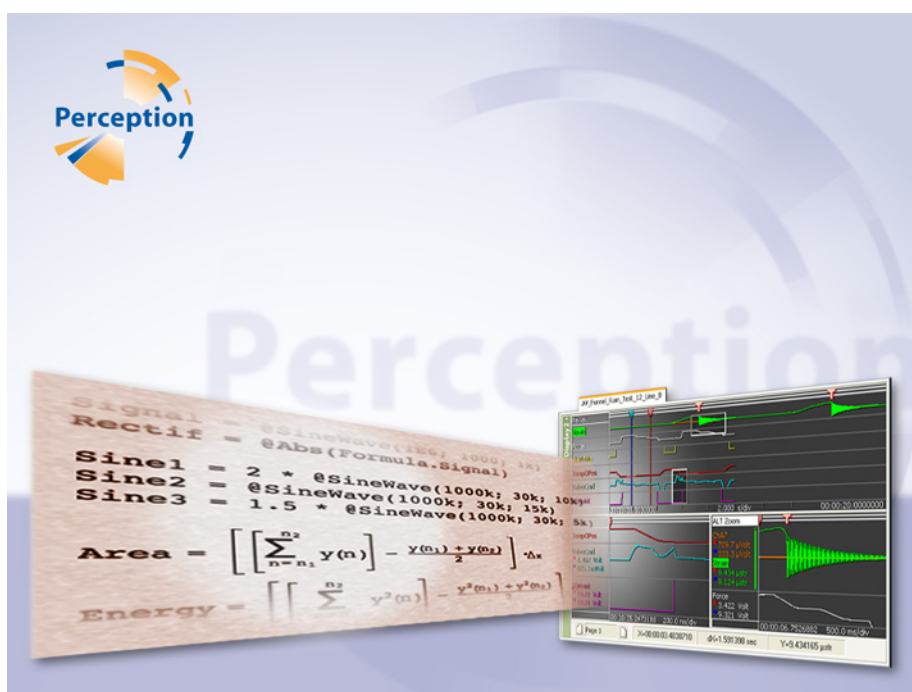


ユーザーマニュアル

日本語



分析オプション Perception

ドキュメントバージョン 2.0 - 2010 年 10 月

Perception 6.0 以降

HBM の諸条件につきましては、www.hbm.com/terms

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

Copyright © 2010

無断複写、転載を禁じます。本書のどの部分も、出版社の書面による許可なく、いかなる形式またはいかなる手段によっても、複製または転載することはできません。

使用許諾契約と保証

使用許諾契約と保証の詳細につきましては、www.hbm.com/terms を参照ください。

目次	ページ
1 分析オプション	9
1.1 はじめに	9
1.1.1 分析オプションのインストール方法	9
1.2 公式データベースシート	11
1.3 定義	14
1.3.1 定数	14
1.3.2 変数	14
1.3.3 関数	14
1.4 レイアウトの修正	16
1.4.1 列を追加、削除、およびクリアする	16
1.4.2 移動	17
1.5 公式の作成	18
1.5.1 コメントの入力	18
1.5.2 公式の入力	18
1.6 公式メニュー	20
1.6.1 公式をロード	20
1.6.2 公式を保存する	21
1.6.3 公式を印刷	21
1.6.4 シートを移動する	21
2 公式データベース機能	23
2.1 一般事項	23
2.2 概要	24
2.2.1 関数の概要	24
3 算術演算	29
3.1 + (加算)	29
3.2 - (減算)	31
3.3 * (乗算)	33
3.4 / (除算)	35
3.5 - (単項マイナス)	37
4 リファレンスガイド	38
4.1 @Abs	38
4.2 @And	39
4.3 @Area	40
4.4 @ATan	42
4.5 @BlockFFT	43

4.6	@Clip	45
4.7	@Cos	46
4.8	@CurveFitting	47
4.9	@Cut	49
4.10	@Cycles	51
4.11	@Diff	53
4.12	@Energy	55
4.13	@EqualTo	57
4.14	@Exp	58
4.15	@ExpWave	59
4.16	@FallTime	61
4.17	@FilterButterworthLP	63
4.18	@FilterButterworthHP	65
4.19	@FilterButterworthBP	67
4.20	@FilterButterworthBS	69
4.21	@FilterBesselLP	71
4.22	@FilterBesselHP	73
4.23	@FilterBesselBP	75
4.24	@FilterBesselBS	77
4.25	@FilterChebyshevLP	79
4.26	@FilterChebyshevHP	82
4.27	@FilterChebyshevBP	84
4.28	@FilterChebyshevBS	86
4.29	@Frequency	89
4.30	@GreaterEqualThan	91
4.31	@GreaterThan	92
4.32	@Histogram	93
4.33	@IIF	95
4.34	@Integrate	97
4.35	@IntLookUp	98
4.36	@IntLookUp12	101
4.37	@Join	103
4.38	@Length	105
4.39	@LessEqualThan	106
4.40	@LessThan	107
4.41	@Ln	108
4.42	@Log	110

4.43	@Max	111
4.44	@MaxNum	113
4.45	@MaxPos	114
4.46	@Mean	116
4.47	@Min	118
4.48	@MinNum	120
4.49	@MinPos	121
4.50	@NextHillPos	123
4.51	@NextLvlCross	125
4.52	@NextValleyPos	127
4.53	@Noise	129
4.54	@Not	130
4.55	@Or	131
4.56	@Period	132
4.57	@Pow	134
4.58	@PrevHillPos	135
4.59	@PrevLvlCross	137
4.60	@PrevValleyPos	139
4.61	@Pulse	141
4.62	@PulseWidth	143
4.63	@Ramp	145
4.64	@ReadAsciiFile	147
4.65	@Reduce	149
4.66	@RefCheck	150
4.67	@RemoveGlitch	152
4.68	@Res2	153
4.69	@RiseTime	154
4.70	@RMS	156
4.71	@SAEJ211Filter	158
4.72	@Sin	159
4.73	@SineWave	160
4.74	@Smooth	161
4.75	@Sqrt	163
4.76	@SquareWave	164
4.77	@StdDev	165
4.78	@Sweep	167
4.79	@Tan	168

4.80	@TriggerTime	169
4.81	@TriggerTimeToText	171
4.82	@TrueRMS	173
4.83	@Value	174
4.84	@XDelta	175
4.85	@XDeltaHigh	176
4.86	@XDeltaLow	177
4.87	@XFirst	178
4.88	@XLast	179
4.89	@XShift	180
4.90	@XYArray	181
4.91	@YArray	183
5	パルス測定および分析	184
5.1	一般事項	184
6	IIR フィルタ	188
6.1	はじめに	188
6.1.1	ベッセル	190
	利点 :	190
	欠点 :	190
6.1.2	バターワース	190
	利点 :	190
	欠点 :	190
6.1.3	チェビシェフ (I 型)	191
	利点 :	191
	欠点 :	191
6.1.4	振幅スペクトル	191
6.1.5	インパルス応答	194
6.1.6	ステップ応答	195
6.1.7	フェーズレスフィルタリング	196
6.1.8	サンプリング率とカットオフ周波数の重要性	197

1 分析オプション

1.1 はじめに

Perception の分析オプションを使って、測定データの計算を実施することができます。内蔵された様々な機能を使うことで、基本的な統計学から高度な数学までに行える分析作業を適切に行うことができます。

分析オプションは、2つの主要な部分で構成されます。

- 様々な機能
- 表示可能な公式データベース

公式データベースを使用すれば、プログラミングもシーケンシングも行うことなく一連の追加的な機能を作成することができます。必要な計算を入力すると、結果が表示されます。公式データベースにおいてだけでなく、表示部とレポートにおいても直接、新しいデータの結果が自動的に更新されます。定義された公式は、今後使用できるように保存することができます。

公式データベースは、無制限の数の公式を名前と単位と一緒に保存することが可能です。公式は、波形とスカラー上で算術演算を用いて作成され、内蔵されたいずれかの機能、カーソル情報、または他の公式の結果と組み合わせることができます。自動記入およびインラインヘルプにより、様々なオプションを利用することができます。

使いやすさを考慮して、機能と公式データベースを含む分析オプションを「公式データベース」と呼びます。

1.1.1 分析オプションのインストール方法

Perception のソフトウェアには HASP キーが必要です。HASP (違法コピーソフトウェア防止ハードウェア) は、ソフトウェアアプリケーションの不正使用を防止するハードウェアをベースとした (ハードウェアキーによる) ソフトウェアの違法コピー防止システムです。

それぞれの HASP キーには、購入した特性とオプションに応じてアプリケーションをパーソナライズするために使用する固有の ID 番号が含まれています。このキーは、ライセンスパラメータ、アプリケーション、および顧客固有のデータを保存するためにも使用されます。

分析オプションを個別のアイテムとして購入した場合、パーソナライズされた「キーファイル」が送られます。このファイルを使用して追加機能のロックを解除してください。

ヘルプ ▶ Perception についての順に選択すれば、シリアル番号を見つけることができます。

キー情報の更新方法 :

- 1 ヘルプ ▶ キーの更新...の順に選んでください。
- 2 開くダイアログでキーファイル (*.pKey) を見つけ、開くをクリックします。
- 3 問題がなければ、以下のメッセージが表示されます。

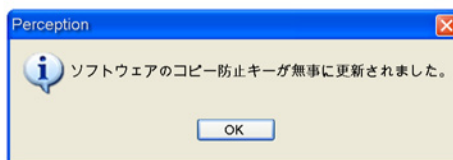


イラスト 1.1: ソフトウェアのコピー防止ダイアログ

- 4 OK をクリックします。
インストールを行ったら、ヘルプ ▶ Perception について ▶ 詳細の順に選択し、すべてのオプションがインストールされているか確認してください。

