованных точек измерения 'Samples' до степени двух.

**Примечание**

Параметр 'Samples' округляется. Должны использоваться как минимум 128 точек измерений.

Типом окна, используемым для вычисления, можно управлять через параметр 'Window'. В качестве опции возможно активировать подавление постоянной составляющей при помощи параметра 'DC-Suppression'. Если параметр 'Zero\_Padding' 1 или Истина(), то для вычисления БПФ окно заполняется нулями.

## 14.5 FftComplex

```
FftComplex('Expression','inv','normalize'=0)
```

**Аргументы**

'Expression'	Выражение, для которого вычисляется преобразование Фурье
'inv'	Опциональный параметр для обеспечения обратного преобразования Фурье
'normalize'	Опциональный параметр для выбора нормализации

**Описание**

Данная функция выполняет преобразование Фурье для комплексного сигнала по всему выражению и возвращает как результат вектор с вещественной и мнимой частью образов Фурье. Входной сигнал может быть отдельным сигналом или вектором, состоящим из вещественной и мнимой части. При этом используется окно для вычисления.

Если параметр 'inv' установлен на True() или 1, вычисляется обратное преобразование Фурье. В данном случае функция ожидает входной сигнал либо на базе частоты либо на базе обратной длины. Результатом действия тогда является соответственно сигнал на базе времени или длины.

Для параметра 'normalize' допустимы следующие значения:

- **0:** Нормализация не выполняется.
- **1:** Результат делится на количество значений. При обратной трансформации результат не изменяется.
- **2:** Результат делится на квадратный корень количества значений. Это касается как обычного, так и обратного преобразования.
- Другие значения: Функция как со значением 1

Количество значений частоты определяется количеством значений входного сигнала. Если N четное число, вычисляются точки частоты  $N/2+1$ , из которых первая (постоянная составляющая) и последняя точка являются чисто вещественными. Если N нечетное число, то вычисляются точки частоты  $(N+1)/2$ , для которых постоянная составляющая является чистой вещественной.

## 14.6 FftReal / FftRealInverse

Например, `FftReal('Expression','normalize'=0)`

### Аргументы

'Expression'	Выражение, для которого проводится вычисление образа Фурье
'normalize'	Оptionальный параметр для активации нормализации

### Описание FftReal

Данная функция выполняет преобразование Фурье для вещественного сигнала по всему выражению и возвращает как результат вектор с вещественной и мнимой частью образов Фурье. При этом используется прямоугольное окно.

Если параметр 'normalize' Истина() или 1, то выполняется нормализация. Если количество значений (N) сигнала нечетное, все значения частоты, кроме постоянной составляющей, делятся на N/2. Если N четное, то все значения частоты, кроме постоянной составляющей и последнего значения, делятся на N/2.

Количество значений частоты задается количеством значений входного сигнала. Если N четное число, вычисляются точки частоты  $N/2+1$ , из которых первая (постоянная составляющая) и последняя точка являются чисто вещественными. Если N нечетное число, то вычисляются точки частоты  $(N+1)/2$ , для которых постоянная составляющая является чистой вещественной.

### Описание FftRealInverse

Данная функция вычисляет обратное преобразование Фурье, как создано при помощи *FftReal*. При этом результат является вещественным числом и входной сигнал должен быть соответственно вектором, состоящим из вещественной и мнимой части. В противном случае функция работает точно так же как и *FftReal*.

## 14.7 AWeighting / DbScale

### AWeighting

`AWeighting('Spectrum','Type')`

### Аргументы

'Spectrum'	Спектр
'Type'	Указание типа

### Описание

Данная функция взвешивает спектр по так называемой шкале А. Это фильтр взвешивания, который соответствует восприятию человеческим ухом.

### Пример

После применения данной функции спектр можно проанализировать относительно воспринимаемых уровней акустического шума.

### DbScale

`DbScale('Spectrum','Reference')`

**Аргументы**

'Spectrum'	Спектр, подлежащий логарифмическому масштабированию
'Reference'	Опционально

**Описание**

Данная функция предоставляет логарифмическое масштабирование в дБ для спектра сигнала. Значимый результат можно ожидать только, если в качестве вход дана амплитуда спектра.