変更を有効化させるために、ソフトウェアを再起動する必要があります。これで分析オプションを使用することが可能です。

1.2 公式データベースシート

公式データベースシートを使用して、公式を作成および編集します。

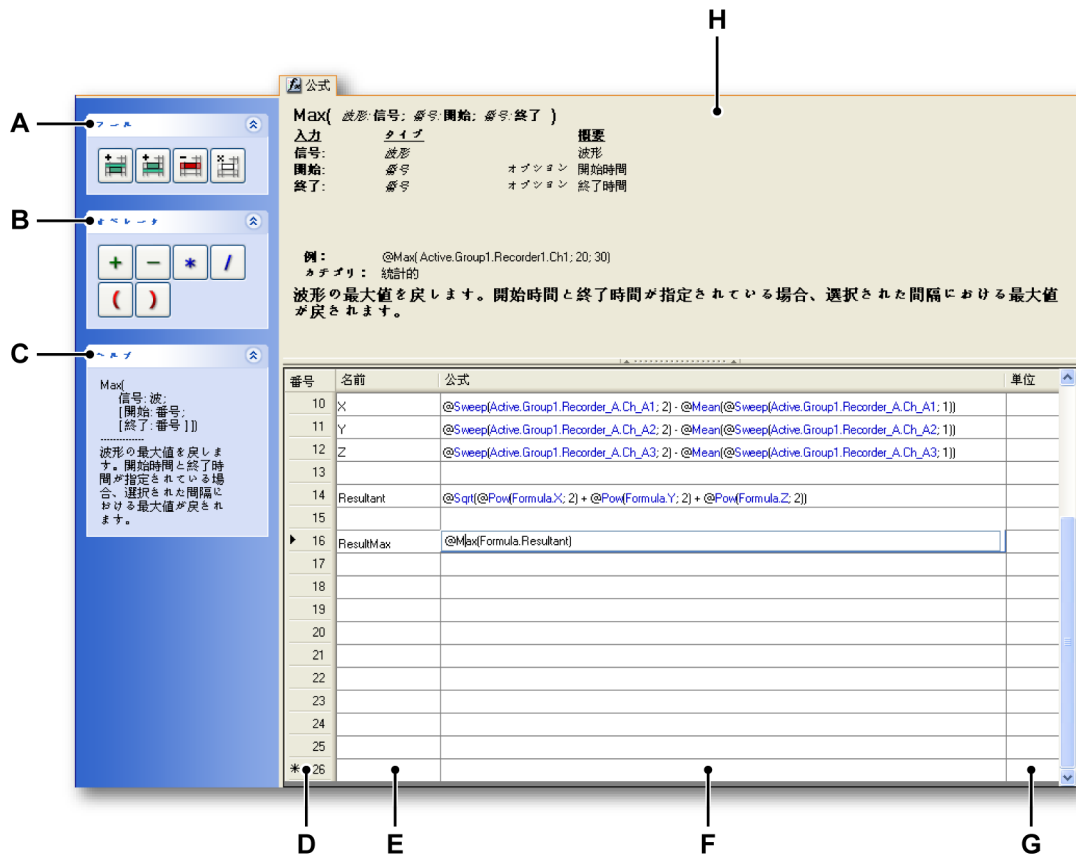


イラスト 1.2: 公式データベースシート

- A ツール
- B オペレータ
- C デフォルトヘルプ
- D 行ポインタを持つ公式データベースの行番号
- E 公式名のコラム
- F 公式コラム
- G 公式単位
- H オプションのヘルプ

A ツールツールは公式がある行を追加、削除、およびクリアすることを目的に提供されます。ツールは、現在選択されている列で動作します。現在選択されている列は、列ポインタによって示されます。

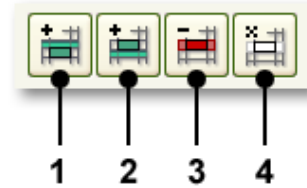


イラスト 1.3: 公式シートツール

- 1 現在の列の上に列を挿入
- 2 現在の列の下に列を挿入
- 3 列を削除
- 4 列をクリア

B 作用素作用素のボタンを使って、基本作用素を挿入します。

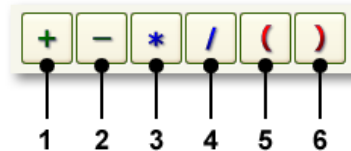


イラスト 1.4: 公式シート作用素

- 1 加算
- 2 減算
- 3 乗算
- 4 除算
- 5 括弧を開く
- 6 括弧を閉じる

C **デフォルトヘルプ**このエリアには、選択された機能に関する基本的なヘルプ情報が表示されます。

D **列番号参照**しやすくするため、各列には列番号が含まれています。現在アクティブな列には、数字の前に三角ポインタが存在しています。

E-G **公式**それぞれの公式には名前、本体、および単位があります。このエリアは「公式エディタ」とも呼ばれます。

H **追加ヘルプ**より詳細なヘルプ情報を示すため、追加ヘルプのエリアをオプションとして使用することができます。

追加ヘルプを表示したり、非表示にすることが必要となる場合があります。

追加ヘルプを表示または非表示にする方法：

- 公式エリアの上に存在するグリップをクリックしてください。



1.3 定義

公式データベースは作用素と関数を使用します。作用素には変数が、関数にはパラメータがそれぞれ必要となります。

1.3.1 定数

定数は、公式に直接入力される数または文字列による値です。数は、浮動小数点または整数による表記を用いて入力することができます。内部ですべての数値は浮動小数点数となります。文字列は、"テキスト"などの引用符と共に入力されます。

1.3.2 変数

変数は、データソース一覧に含まれるあらゆる変数（波形、数値、または文字列）となります。変数は、データソースを使用して選択することができます。前に定義した公式も変数として使用することが可能です。

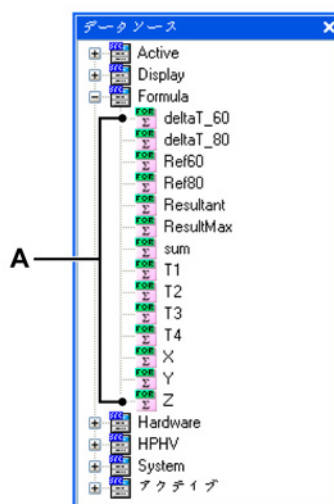


イラスト 1.5: 公式の結果を持つデータソースナビゲータ

A 公式の結果

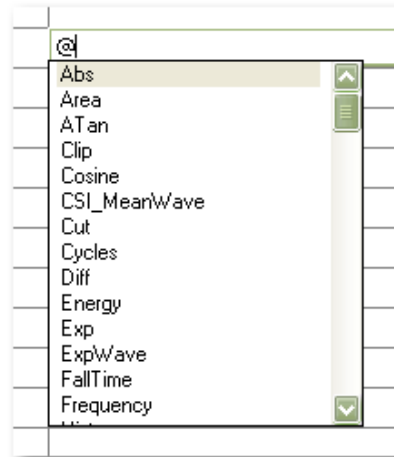
公式の本体では、変数はデータソースの変数の完全なパスによって参照されます。例えば上の図では、公式の結果 T2 は公式.T2として参照されます。

1.3.3 関数

関数は数多くのパラメータ（変数および/または定数）を必要とし、それらを使用して結果を生み出します。パラメータのタイプを結果のタイプは関数によって異なります。

関数の入力方法：

- 1 空の列の公式の本体を 1 回クリックしてそれを選ぶか、空の列の公式の本体を 2 回クリックして編集モードに入ります。
- 2 記号「@」を入力します。
 - 使用可能な関数の一覧が表示されます。



- 3 一覧から関数を選択するには、以下の手順に従います。
 - スクロールバーを使用して、使用可能な関数の一覧をスクロールします。
 - マウスのカーソルを関数に合わせてクリックしてください。
 - 矢印キーを使用して関数を選択し、ENTER を押します。

選択された関数の詳しい説明がヘルプエリアに表示されます。

公式に名前がないと、コメントの行として解釈されます。

1.4 レイアウトの修正

列を追加、削除、およびクリアすることで、列のレイアウトを修正することができます。 ツールを直接使用するか、右クリックを用いることで同じコマンドでコンテキストメニューを呼び出すことができます。



1.4.1 列を追加、削除、およびクリアする

列を追加、削除、およびクリアすることができます。 一つの列または複数の列でこれを行うことができます。

複数の列の選択方法：



- 連続する複数の列の選択方法：
 - 最初の列をクリックし、最後の列にドラッグします。
 - 最初の列をクリックして SHIFT を押した状態に維持し、最後の列をクリックします。
- 連続しない複数の列を選択するには、Ctrl キーを押した状態に維持した上でそれぞれの列をクリックします。

一つ以上の列を選択する方法：

- 1 列を一つ以上選択してください。
 - 列ポインタが選択された最後の列に移動します。
- 2 適切なツールをクリックします。
 - 上に列を挿入  を使って、現在選択されている列の上に列を挿入します。
 - 下に列を挿入  を使って、現在選択されている列の下に列を挿入します。
 - 空の列が挿入されます。

一つ以上の列を削除する方法：

一つ以上の列を削除して、以下を実施してください。

- 列と内容を削除するには、**選択した列を削除**  ツールをクリックしてください。
- 列に含まれる内容だけを削除する場合は、**選択した列をクリア**  ツールをクリックしてください。
- 確認ダイアログが表示されたら OK をクリックしてください。

フィールドのクリア方法：

以下の手順に従って、単一のフィールドをクリアすることができます。

- クリアしたいフィールドを選択し、**Del** を押します。

- クリアしたいフィールドを選択し、**Enter** を押します。編集のためにフィールドが開かれ、テキスト全体が選択されます。**Del** を押します。

1.4.2 移動

タブキーと矢印キーを使用してフィールドの中を移動することができます。

1.5 公式の作成

公式とコメントは容易に作成することができます。様々な自動記入技術が使用されることで、入力を何度も行わなくても公式を定義する作業がスムーズに行えます。このシステムが採用されることで、エラーが発生する可能性も低減されます。

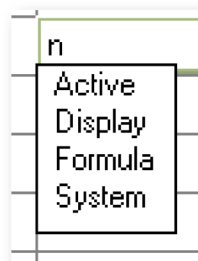
公式エディタは、記録を示す列を持つレイアウトに類似したデータベースを持っています。各列には、コメントと公式を入力する編集可能な3つのフィールドが存在します。

1.5.1 コメントの入力

公式エディタにコメントを入力して、作成する公式を明確化することができます。コメントは緑色で表示されます。コメントの列には名前は存在しません。

コメントの入力方法：

- 1 空の列の公式の本体を1回クリックしてそれを選ぶか、空の列の公式の本体を2回クリックして編集モードに入ります。
- 2 テキストの入力を開始してください。最初の文字を入力すると一覧が表示されますのでご注意ください。



- 3 この一覧は無視してください。最初の単語を入力し終わるとこの一覧は消えます。最後まで入力を行ってください。
- 4 入力が完了したら **Enter** を押すか、公式エディタの他の場所でクリックしてください。

コメントの列の名前のコラムにテキストを入力しないでください。

1.5.2 公式の入力

公式とは、既存のデータソース、変数、および定数から新しい波形、数値、または文字列を作成するための規則です。公式は、データソース、定数、および関数を含む数式となります。括弧を使用して作用素の優先度を変更することができます。

公式には名前、本体、および単位があります。公式は青色（既知のキーワード）と黒色で表示されます。

公式の入力方法：

以下を実施してください。

- 1 名前フィールドに名前を入力します。
 - a 空の列の名前のフィールドを 1 回クリックしてそれを選ぶか、空の列の名前のフィールドを 2 回クリックして編集モードに入ります。
 - b 説明的な名前を入力してください。
- 2 以下の手順に従って、公式の本体フィールドにナビゲートしてください。
 - TAB を押します。
 - 右向きの矢印キーを押してください。
- 3 以下の技術を 1 つ以上使用して、公式の本体を入力します。
 - 記号「@」を入力して使用可能な関数の一覧を表示させ、希望する関数を選択します。
 - 文字を入力します。使用可能なデータソースルートを含むドロップダウンリストが表示されます。ルートを選択して、点を入力します。選択されたルートの枝を含む一覧が新たに表示されます。枝を選択して点などを入力し、希望する変数またはデータソースの選択を完了させます。
 - 変数/データソースのフルパスを直接入力します。
 - 作用素と括弧を使用してより複雑な公式を作成します。
 - 利用可能な場合は、ヘルプエリアにヘルプテキストが表示されます。
- 4 完了したら単位のフィールドにナビゲートして、単位を適宜入力してください。



ヒント

公式データベース内の公式に、直接データソースをドラッグすることもできます。これによって、定数や変数を素早く関数に挿入することができ、その変数までの完全なパスを知っている必要はありません。例えば、カーソル X 位置をただ公式までドラッグすればよく、`Display.Display1.Cursor1.XPosition` のような完全なパスをタイプしなくてもいいわけです。

1.6 公式メニュー

公式メニューには、公式のファイルハンドリングに関連したコマンドが一覧表示されます。レイアウトと内容を管理するには、公式エディタの左側のタスクペインに存在するツールを使用してください。

公式メニューはダイナミックメニューで、公式シートが最も上にあるとき、つまり表示されているときにのみ使用することができます。

メニューは、公式を個別のファイルに保存する方法も提供します。一般的に、公式データベースの設定は以下の通りになっています。

- 公式シートで指定する、すべての公式/機能で構成されます。
- ファイル拡張子.pFormulas を付けて個別のファイルに保存することができます。
- ワークベンチが保存されるときに、記録の一部として自動的に保存されます。
- ワークベンチ全体を構成する一部として自動的に読み込まれます。
- ワークベンチまたは記録から個別の設定として抽出および読み出すことができます。
- ワークベンチまたは記録に個別の設定として保存することができます。

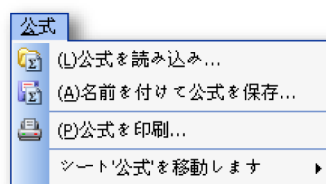



イラスト 1.6: 公式メニュー

1.6.1 公式をロード

様々なソースから公式を読み込むことができます。

公式を読み込む方法：

外部ソースから公式を読み込むには、以下の手順に従ってください。

- 1 以下のいずれかを実施してください。
 - 公式メニューで**公式を読み込む...**をクリックしてください。
 - ツールバーに存在していれば、**公式を読み込む...**ボタンをクリックしてください。


- 2 表示される公式を読み込むダイアログで、必要に応じてファイルのタイプを選択してください。
 - 公式ファイル (*.pFormulas)
 - 仮想ワークベンチからの公式 (*.pVWB)
 - 実験からの公式 (*.pNRF)
- 3 使用したいファイルを選択してください。
- 4 **開く**をクリックしてください。

1.6.2 公式を保存する

公式を読み込むときと同じ方法で、公式を保存することができます。また、既存の仮想ワークベンチや実験に保存することもできます。こうすることで、そのファイルに存在する公式が置き換えられます。その他のデータは変更されません。

公式の保存方法：


外部ファイルに公式を保存するには、以下の手順に従ってください。

- 1 以下のいずれかを実施してください。
 - **公式メニュー**で、**公式に名前を付けて保存...**をクリックしてください。
 - ツールバーに存在していれば、**公式に名前を付けて保存ボタン**をクリックしてください。
- 2 表示される**公式に名前を付けて保存**ダイアログで、必要に応じてファイルのタイプを選択してください。
 - 公式ファイル (*.pFormulas)
 - 仮想ワークベンチからの公式 (*.pVWB)
 - 実験からの公式 (*.pNRF)
- 3 保存または置き換えたいファイルを選択するか、新しいファイルの名前を入力してください。
- 4 **保存**をクリックしてください。

1.6.3 公式を印刷

プリンタを使用して公式の内容を印刷することが可能です。

公式を印刷する方法：

- 1 以下のいずれかを実施してください。
 - **公式メニュー**で、**公式を印刷...**をクリックしてください。
 - ツールバーに存在していれば、**公式を印刷...**ボタンをクリックしてください。
- 2 表示される印刷ダイアログでプリファレンスを入力してください。
- 3 **印刷**をクリックしてください。

1.6.4 シートを移動する

複数のワークブックのオプションがインストールされていれば、公式シートを別のワークブックに移動することができます。

公式シートを別のワークブックに移動する方法：

- 1 公式メニューで、シート「公式」を移動する、▶の順に選択します。
- 2 表示されるサブメニューで、ワークブックを選択します。

2 公式データベース機能

2.1 一般事項

本書では、Perception 公式データベースで使用できるすべての関数を詳細に説明しています。

公式データベースの説明については、ユーザーマニュアルの該当する項を参照してください。

パルス特性、測定、および分析に関する詳しい説明は、「パルス測定および分析」ページ 184 の章に記載しています。

公式データベースに含まれる公式は、以下のように定義されます。

出力 = 公式

出力は、データソースナビゲータの中で変数として使用することが可能です。これは、公式の枝に含まれる項目として見つけることが可能です。

データベースの中での物理的な順序に関係なく、出力は別の関数のパラメータとして使用することができます。

紙の上に記録するようにして、公式を入力することができます。標準的な数学的規則を適用することができます。

例外

「@」関数のパラメータとして数式は認められません。

例：

	Angle	=	33
正	AngleRad	=	System.Constants.Pi * Formula.Angle / 180
	CosAngle	=	@Cos (Formula.AngleRad)
誤	CosAngle	=	@Cos (System.Constants.Pi * Formula.Angle / 180)

ノート

すべての関数は、静的な、単一の時間ベース、単一のスイープデータに作用します。これらの関数は、複数の時間ベース、および/または複数のスイープデータにも作用します。しかし、使用される関数の性質上、結果は予想することができません。いくつかの関数は、ダイナミックな (リアルタイムの) データにも作用します。このことは、関数の説明で適宜指摘しています。

2.2 概要

この項では、名前、短い説明、およびページ番号を示しながら、すべての関数についてアルファベット順に概説します。

2.2.1 関数の概要

名前	記述
+ (加算)	2つの数式を加算: "+ (加算)" ページ 29
- (減算)	2つの数式を減算: "- (減算)" ページ 31
* (乗算)	2つの数式を乗算: "*" (乗算)" ページ 33
/ (除算)	2つの数式を除算: "/" (除算)" ページ 35
- (反転)	数式を反転: "- (単項マイナス)" ページ 37
@Abs	数値または波形の絶対値: "@Abs" ページ 38
@And	論理 AND: "@And" ページ 39
@Area	波形曲線の下領域: "@Area" ページ 40
@ATan	逆正接を計算します: "@ATan" ページ 42
@BlockFFT	データブロックの主要周波数を計算します: "@BlockFFT" ページ 43
@Clip	波形の振幅をクリップします: "@Clip" ページ 45
@Cos	余弦を計算します: "@Cos" ページ 46
@CurveFitting	既存の波形に線形または放物線状の波形をフィッティングさせます: "@CurveFitting" ページ 47
@Cut	波形の特定のセグメントを切り取ります: "@Cut" ページ 49
@Cycles	一つの波形のサイクル数をカウントします: "@Cycles" ページ 51
@Diff	波形を微分します: "@Diff" ページ 53
@Energy	波形曲線の下領域のエネルギーを計算します: "@Energy" ページ 55
@EqualTo	Equal-to (等しい) 評価: "@EqualTo" ページ 57
@Exp	指数: 入力の累乗指数 (底 e) を計算します: "@Exp" ページ 58
@ExpWave	指数波形を生成します: "@ExpWave" ページ 59
@FallTime	波形におけるパルスの立ち下り時間を決定します: "@FallTime" ページ 61
@FilterButterworthLP	入力信号をフィルタリングします: "@FilterButterworthLP" ページ 63
@FilterButterworthHP	入力信号をフィルタリングします: "@FilterButterworthHP" ページ 65

名前	記述
@FilterButterworthBP	入力信号をフィルタリングします : "@FilterButterworthBP" ページ 67
@FilterButterworthBS	入力信号をフィルタリングします : "@FilterButterworthBS" ページ 69
@FilterBesselLP	入力信号をフィルタリングします : "@FilterBesselLP" ページ 71
@FilterBesselHP	入力信号をフィルタリングします : "@FilterBesselHP" ページ 73
@FilterBesselBP	入力信号をフィルタリングします : "@FilterBesselBP" ページ 75
@FilterBesselBS	入力信号をフィルタリングします : "@FilterBesselBS" ページ 77
@FilterChebyshevLP	入力信号をフィルタリングします : "@FilterChebyshevLP" ページ 79
@FilterChebyshevHP	入力信号をフィルタリングします : "@FilterChebyshevHP" ページ 82
@FilterChebyshevBP	入力信号をフィルタリングします : "@FilterChebyshevBP" ページ 84
@FilterChebyshevBS	入力信号をフィルタリングします : "@FilterChebyshevBS" ページ 86
@Frequency	波形の周波数を決定します : "@Frequency" ページ 89
@GreaterEqualThan	Greater-than-or-equal-to (以上) 評価 : "@GreaterEqualThan" ページ 91
@GreaterThan	Greater-than (超える) 評価 : "@GreaterThan" ページ 92
@Histogram	振幅ヒストグラムを計算します : "@Histogram" ページ 93
@IIF	条件付き結果 : "@IIF" ページ 95
@Integrate	波形を積分します : "@Integrate" ページ 97
@IntLookUp	波形データを外部変換表のインデックスとして 使用することで波形を変更します : "@IntLookUp" ページ 98
@IntLookUp12	波形データを変換表のインデックスとして使用 することで波形を変更します : 12ビットデータ のために最適化されます : "@IntLookUp12" ページ 101
@Join	複数の波形を連結 : "@Join" ページ 103
@Length	波形の長さ (サンプルで) を戻します : "@Length" ページ 105
@LessEqualThan	Less-than-or-equal-to (以下) 評価 : "@LessEqualThan" ページ 106
@LessThan	Less-than (未滿) 評価 : "@LessThan" ページ 107

名前	記述
@Ln	自然対数を計算します : "@Ln" ページ 108
@Log	対数の底 10 を計算します : "@Log" ページ 110
@Max	波形の最大値 (振幅) を決定します : "@Max" ページ 111
@MaxNum	一定範囲の数値の最大値を決定します : "@MaxNum" ページ 113
@MaxPos	波形の最大値の位置を戻します : "@MaxPos" ページ 114
@Mean	波形の平均値を計算します : "@Mean" ページ 116
@Min	波形の最小値 (振幅) を決定します : "@Min" ページ 118
@MinNum	波形の最小値の位置を戻します : "@MinNum" ページ 120
@MinPos	絶対最小値の (時間的な) 位置を戻します : "@MinPos" ページ 121
@NextHillPos	波形における次のローカル最大値の位置を決定します : "@NextHillPos" ページ 123
@NextLvlCross	指定された信号レベルで波形の次の交差の位置を決定します : "@NextLvlCross" ページ 125
@NextValleyPos	波形における次のローカル最小値の位置を決定します : "@NextValleyPos" ページ 127
@Noise	ノイズを含む波形を生成します : "@Noise" ページ 129
@Not	論理 NOT : "@Not" ページ 130
@Or	論理 OR : "@Or" ページ 131
@Period	波形の期間を決定します : "@Period" ページ 132
@Pow	累乗法、累乗指数で累乗された底 : "@Pow" ページ 134
@PrevHillPos	波形における前のローカル最大値の位置を決定します : "@PrevHillPos" ページ 135
@PrevLvlCross	指定された信号レベルで波形の前の交差の位置を決定します : "@PrevLvlCross" ページ 137
@PrevValleyPos	波形における前のローカル最小値の位置を決定します : "@PrevValleyPos" ページ 139
@PulseWidth	波形におけるパルスの幅を決定します : "@PulseWidth" ページ 143
@Ramp	線形ランプの波形を生成します : "@Ramp" ページ 145
@ReadAsciiFile	ASCII (テキスト) ファイルから波形データを読み込みます : "@ReadAsciiFile" ページ 147