**Совет**

Функции, вычисляющие такую амплитуду: *FftAmpl*, *FftInTimeAmpl*, *FftOrderAnalysisAmpl*

## 14.8 IntSpectrum

`IntSpectrum('Spectrum')`

**Описание**

Данная функция интегрирует заданный спектр. Таким образом из частоты датчика ускорения можно вычислить виброскорость.

## 15 Электрические функции

### 15.1 СКЗ / Eff

СКЗ('Spectrum', 'Frequency') или Eff('Spectrum', 'Frequency')

#### Аргументы

'Spectrum'	Измеренное значение, для которого рассчитывается действующее значение
'Frequency'	Основная частота

#### Описание

Данная функция вычисляет так называемое среднеквадратичное значение или действующее (эффективное значение) 'Expression' с основной частотой 'Expression':

$$E_{\text{eff}} = \sqrt{\frac{1}{N} \sum_{n=1}^N e^2(n)}$$

e(n): Точка измерения n сигнала e ('expr')

N: Количество измеренных значений за период

#### Пример

Для графика переменного напряжения с частотой 0,1 кГц с наложением второго переменного напряжения 0,5 кГц требуется рассчитать эффективное значение напряжения для обоих частот, применив функцию *Eff* со вторым аргументом 0,1 или 0,5.

#### Примечание



Между функциями *СКЗ* и *Eff* нет различий. Обе функции поддерживаются в целях совместимости *ibaAnalyzer*.

## 15.2 Функции для сети "треугольник»

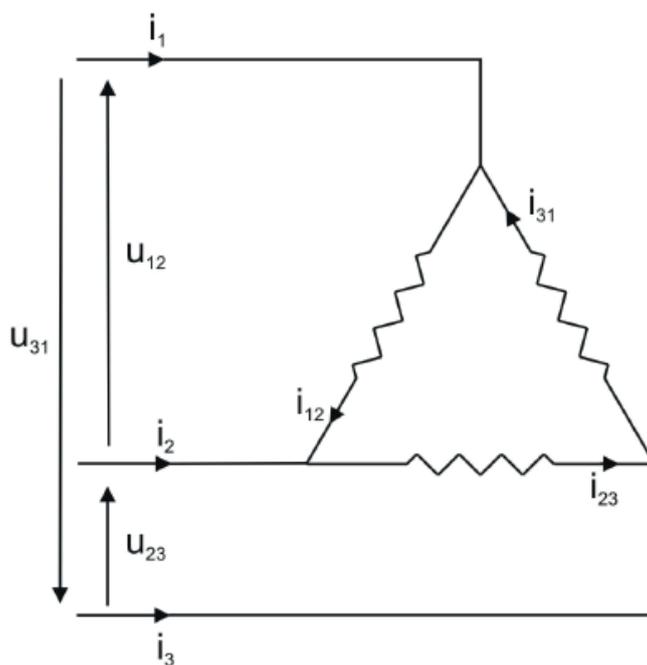


Рис. 31: Электрические функции, сеть «треугольник»

Функции для сети «треугольник» используют напряжения и токи питающей сети в сети "треугольник», чтобы вычислить значения мощности.

### Аргументы

'u12', 'u13', 'u23'	Напряжение питающей сети (равны фазным напряжением)
'i1', 'i2', 'i3'	Токи питающей сети
'freq'	Основная частота

### Примечание



Функции применяются обычно для сетей с топологией «треугольник», но они могут использоваться и для других сетей, в которых могут измеряться напряжения и токи питающей сети.

### DeltaCollectiveUeff

DeltaCollectiveUeff('u12', 'u13', 'u23', 'freq')

Вычисляет общее эффективное напряжение в сети «треугольник»:

$$U_{\text{eff}} = \sqrt{\frac{1}{3}(U_{12,\text{eff}}^2 + U_{23,\text{eff}}^2 + U_{31,\text{eff}}^2)}$$

$U_{xy,\text{eff}}$  : эффективное значение напряжения питающей сети  $U_{xy}$

**DeltaCollectiveIeff**

DeltaCollectiveIeff('i1', 'i2', 'i3', 'freq')

вычисляет общий эффективный ток в сети «треугольник»

$$I_{\text{eff}} = \sqrt{\sum_{x=1}^3 I_{x,\text{eff}}^2}$$

 $I_{x,\text{eff}}$  : эффективное значение тока питающей сети ix**DeltaActiveP**

DeltaActiveP('u13', 'u23', 'i1', 'i2', 'freq')