名前	記述
@Reduce	再度サンプリングを行って波形に存在するサンプルの数を減らします: "@Reduce" ページ 149
@RefCheck	1 つまたは 2 つの波形範囲に対する波形を検証します: "@RefCheck" ページ 150
@RemoveGlitch	波形に存在する不適切なサンプルを除去します: "@RemoveGlitch" ページ 152
@Res2	長さが 2 の累乗数になるように波形をサンプリングします: "@Res2" ページ 153
@RiseTime	波形におけるパルスの立ち上がり時間を決定します: "@RiseTime" ページ 154
@RMS	波形の二乗平均平方根の値を計算します: "@RMS" ページ 156
@SAEJ211Filter	SAE J211 の推奨事項に基づき波形をフィルタリングします: "@SAEJ211Filter" ページ 158
@Sin	正弦を計算します: "@Sin" ページ 159
@SineWave	正弦波を生成します: "@SineWave" ページ 160
@Smooth	選択された数のサンプルについて波形を円滑にします: "@Smooth" ページ 161
@Sqrt	平方根を計算します: "@Sqrt" ページ 163
@SquareWave	方形波を生成します: "@SquareWave" ページ 164
@StdDev	波形の標準偏差を計算します: "@StdDev" ページ 165
@Sweep	マルチスイープの記録の中でスイープを選択します: "@Sweep" ページ 167
@Tan	接線を計算します: "@Tan" ページ 168
@TriggerTime	トリガ位置を戻します: "@TriggerTime" ページ 169
@TriggerTimetoText	トリガ位置を時間-日付フォーマットの文字列に戻します: "@TriggerTimeToText" ページ 171
@TrueRMS	RMS 値を計算します: "@TrueRMS" ページ 173
@Value	指定された x 位置における波形の振幅値を戻します: "@Value" ページ 174
@XDelta	波形のサンプリング間隔を戻します: "@XDelta" ページ 175
@XDeltaHigh	マルチ時間ベースの記録における最大のサンプリング間隔を戻します: "@XDeltaHigh" ページ 176
@XDeltaLow	マルチ時間ベースの記録における最小のサンプリング間隔を戻します: "@XDeltaLow" ページ 177

名前	記述
@XFirst	波形における最初のサンプルの (トリガポイントに対する) x 座標を戻します : "@XFirst" ページ 178
@XLast	波形における最後のサンプルの (トリガポイントに対する) x 座標を戻します : "@XLast" ページ 179
@XShift	時間における波形をシフトします : "@XShift" ページ 180
@XYArray	X/Y の値の組み合わせ一覧から波形を生成します : "@XYArray" ページ 181
@YArray	Y の値の一覧から波形を生成します : "@YArray" ページ 183

3 算術演算

3.1 + (加算)

関数

左と右の数式を加算します。

構文

数式 1 + 数式 2

パラメータ

数式 1	左の数式
数式 2	右の数式

出力

結果は、左と右の数式の和となります。

記述

数式は以下の通りとなります。

- 波形変数
- 関数呼び出し
- 数値変数
- 定数値

正規作用素の優位性 (最初に乗算と除算を行い、次に加算と減算を行う) が適用されます。複雑な数式では、括弧を使用して作用素の優位性を変更することができます。

例

加算を使用した有効な数式の例を以下に示します。

- 2 + 3
- var1 + 4
- 5 + var2
- 1000 + @Noise(1E6; 1000)
- (Var1 + Var2) * (Var3 - Var4)

数値を加算すると、出力も数値になります。

波形と数値を加算すると、出力は波形になります。出力される波形のポイント (長さ) の数は入力される波形の長さに等しくなります。

2つの波形を加算すると、出力は2つの入力された波形のポイント毎の加算により形成される波形となります。出力される波形のxスケールは、左側波形のxスケールに等しくなります。出力される波形の長さは、入力される2つの波形のうち短いものの長さに等しくなります。意味のある結果を得るには、両方の波形のxスケールは等しくなければなりません。

以下も参照してください。

"* (乗算)" ページ 33, "-" (減算)" ページ 31, "/" (除算)" ページ 35

3.2 - (減算)

関数

左と右の数式を減算します。

構文

数式 1 - 数式 2

パラメータ

数式 1	左の数式
数式 2	右の数式

出力

結果は、左と右の数式の差となります。

記述

数式は以下の通りとなります。

- 波形変数
- 関数呼び出し
- 数値変数
- 定数値

正規作用素の優位性 (最初に乗算と除算を行い、次に加算と減算を行う) が適用されます。複雑な数式では、括弧を使用して作用素の優位性を変更することができます。

例

減算を使用した有効な数式の例を以下に示します。

- 3 - 2
- var1 - 4
- 5 - var2
- @SineWave(1E6; 1000; 1k) - 1
- (Var1 + Var2) * (Var3 - Var4)

数値を減算すると、出力も数値になります。

波形と数値を減算すると、出力は波形になります。出力される波形のポイント (長さ) の数は入力される波形の長さに等しくなります。

2つの波形を減算すると、出力は2つの入力された波形のポイント毎の減算により形成される波形となります。出力される波形のxスケールは、左側波形のxスケールに等しくなります。出力される波形の長さは、入力される2つの波形のうち短いものの長さに等しくなります。意味のある結果を得るには、両方の波形のxスケールは等しくなければなりません。

以下も参照してください。

"* (乗算)" ページ 33, "+ (加算)" ページ 29, "/" (除算)" ページ 35

3.3 * (乗算)

関数

左と右の数式を乗算します。

構文

数式 1 * 数式 2

パラメータ

数式 1	左の数式
数式 2	右の数式

出力

結果は、左と右の数式の積となります。

記述

数式は以下の通りとなります。

- 波形変数
- 関数呼び出し
- 数値変数
- 定数値

正規作用素の優位性 (最初に乗算と除算を行い、次に加算と減算を行う) が適用されます。複雑な数式では、括弧を使用して作用素の優位性を変更することができます。

例

乗算を使用した有効な数式の例を以下に示します。

- $2 * 3$
- $\text{var1} * 4$
- $5 * \text{var2}$
- $10 * @\text{SineWave}(1\text{E}6; 1000; 1\text{k})$
- $(\text{Var1} - \text{AvgVar1}) * (\text{Var2} - \text{AvgVar2})$

数値を乗算すると、出力も数値になります。

波形と数値を乗算すると、出力はスケーリングされた波形になります。出力される波形のポイント (長さ) の数は入力される波形の長さに等しくなります。

2つの波形を乗算すると、出力は2つの入力された波形のポイント毎の乗算により形成される波形となります。出力される波形のxスケールは、左側波形のxスケールに等しくなります。出力される波形の長さは、入力される2つの波形のうち短いものの長さに等しくなります。意味のある結果を得るには、両方の波形のxスケールは等しくなければなりません。

以下も参照してください。

"+ (加算)" ページ 29, "- (減算)" ページ 31, "/" (除算)" ページ 35

3.4 / (除算)

関数

左と右の数式を除算します。

構文

数式 1 / 数式 2

パラメータ

数式 1 左の数式 (被除数)

数式 2 右の数式 (除数)

出力

結果は、左と右の数式の商となります。

記述

数式は以下の通りとなります。

- 波形変数
- 関数呼び出し
- 数値変数
- 定数値

正規作用素の優位性 (最初に乗算と除算を行い、次に加算と減算を行う) が適用されます。複雑な数式では、括弧を使用して作用素の優位性を変更することができます。

例

除算を使用した有効な数式の例を以下に示します。

- 3 / 2
- var1 / 4
- 5 / var2
- @SineWave (1E6; 1000; 1k) / 10
- (Var1 + Var2) / (Var3 - Var4)

数値を除算すると、出力も数値になります。

波形と数値を除算すると、出力は波形になります。出力される波形のポイント (長さ) の数は入力される波形の長さに等しくなります。

2つの波形を除算すると、出力は2つの入力された波形のポイント毎の除算により形成される波形となります。出力される波形のxスケールは、左側波形のxスケールに等しくなります。出力される波形の長さは、入力される2つの波形のうち短いものの長さに等しくなります。意味のある結果を得るには、両方の波形のxスケールは等しくなければなりません。

除数が0に等しい数値であるとき、または分母が値0(ゼロ)と共にサンプルを含む波形であるとき、定義されていない(不明の)値を戻します。

以下も参照してください。

"* (乗算)" ページ 33, "+ (加算)" ページ 29, "- (減算)" ページ 31

3.5 - (単項マイナス)

関数

数式の記号を反転します。

構文

- 数式

パラメータ

数式 反転する数式。

出力

結果は、数式に - 1 を乗算した数になります。

記述

数式は以下を含むあらゆる数式となります。

- 波形の変数
- 関数呼び出し
- 数値変数
- 定数値

正規作用素の優位性 (最初に乗算と除算を行い、次に加算と減算を行う) が適用されます。複雑な数式では、括弧を使用して作用素の優位性を変更することができます。

例

単項 - を使用した有効な数式の例を以下に示します。

- - 2
- - var1
- - @SineWave (1E6; 1000; 1k) - 1
- - ((Var1 + Var2) * (Var3 - Var4))

数値を反転すると、出力は数値になります。

波形を反転すると、出力は無効な波形になります。出力される波形のポイント (長さ) の数は入力される波形の長さに等しくなります。

以下も参照してください。

"* (乗算)" ページ 33, "+ (加算)" ページ 29, "- (減算)" ページ 31, "/" (除算)" ページ 35

4 リファレンスガイド

4.1 @Abs

関数

パラメータの絶対値を計算します。

構文

@Abs (Par)

パラメータ

Par 入力波形または数値

出力

波形または数値の絶対値

記述

入力波形または数値の絶対値を計算します。正の値は変わることなく、負の値は記号を変えます。この関数を使って信号を修正するか、結果を正の値にすることができます。

例

以下の例では、正弦波を生成し、この信号を修正します。

```
Signal = @SineWave(1E6; 1000; 1k)
Rectif = @Abs(Formula.Signal)
```

4.2 @And

関数

入力パラメータについて論理 AND の評価を実施します。

構文

@And (パラメータ 1; ...; パラメータ N)

パラメータ

パラメータ 1 番号 : AND 評価のために使用される最初のパラメータ
 パラメータ N AND 評価のために使用される最後のパラメータ N >= 2 とする。

出力

出力は 1 または 0 です。

記述

@AND 関数は、入力パラメータについて論理 AND の評価を実施します。評価の内容に応じて、結果 1 または 0 になります。0 に等しくない数値は論理「True (真)」に対応し、数値 0 は論理「False (偽)」に対応します。

AND 関数の真の表は以下の通りです。

パラメータ 1	パラメータ 2	結果
真	真	真
真	偽	偽
偽	真	偽
偽	偽	偽

例

以下に例とその戻り値の一覧を示します。

```

AndExamp11 = @And(1; 1; 1)    => 1 (=true)
AndExamp12 = @And(1; 4; 10)  => 1 (=true)
AndExamp13 = @And(1; 4; 0)   => 0 (=false)
AndExamp13 = @And(0; 0; 0)   => 0 (=false)

```

以下も参照してください。

"@Not" ページ 130 および "@Or" ページ 131

4.3 @Area

関数

波形曲線の下領域を計算します。

構文

@Area (波形)

@Area (波形; 開始)

@Area (波形; 開始; 終了)

パラメータ

波形	曲線の下領域が計算される入力波形
開始	番号：セグメントの開始。
終了	番号：セグメントの終了。

出力

出力は数値です。

記述

曲線の下領域は、以下の公式を使って計算されます。

$$\text{Area} = \left[\left[\sum_{n=n_1}^{n_2} y(n) \right] - \frac{y(n_1) + y(n_2)}{2} \right] \cdot \Delta X$$

$n_1 = x \geq$ 開始での最初のサンプル

$n_2 = x \leq$ 終了での最後のサンプル

$\Delta x = x - 2$ つのサンプルの差

セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート *開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。*

サンプル間の曲線の線形補間を想定して、曲線の数値の積分が実行されます。

例

以下の例では 50 Hz の正弦波を生成し、信号の前半期間の曲線の下領域を計算します。

```
Signal = @SineWave(50k; 1000; 50)
Area    = @Area(Formula.Signal; 0; 10m)
```

以下も参照してください。

"@Energy" ページ 55 および "@Mean" ページ 116

4.4 @ATan

関数

入力パラメータの逆正接を計算します。

構文

@ATan (*Par*)

パラメータ

Par 入力波形または数値

出力

入力の逆正接を含む波形または数値

記述

逆正接の関数は、正接が独立変数に等しくなる角度を戻します。角度はラジアンで戻されます。逆正接は、正接の逆三角関数です。

例

以下の例では、ATan (1) を 4 で乗じることで Pi を計算します。

```
Pi = 4 * @ATan(1)
```

以下も参照してください。

"@Cos" ページ 46、"@Sin" ページ 159 および"@Tan" ページ 168

4.5 @BlockFFT

関数

入力波形の**ブロック**毎に検出される最大周波数を示す波形を戻します。

構文

@BlockFFT (*波形*; *サイズ*; *スペース*)

@BlockFFT (*波形*; *サイズ*; *スペース*; *開始*)

@BlockFFT (*波形*; *サイズ*; *スペース*; *開始*; *終了*)

パラメータ

<i>波形</i>	入力波形
<i>サイズ</i>	番号：ミリ秒によるブロックサイズ
<i>スペース</i>	番号：ミリ秒による二つの連続するブロックの開始の間の空間設定
<i>開始</i>	番号：ブロック FFT 関数の開始位置
<i>終了</i>	番号：ブロック FFT 関数の終了位置

出力

周波数と時間を含む波形

記述

この関数は FFT アルゴリズムを使用してブロック毎の最大周波数を計算します。パラメータの大きさによって、ミリ秒によるブロックの大きさ（長さ）が決まります。スペースのパラメータによって、ミリ秒による二つの連続するブロックの開始の間の空間設定が決まります。

スペースとブロックの関係性については、以下の図を例として参照してください。

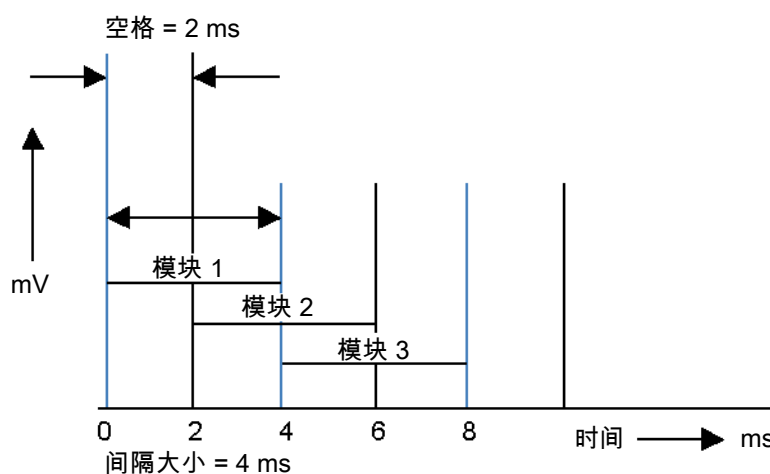


イラスト 4.1: 例 - 空間設定とブロックのサイズの間関係性

この関数は時間の関数としてブロック毎の単一の最大周波数を含む波形として出力されます。この出力される波形のサンプルの空間設定は、スペースパラメータに等しくなります。

セグメントの限界（開始と終了）を使用して、ブロック FFT の計算が行われる一定の範囲の波形を選択します。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

例

以下の例では、合成波形のブロック FFT を計算します。

```
Sine1 = 2 * @SineWave(1000k; 30k; 10k)
Sine2 = @SineWave(1000k; 30k; 15k)
Sine3 = 1.5 * @SineWave(1000k; 30k; 5k)
Signal = @Join(Formula.Sine1; Formula.Sine2;
               Formula.Sine3)
Result = @BlockFFT(Formula.Signal; 4; 2)
```

4.6 @Clip

関数

指定される下部区域および上部区域の間の波形をクリップします。

構文

@Clip (波形; 下部区域; 上部区域)

パラメータ

波形	振幅範囲がクリップされる波形
下部区域	番号：クリッピングのために使用される下部区域値
上部区域	番号：クリッピングのために使用される上部区域値

出力

下部区域と上部区域の間にクリップされるすべてのサンプル値を持つ波形

記述

波形にあるすべてのサンプルについて、クリッピング範囲の下部区域と上部区域について比較が行われます。サンプル値がこれら2つの値の間にあるときは、修正されません。サンプル値が上部区域よりも大きい場合、上部区域に設定されます。サンプル値が下部区域よりも小さい場合、下部区域に設定されます。

例

例では、1.2 V 正弦波を限界値 -1 ~ 1 にクリップし、入力オーバーフローをシミュレートします。

```
Signal      = 1.2 * @SineWave(20k; 1000; 50)
InpSignal   = @Clip(Formula.Signal; -1; 1)
```

以下の例では、加速信号が 150 g を超える間の累積時間を決定します。最初に信号を 150 g、および 150 g よりも 0.001 g 高い値の間にクリップするために使用される技術 クリップされた信号 150 が減じられ、クリップされた信号が倍数 1000 によって高められます。この技術を用いることで、加速が 150 g 未満のときは 0、および加速が 150 g を超えるときは 1 となります。この曲線の下領域は、150 g を超える間の累積時間です。

```
Accel      = 150 + @SineWave(20k; 100; 50)
Temp       = 1000 * (@Clip(Formula.Accel; 150; 150.001) - 150)
CumTime    = @Area(Formula.Temp)
```

以下も参照してください。

"@Cut" ページ 49

4.7 @Cos

関数

入力パラメータの余弦を計算します。

構文

@Cos (*Par*)

パラメータ

Par 入力波形または数値

出力

入力の余弦を含む波形または数値

記述

三角関数の余弦は、入力パラメータがラジアンのものであると想定された上で計算されます。波形パラメータが使用されると、各サンプルについて余弦が個別に計算されます。

例

以下の例では、度数で指定された変数「角度」の余弦を計算します。

```

Angle      = 33
AngleRad   = System.Constants.Pi * Formula.Angle / 180
CosAngle   = @Cos (Formula.AngleRad)

```

以下も参照してください。

"@ATan" ページ 42、"@Sin" ページ 159 および"@Tan" ページ 168

4.8 @CurveFitting

関数

線形補間または放物線回帰を使用して、入力信号からの一連のデータポイントに最も合う波形を戻します。

構文

@CurveFitting (波形)

@CurveFitting (波形; 順序)

@CurveFitting (波形; 順序; 開始間隔)

@CurveFitting (波形; 順序; 開始間隔; 終了間隔)

@CurveFitting (線形; 順序; 開始間隔; 終了間隔; 開始)

@CurveFitting (線形; 順序; 開始間隔; 終了間隔; 開始; 終了)

パラメータ

波形	入力波形
順序	番号： 回帰順序 1 = 線形、2 = 放物線 デフォルトは線形回帰です。
BeginIntv	番号： 補間法のために使用される間隔の開始 デフォルトでは、入力波形の開始が使用されます。
EndIntv	番号： 補間法のために使用される間隔の終了 デフォルトでは、入力波形の終了が使用されます。
開始	番号： 出力曲線が開始する時間 デフォルトでは、入力波形の開始時間が使用されます。
終了	番号： 出力曲線が終了する時間 デフォルトでは、入力波形の終了時間が使用されます。

出力

入力波形またはその一部の線形または放物線回帰である波形

記述

この関数は、入力信号からの一連のデータポイントに最も合う曲線を探します。パラメータに応じて、すべてのデータポイント、または入力信号からのデータポイントの限定された間隔のみが使用されます。

順序パラメータが 1 に設定されると、線形回帰が行われます。出力信号は直線になります。

順序パラメータが 2 に設定されると、放物線回帰が行われます。出力信号は放物線になります。

計算では最小二乗アルゴリズムが使用されます。

例

以下に例では、線形および放物線のフィッティングが作られます。

```
Signal      = @SineWave(8000; 8001; 5)
LineFit     = @CurveFitting(Formula.Signal; 1; 90m;
                          110m)
ParabolicFit = @CurveFitting(Formula.Signal; 2; 125m;
                          175m)
```


4.9 @Cut

関数

波形の特定の部分を切り取ります。

構文

@Cut (波形; 開始; 終了)

パラメータ

波形	セグメントの選択が行われる入力波形
開始	番号 : セグメントの開始。
終了	番号 : セグメントの終了。

出力

波形セグメント

記述

波形の特定の部分を切り取って、さらに処理を加えることができます。この関数は、開始および終了の値に基づいて出力のためのサンプルの数を計算します。最初のサンプルは、開始に最も近い x 座標を持つサンプルです。最後のサンプルは、終了に最も近い x 座標を持つサンプルです。

ノート 開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。セグメントの限界が波形の x 範囲の外に存在する場合、これらの値はその範囲に制限されます。

例

以下の例では、100 ms から 200 ms までの間における信号のセグメントを切り取ります。

```
Signal = @SineWave(10000; 10000; 100)
Segment = @Cut(Formula.Signal; 100m; 200m)
```

特定の範囲のサンプルが必要である場合、情報関数を使って選択できる特定の範囲を計算することができます。

以下の例では、前の信号の最初の 1024 ポイントを選択し、-1E20 を非常に小さい数値として使用します。

```
XEnd = @XBegin(Formula.Signal)
      + 1023 * @XDelta(Formula.Signal)
First1024 = @Cut(Formula.Signal; -1E20; Formula.XEnd)
```

以下も参照してください。

"@Join" ページ 103、"@Length" ページ 105、"@XFirst" ページ 178、"@XDelta"
ページ 175 および"@XLast" ページ 179

4.10 @Cycles

関数

波形または波形セグメントに含まれるサイクルの数を計算します。

構文

@Cycles (波形)

@Cycles (波形; 開始)

@Cycles (波形; 開始; 終了)

パラメータ

波形	サイクルの計算が行われる入力波形
開始	番号 : セグメントの開始
終了	番号 : セグメントの終了

出力

サイクル数

記述

@Cycles 関数は、波形においてサンプルのシーケンス (= サイクル) が発生する回数を計算します。最大と最小の振幅値の間の 50% レベルを使って平面交差の数を計算し、サイクルの数を得ます。

セグメントの限界 (開始と終了) を使用して、サイクルの計算が行われる一定の範囲のサンプルを選択します。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート *開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。*