вычисляет общее активное напряжение в сети «треугольник»:

$$P = \frac{1}{N} \sum_{n=1}^N [u_{23}(n)i_2(n) + u_{13}(n)i_1(n)]$$

N : количество измеренных значений за период

 $u_{xy}$  : напряжение между соединением x и y ( $u_{13} = -u_{31}$ ) $i_x$  : ток при подключении в линии x**DeltaApparentP**

DeltaApparentP('u12', 'u13', 'u23', 'i1', 'i2', 'i3', 'freq')

вычисляет полное напряжение в сети «треугольник»:

$$S = U_{\text{eff}} I_{\text{eff}}$$

 $U_{\text{eff}}$  : общее эффективное напряжение $I_{\text{eff}}$  : общий эффективный ток**DeltaReactiveP**

DeltaReactiveP('u12', 'u13', 'u23', 'i1', 'i2', 'i3', 'freq')

вычисляет реактивное напряжение в сети «треугольник»:

$$Q = \sqrt{S^2 - P^2}$$

S : Полная мощность

P : Активная мощность

**DeltaReactivePS**`DeltaReactivePS ('u12', 'u13', 'u23', 'i1', 'i2', 'i3', 'freq')`

вычисляет знакопеременную реактивную мощность QS в сети «треугольник»:

**DeltaActivePFactor**`DeltaActivePFactor ('u12', 'u13', 'u23', 'i1', 'i2', 'i3', 'freq')`

вычисляет коэффициент активной мощности в сети «треугольник»:

$$\cos \varphi = \frac{P}{S}$$

S : Полная мощность

P : Активная мощность

**DeltaReactivePFactor**`DeltaReactivePFactor ('u12', 'u13', 'u23', 'i1', 'i2', 'i3', 'freq')`

вычисляет коэффициент реактивной мощности в сети «треугольник»:

$$\tan \varphi = \frac{Q}{P}$$

Q : Реактивная мощность

P : Активная мощность

**DeltaReactivePFactorS**`DeltaReactivePFactorS ('u12', 'u13', 'u23', 'i1', 'i2', 'i3', 'freq')`

вычисляет знакопеременный коэффициент реактивной мощности в сети «треугольник»:

$$\tan \varphi = \frac{Q_s}{P}$$

$Q_s$  : знакопеременная реактивная мощность

P : Активная мощность

### 15.3 Функции для сети "звезда»

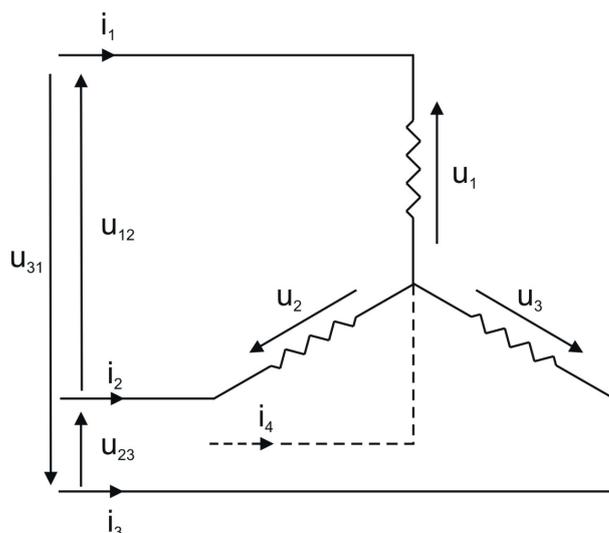


Рис. 32: Электрические функции, сеть «звезда»

Функции для сети «звезда» используют фазные напряжения и токи, чтобы вычислить различные значения мощности.

#### Аргументы

'u1', 'u2', 'u3'	Фазные напряжения
'i1', 'i2', 'i3'	Фазные токи
'i4'	Подключение «звездой»
'freq'	Основная частота

#### Примечание



Функции применяются обычно для сетей с топологией «звезда», но они могут использоваться и для других сетей, в которых могут измеряться фазные напряжения и токи.