例

以下の例では、一つの完全な信号のサイクル数、および「表示」という名称のディスプレイに存在する二つのカーソルの間のサイクル数を計算します。

```
Signal = @SineWave(10k; 10001; 50)
Cycles = @Cycles(Formula.Signal)

Start   = Display.Display.Cursor1.XPosition
End     = Display.Display.Cursor2.XPosition
```

```
Cycles_BC = @Cycles (Formula.Signal;  
                  Formula.Start; Formula.End)
```

以下も参照してください。

"@Frequency" ページ 89 および "@Period" ページ 132

4.11 @Diff

関数

波形を微分します。

構文

@Diff (波形)

パラメータ

波形 微分する波形

出力

微分された波形

記述

派生物は、入力値が変化するときの一つの関数がどのようにして変わるのに関する測定値です。簡単に言えば、派生物とはある特定の地点で起きる変化の規模であると考えられます。派生物を見つけるためのプロセスは、微分法と呼ばれます。また微分法は、信号の値ではなく傾きを決定するためにも使われます。傾きは、近接するサンプルの間の差を計算し、この差をサンプリングの間隔で除算することで求めることができます。

$$\text{Diff}(n) = \frac{y(n) - y(n-1)}{\Delta x} \quad \text{for } n = 2, \dots, N$$

$$\text{Diff}(1) = 0$$

Δx = x - 2 つのサンプルの差

N = サンプル数

微分法は、丸めの影響によって引き起こされる高周波ノイズおよびデジタル化の誤差などの高周波成分を増進させます。傾きをより正確に推定するには、(微分の前または後に) 波形の円滑化を行うことが推奨されます。

例

以下の例では、ノイズを発生させる正弦波を生成し、この信号を微分します。結果として生じる波形は円滑化され、正弦波の傾きをより正確に推定します。

```
Signal = @SineWave(10k; 1000; 50)
Differ = @Diff(Formula.Signal)
Slope = @Smooth(Formula.Differ; 7)
```

以下も参照してください。

"@Integrate" ページ 97 および "@Smooth" ページ 161

4.12 @Energy

関数

波形曲線の下でのエネルギーを計算します。

構文

@Energy (波形)

@Energy (波形; 開始)

@Energy (波形; 開始; 終了)

パラメータ

波形	曲線の下でのエネルギーが計算される入力波形
開始	番号 : セグメントの開始。
終了	番号 : セグメントの終了。

出力

出力は数値です。

記述

曲線の下でのエネルギーは、以下の公式を使って計算されます。

$$\text{Energy} = \left[\left[\sum_{n=n_1}^{n_2} y^2(n) \right] - \frac{y^2(n_1) + y^2(n_2)}{2} \right] \cdot \Delta x$$

$n_1 = x \geq$ 開始での最初のサンプル

$n_2 = x \leq$ 終了での最後のサンプル

$\Delta x = x - 2$ つのサンプルの差

セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。サンプル間の平方曲線の線形補間を想定して、数値の積分が実行されます。

例

以下の例では 50 Hz の正弦波を生成し、信号の一定期間のエネルギーを計算します。

```
Signal = @SineWave(20k; 1000; 50)  
E1      = @Energy(Formula.Signal; 0; 20m)
```

以下も参照してください。
"@Area" ページ 40 および "@RMS" ページ 156

4.13 @EqualTo

関数

この関数は、2つの数値入力パラメータについて `equal-to (等しい)` (=) 評価を実施します。

構文

`@EqualTo (パラメータ 1; パラメータ 2)`

パラメータ

パラメータ 1 番号: 評価のために使用される最初のパラメータ

パラメータ 2 番号: 評価のために使用される 2 つ目のパラメータ

出力

出力は 1 または 0 です。

記述

`EqualTo (等しい)` の関数は、入力パラメータについて「`equal-to (等しい)`」評価を実施します。パラメータ 1 = パラメータ 2 であるとき戻り値は 1 (真) となり、その他の場合は 0 (偽) となります。

一般的に `EqualTo (等しい)` の関数は、`IIF` 関数と組み合わせて使用されます。

例

以下の例では入力パラメータの比較を行い、それに応じて結果を出します。

```
EqualToExample1 = @EqualTo(5; 5) => 1 (true)
EqualToExample2 = @EqualTo(12; 10) => 0 (false)
IIFExample      = @IIF(EqualToExample2; "TRUE"; "FALSE")
```

以下も参照してください。

"@IIF" ページ 95、"@GreaterEqualThan" ページ 91、"@GreaterThan" ページ 92、"@LessEqualThan" ページ 106 および"@LessThan" ページ 107

4.14 @Exp

関数

指数関数、数学演算、書き込まれた e^n 、以下の 2 つパラメータを含む：底「e」(2.7...) および指数「n」

構文

@Exp (Par)

パラメータ

Par 入力波形または数値

出力

入力の累乗法 (底 e) を含む波形または数値

記述

指数関数は入力の累乗指数 (底 e) を計算します。波形パラメータが使用されると、各サンプルについて指数関数が個別に計算されます。この関数は、@Ln の逆関数です。

例

指数関数は@Pow 関数を使用して遂行することも可能です。以下の公式は、入力と呼ばれる公式データベース変数を想定することで等しくなります。

```
Result = @Exp (Formula.Input)
```

```
Result = @Pow (System.Constants.e; Formula.Input)
```

以下の例では、システム変数システム.定数.e の代替手段を示します。

```
Euler = @Exp (1)
```

以下も参照してください。

"@ExpWave" ページ 59、"@Ln" ページ 108 および"@Pow" ページ 134

4.15 @ExpWave

関数

指数関数を含む波形を生成します。

構文

@ExpWave (速度; カウント; アルファ; ベータ)

@ExpWave (速度; カウント; アルファ; ベータ; X0)

パラメータ

速度	番号 : サンプル周波数
カウント	サンプルの数
アルファ	番号 : 振幅の乗数
ベータ	番号 : 指数乗数
X0	番号 : 指数乗数の修飾因子

出力

出力は指数関数的に成長する信号を含む波形となります。

記述

この関数は、公式を使用した波形を生成します。

$$f(x) = \alpha \cdot e^{(\beta \cdot (x-x_0))}$$

サンプリング周波数、サンプルの数、成長速度、および増倍率を指定することができます。生成された波形の長さは1ギガサンプル(1 000 000 000)に制限されます。

自然対数 $\ln(x)$ は指数関数の逆関数になります。

指数波動関数を生成できることを利用して、様々な波形を合成することができます。シミュレーションしたデータを、他の分析関数のための入力として使用することができます。

例

以下の例では、指数波形を生成します。

```
Signal = @ExpWave(1; 100; 2; 100m)
```

以下も参照してください。

"@Ln" ページ 108、"@Pulse" ページ 141、"@SineWave" ページ 160 および
"@SquareWave" ページ 164

4.16 @FallTime

関数

波形におけるパルスの立ち下り時間を決定します。

構文

@FallTime (波形)

@FallTime (波形; 開始)

@FallTime (波形; 開始; 終了)

パラメータ

波形	パルスの立ち下り区間を含む入力波形
開始	番号：セグメントの開始。
終了	番号：セグメントの終了。

出力

出力は数値です。開始と終了のパラメータが設定されているとき、この関数は「記録が行われる間」に機能します。結果は、開始と終了の間のデータが使用可能である場合に計算されます。

記述

立ち下り時間は、波形 (波形セグメント) でのパルスの最初の立ち下り区間における遠位地点 (90% の大きさの移行) と近位地点 (10% の大きさの移行) の間での時間差を使って計算されます。

セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

例

以下の例ではパルスを生成し、立ち下り時間の 90% ~ 10% の立ち下り時間を計算します。結果は 40 ms となります。

```

Sig1      = @Ramp(1k; 100; 0; 0)
Sig2      = @Ramp(1k; 51; 0; 10)
Sig3      = @Ramp(1k; 100; 10; 10)
Sig4      = @Ramp(1k; 51; 10; 0)
Signal    = @Join(Formula.Sig1; Formula.Sig2;
                  Formula.Sig3; Formula.Sig4; Formula.Sig1)

```

`Falltime = @FallTime(Formula.Signal)`

以下も参照してください。

"@PulseWidth" ページ 143 および "@RiseTime" ページ 154

4.17 @FilterButterworthLP

関数

入力信号を直接型 IIR 低パスバターワースフィルタでフィルタリングします。

構文

@FilterButterworthLP(信号; 次数; F_c)

@FilterButterworthLP(信号; 次数; F_c ; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
F_c	番号：ヘルツによるカットオフ周波数
フェーズレス	番号：フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR 低パスのバターワースフィルタリングを行います。

カットオフ周波数は、応答の振幅が -3 dB の周波数です。

詳細については、ページ 188 の **IIR フィルタ** の章を参照してください。

例

```
Signal = @FilterButterworthLP(Formula.Signal; 2; 200)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のようになります。

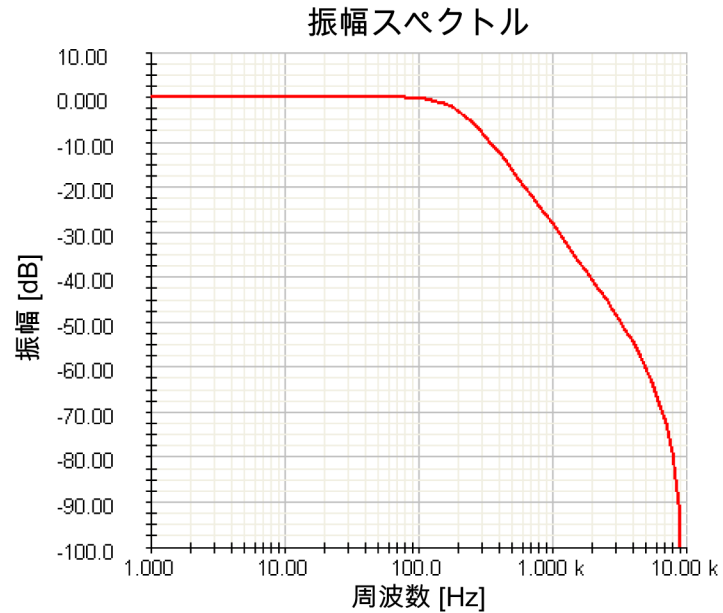


イラスト 4.2: 振幅スペクトル - FilterButterworthLP

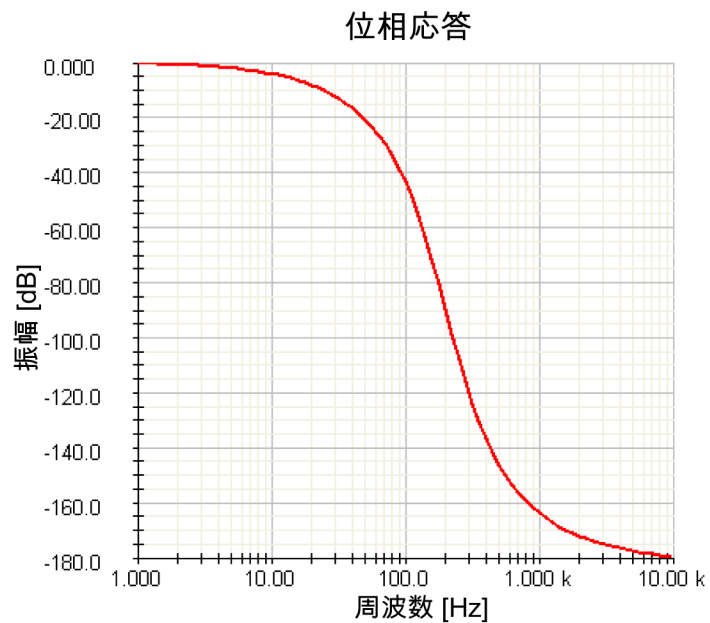


イラスト 4.3: 位相応答 - FilterButterworthLP

以下も参照してください。

"@FilterButterworthBS" ページ 69、"@FilterButterworthBP" ページ 67 および
"@FilterButterworthHP" ページ 65

4.18 @FilterButterworthHP

関数

入力信号を直接型 IIR 高パスバターワースフィルタでフィルタリングします。

構文

@FilterButterworthHP(信号; 次数; Fc)

@FilterButterworthHP(信号; 次数; Fc; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
Fc	番号：ヘルツによるカットオフ周波数
フェーズレス	番号：フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR 高パスのバターワースフィルタリングを行います。

カットオフ周波数は、応答の振幅が-3 dB の周波数です。

詳細については、ページ 188 の **IIR フィルタ** の章を参照してください。

例

```
Signal = @FilterButterworthHP(Formula.Signal; 2; 200)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

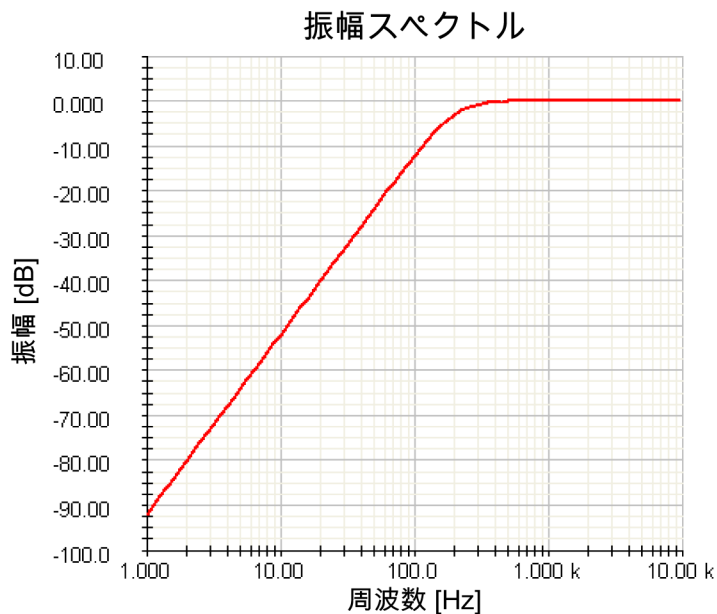


イラスト 4.4: 振幅スペクトル - FilterButterworthHP

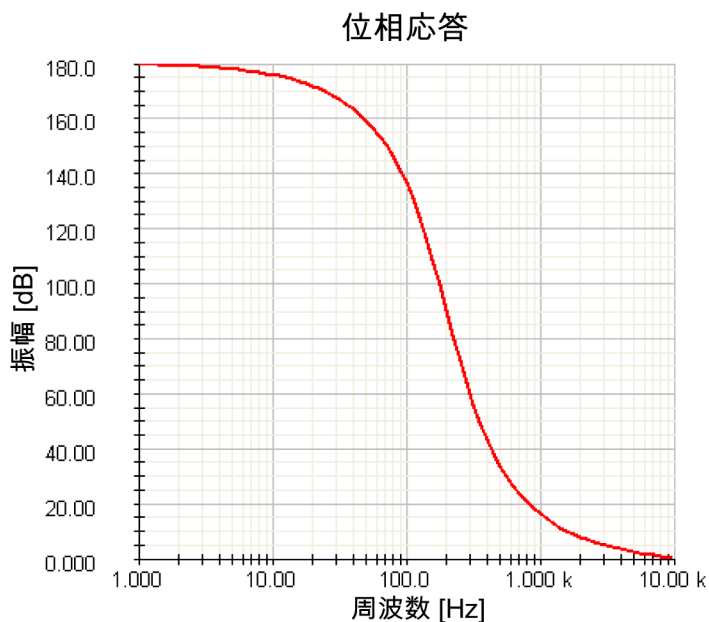


イラスト 4.5: 位相応答 - FilterButterworthHP

以下も参照してください。

"@FilterButterworthBS" ページ 69、"@FilterButterworthBP" ページ 67 および
"@FilterButterworthLP" ページ 63

4.19 @FilterButterworthBP

関数

入力信号を直接型 IIR バンドパスバターワースフィルタでフィルタリングします。

構文

@FilterButterworthBP (信号; 次数; 低域 F_c ; 高域 F_c)

@FilterButterworthBP (信号; 次数; 低域 F_c ; 高域 F_c ; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
低域 F_c	番号：ヘルツによる低域カットオフ周波数
高域 F_c	番号：ヘルツによる高域カットオフ周波数
フェーズレス	番号：フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR バンドパスのバターワースフィルタリングを行います。

カットオフ周波数は、応答の振幅が -3 dB の周波数です。

詳細については、ページ 188 の IIR フィルタの章を参照してください。

例

```
Signal = @FilterButterworthBP (Formula.Signal; 2; 200;
                               1000)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

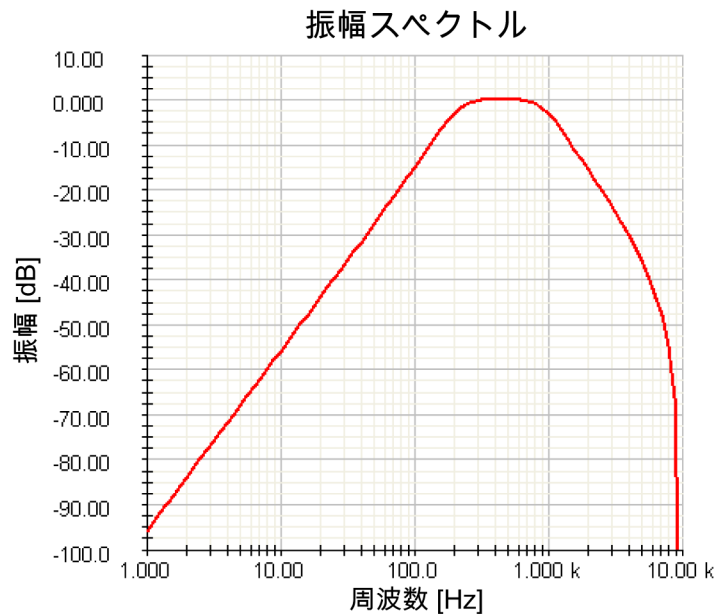


イラスト 4.6: 振幅スペクトル - FilterButterworthBP

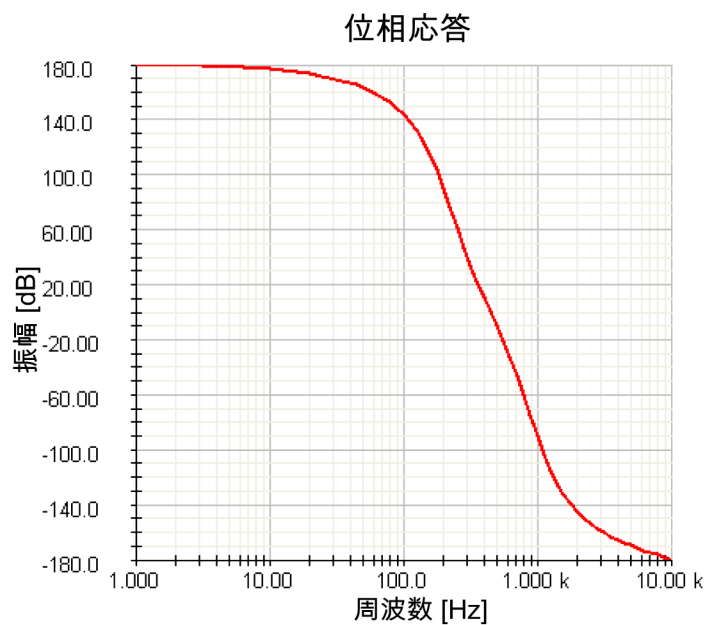


イラスト 4.7: 位相応答 - FilterButterworthBP

以下も参照してください。

"@FilterButterworthBS" ページ 69、"@FilterButterworthHP" ページ 65 および
"@FilterButterworthLP" ページ 63

4.20 @FilterButterworthBS

関数

入力信号を直接型 IIR バンドストップバターワースフィルタでフィルタリングします。

構文

@FilterButterworthBS(信号; 次数; 低域 F_c ; 高域 F_c)

@FilterButterworthBS(信号; 次数; 低域 F_c ; 高域 F_c ; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
低域 F_c	番号：ヘルツによる低域カットオフ周波数
高域 F	番号：ヘルツによる高域カットオフ周波数
フェーズレス	番号：フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR バンドストップのバターワースフィルタリングを行います。

カットオフ周波数は、応答の振幅が-3 dB の周波数です。

詳細については、ページ 188 の IIR フィルタの章を参照してください。

例

```
Signal = @FilterButterworthBS(Formula.Signal; 2; 200; 1000)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

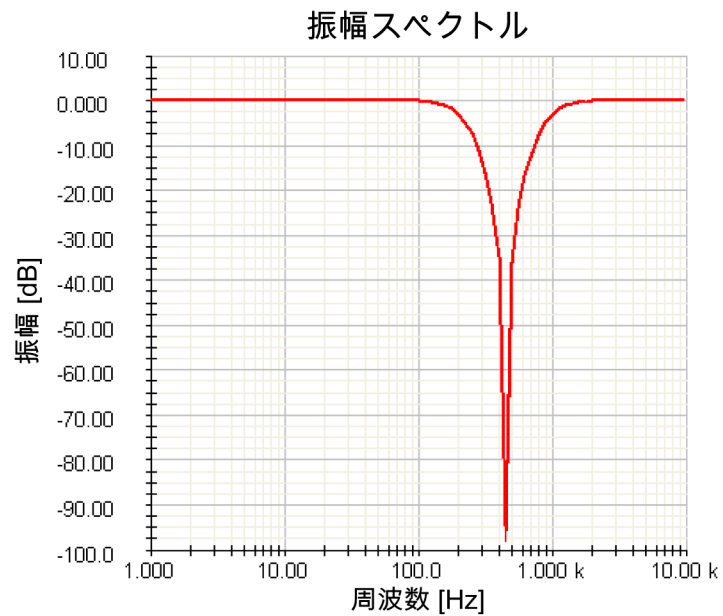


イラスト 4.8: 振幅スペクトル - FilterButterworthBS

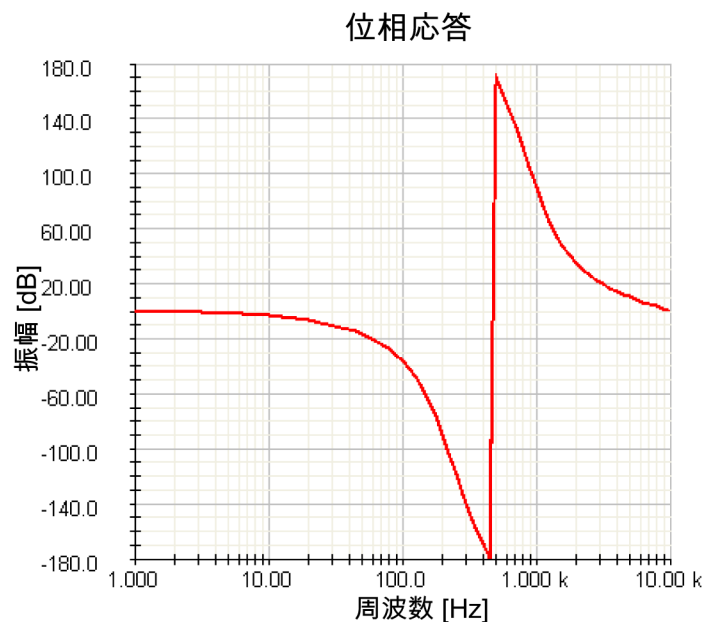


イラスト 4.9: 位相応答 - FilterButterworthBS

以下も参照してください。

"@FilterButterworthBP" ページ 67、"@FilterButterworthHP" ページ 65 および
"@FilterButterworthLP" ページ 63