#### StarCollectiveUeff

StarCollectiveUeff('u1', 'u2', 'u3', 'freq')

Вычисляет общее эффективное напряжение в сети «звезда»

$$U_{\text{eff}} = \sqrt{\sum_{x=1}^4 U_{x\text{-eff}}^2}$$

$U_{x\text{-eff}}$  : эффективное значение фазного напряжения  $u_x$

$$u_4 = u_1 + u_2 + u_3$$

**StarCollectiveIeff**

StarCollectiveIeff('i1','i2','i3','i4','freq')

вычисляет общий эффективный ток в сети «звезда»

$$I_{\text{eff}} = \sqrt{\sum_{x=1}^4 I_{x,\text{eff}}^2}$$

$I_{x,\text{eff}}$  : эффективное значение тока подключения ix

**StarActiveP**

StarActiveP('u1','u2','u3','i1','i2','i3','freq')

Вычисляет общее активное напряжение в сети «звезда»:

$$P = \sum_{x=1}^3 \left( \frac{1}{N} \sum_{n=1}^N u_x(n) i_x(n) \right)$$

N : Количество измеренных значений за период

$u_x$  : Напряжение фазы x

$i_x$  : Ток фазы x

**StarApparentP**

StarApparentP('u1','u2','u3','i1','i2','i3','i4','freq')

Вычисляет полную мощность в сети «звезда»:

$$S = U_{\text{eff}} I_{\text{eff}}$$

$U_{\text{eff}}$  : общее эффективное напряжение

$I_{\text{eff}}$  : общий эффективный ток

**StarReactiveP**

StarReactiveP('u1','u2','u3','i1','i2','i3','i4','freq')

Вычисляет общее реактивное напряжение в сети «звезда»:

$$Q = \sqrt{S^2 - P^2}$$

S : Полная мощность

P : Активная мощность

**StarReactivePS**

```
StarReactivePS('u1','u2','u3','i1','i2','i3','i4','freq')
```

Вычисляет знакопеременную реактивную мощность QS в сети «звезда».

**StarActivePFactor**

```
StarActivePFactor('u1','u2','u3','i1','i2','i3','i4','freq')
```

Вычисляет коэффициент активной мощности в сети «звезда»:

$$\cos \varphi = \frac{P}{S}$$

S : Полная мощность

P : Активная мощность

**StarReactivePFactor**

```
StarReactivePFactor('u1','u2','u3','i1','i2','i3','i4','freq')
```

Вычисляет коэффициент реактивной мощности в сети «звезда»:

$$\tan \varphi = \frac{Q}{P}$$

Q : Реактивная мощность

P : Активная мощность

**StarReactivePFactorS**

```
StarReactivePFactorS('u1','u2','u3','i1','i2','i3','i4','freq')
```

Вычисляет знакопеременный коэффициент реактивной мощности в сети «звезда»:

$$\tan \varphi = \frac{Q_s}{P}$$

$Q_s$  : знакопеременная реактивная мощность

P : Активная мощность

**15.4 Гармонические функции****HarmEff**

```
HarmEff('u','Nharm','freq')
```

Вычисляет действующее значение 'Nharm'-ной гармоники сигнала 'u':

$$u_{\text{Real},k} = \frac{2}{N} \sum_{n=0}^{N-1} u(n) \cos \frac{2\pi kn}{N}$$

$$u_{\text{Imag},k} = \frac{2}{N} \sum_{n=0}^{N-1} u(n) \sin \frac{2\pi kn}{N}$$

$$U_k = \frac{\sqrt{u_{\text{Real},k}^2 + u_{\text{Imag},k}^2}}{\sqrt{2}}$$

$u(n)$  : точка измерения  $n$  сигнала  $u$

$u_{\text{Real},k}$  : вещественная часть  $k$ -ого гармонического компонента сигнала  $u$

$u_{\text{Imag},k}$  : мнимая часть  $k$ -ого гармонического компонента сигнала  $u$

$U_k$  : эффективное значение  $k$ -ого гармонического компонента сигнала  $u$

### HarmPhase

`HarmPhase('u', 'Nharm', 'freq')`

вычисляет смещение фаз 'Nharm'-ного гармонического компонента сигнала 'u':

$$\varphi_k = -a \tan \left( \frac{u_{\text{Imag},k}}{u_{\text{Real},k}} \right)$$

$u_{\text{Real},k}$  : вещественная часть  $k$ -ного гармонического компонента сигнала  $u$

$u_{\text{Imag},k}$  : мнимая часть  $k$ -ого гармонического компонента сигнала  $u$

$\varphi_k$  : смещение фаз  $k$ -ого гармонического компонента сигнала  $u$

### StarHarmUGeff

`StarHarmUGeff('u1', 'u2', 'u3', 'freq')`

вычисляет эффективное напряжение обратной последовательности  $UGeff$ :

$$U_G = \frac{1}{3} \left[ u_{1,1} + u_{2,1} \left(-\frac{2}{3}\pi\right) + u_{3,1} \left(-\frac{4}{3}\pi\right) \right]$$

$$U_{\text{Geff}} = \frac{\sqrt{U_{G,\text{real}}^2 + U_{G,\text{imag}}^2}}{\sqrt{2}}$$

$u_{x,1}$  : Указатель основной гармоники (комплексный) фазного напряжения  $u_x$

### StarHarmUMeff

`StarHarmUMeff('u1', 'u2', 'u3', 'freq')`

Вычисляет напряжение системы прямой последовательности  $U_{\text{Meff}}$ :

$$U_M = \frac{1}{3} \left[ u_{1,1} + u_{2,1} \left( \frac{2}{3} \pi \right) + u_{3,1} \left( \frac{4}{3} \pi \right) \right]$$

$$U_{\text{Meff}} = \frac{\sqrt{U_{M,\text{real}}^2 + U_{M,\text{imag}}^2}}{\sqrt{2}}$$

$u_{x,1}$  : Указатель основной гармоники (комплексный) фазного напряжения  $u_x$

### StarHarmUnSym

StarHarmUnSym('u1', 'u2', 'u3', 'freq')

Вычисляет асимметрию напряжения в сети «звезда»:

$$\text{SYM} = \frac{U_{\text{Geff}}}{U_{\text{Meff}}} \times 100$$

Результат выводится в %.

### WeightedDistortionFactor

WeightedDistortionFactor('u', 'Nharm'=50, 'freq')

Вычисляет взвешенный коэффициент нелинейных искажений 'u' (всех фаз) с гармониками 'NHarm':

$$D_w = \frac{\sqrt{\sum_{n=2}^{\text{Nharm}} n^2 U_n^2}}{U_1}$$

$U_n$  : Эффективное значение n-ной гармоники величины u

### UnweightedDistortionFactor

UnweightedDistortionFactor('u', 'Nharm'=50, 'freq')

Вычисляет невзвешенный коэффициент нелинейных искажений 'u' (всех фаз) с гармониками 'NHarm':

$$D_{\text{uw}} = \frac{\sqrt{\sum_{n=2}^{\text{Nharm}} U_n^2}}{\sqrt{\sum_{n=1}^{\text{Nharm}} U_n^2}}$$

$U_n$  : Эффективное значение n-ной гармоники величины u

## 15.4.1 TIF

TIF('u', 'Nharm'=50, 'freq')

Вычисляет коэффициент помех телефонной связи величины 'u' с учетом гармоник 'NHarm'

$$\text{TIF} = \frac{1}{U_1} \sqrt{\sum_{n=2}^{\text{NHarm}} (K_n \times P_n \times U_n)^2}$$

$K_n = 5 * n * \text{freq}$

$P_n$  = коэффициент взвешивания по BTS (British Telephone System)

$U_n$  : Эффективное значение напряжения n-ной гармоники величины u

$U_1$  : Эффективное значение напряжения основной частоты величины u

## 16 Техподдержка и контакты

### Служба поддержки

Тел.: +49 911 97282-14  
Факс +49 911 97282-33  
E-Mail: support@iba-ag.com

---

#### Важно



При обращении в техподдержку указывайте серийный номер (iba-S/N) продукта.

---

### Контактные данные

#### Фактический адрес

iba AG  
Königswarterstraße 44  
90762 Fürth  
Deutschland иба АГ, Кёнигсвартерштрассе 44, 90762 Фюрт Германия

Тел.: +49 911 97282-0  
Факс +49 911 97282-33  
E-Mail: iba@iba-ag.com  
Контактное Харальд Опель  
лицо:

#### Адрес для корреспонденции

iba AG  
Gebhardtstraße 10  
90762 Fürth  
Deutschland иба АГ, Гебхардтштрассе 10, 90762 Фюрт Германия

#### В Германии и по всему миру

Адреса других наших региональных филиалов и представительств можно найти на нашей веб-странице.

**[www.iba-ag.com](http://www.iba-ag.com)**.