4.21 @FilterBesselLP

関数

入力信号を直接型 IIR 低パスベッセルフィルタでフィルタリングします。

構文

@FilterBesselLP (信号; 次数; F_c)

@FilterBesselLP (信号; 次数; F_c ; フェーズレス)

パラメータ

信号	入力波形
順序	番号 : フィルタ次数
F_c	番号 : ヘルツによるカットオフ周波数
フェーズレス	番号 : フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR 低パスのベッセルフィルタリングを行います。

カットオフ周波数は、応答の振幅が -3 dB の周波数です。

詳細については、ページ 188 の **IIR フィルタ** の章を参照してください。

例

```
Signal = @FilterBesselLP (Formula.Signal; 2; 200)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

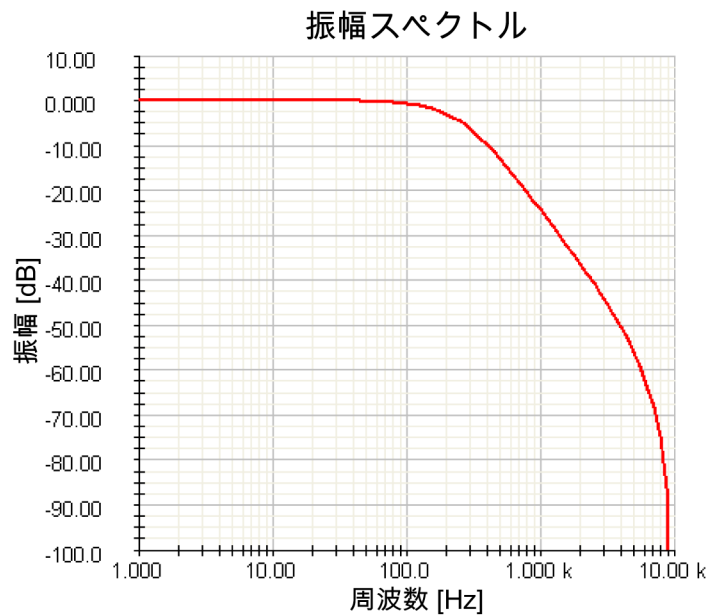


イラスト 4.10: 振幅スペクトル - FilterBesselLP

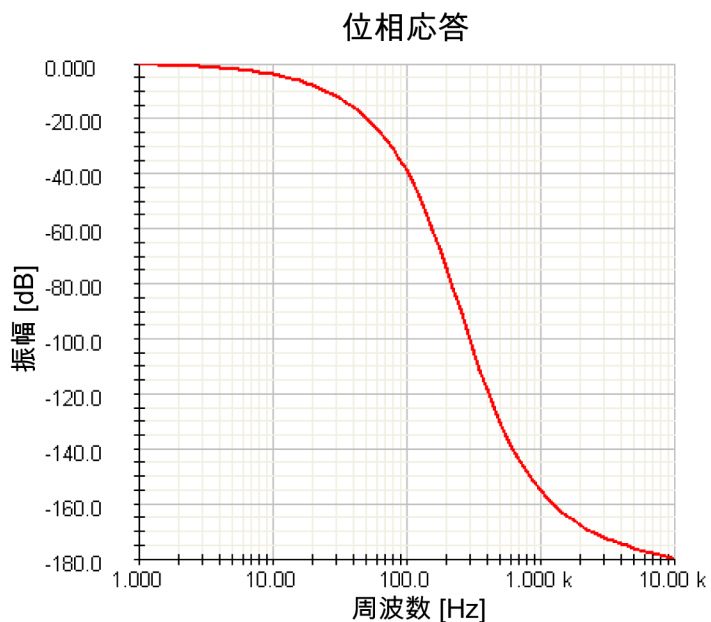


イラスト 4.11: 位相応答 - FilterBesselLP

以下も参照してください。

"@FilterBesselBP" ページ 75、"@FilterBesselBS" ページ 77 および
"@FilterBesselHP" ページ 73

4.22 @FilterBesselHP

関数

入力信号を直接型 IIR 高パスベッセルフィルタでフィルタリングします。

構文

@FilterBesselHP (信号; 次数; F_c)

@FilterBesselHP (信号; 次数; F_c ; フェーズレス)

パラメータ

信号	入力波形
順序	番号: フィルタ次数
F_c	番号: ヘルツによるカットオフ周波数
フェーズレス	番号: フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR 高パスのベッセルフィルタリングを行います。

カットオフ周波数は、応答の振幅が -3 dB の周波数です。

詳細については、ページ 188 の **IIR フィルタ** の章を参照してください。

例

```
Signal = @FilterBesselHP (Formula.Signal; 2; 200)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

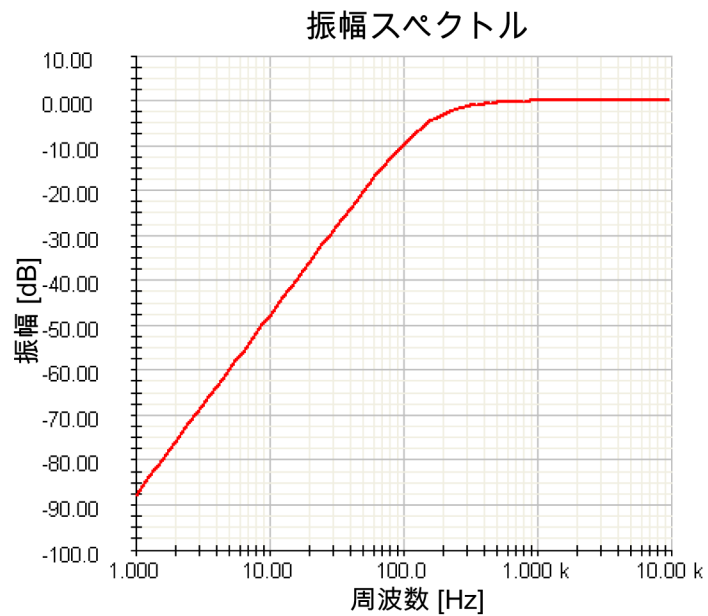


イラスト 4.12: 振幅スペクトル - FilterBesselHP

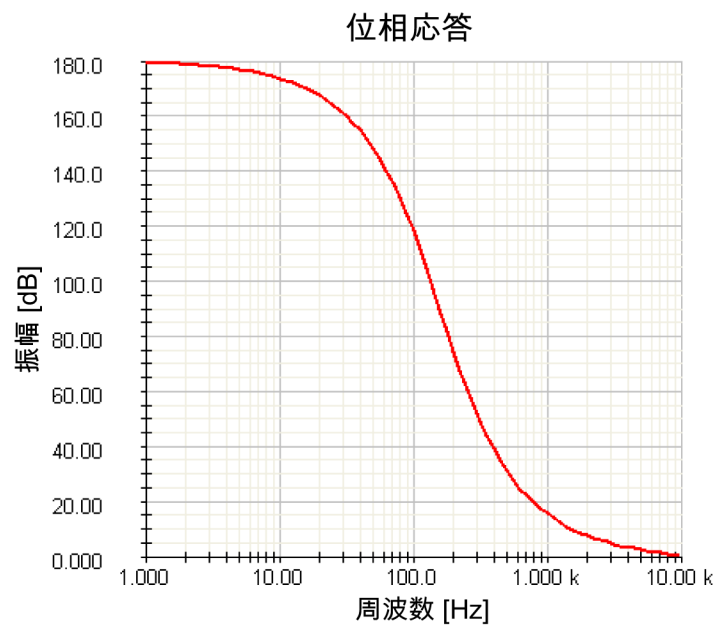


イラスト 4.13: 位相応答 - FilterBesselHP

以下も参照してください。

"@FilterBesselBP" ページ 75、"@FilterBesselBS" ページ 77 および
"@FilterBesselLP" ページ 71

4.23 @FilterBesselBP

関数

入力信号を直接型 IIR バンドパスベッセルフィルタでフィルタリングします。

構文

@FilterBesselBP (信号; 次数; 低域 F_c ; 高域 F_c)

@FilterBesselBP (信号; 次数; 低域 F_c ; 高域 F_c ; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
低域 F_c	番号：ヘルツによる低域カットオフ周波数
高域 F_c	番号：ヘルツによる高域カットオフ周波数
フェーズレス	番号：フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR バンドパスのベッセルフィルタリングを行います。

カットオフ周波数は、応答の振幅が-3 dB の周波数です。

詳細については、ページ 188 の IIR フィルタの章を参照してください。

例

```
Signal = @FilterBesselBP (Formula.Signal; 2; 200; 1000)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

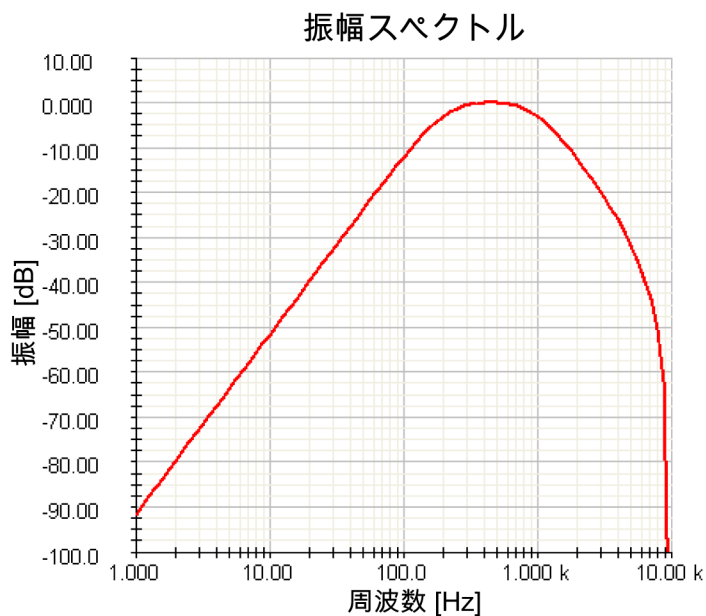


イラスト 4.14: 振幅スペクトル - FilterBesselBP

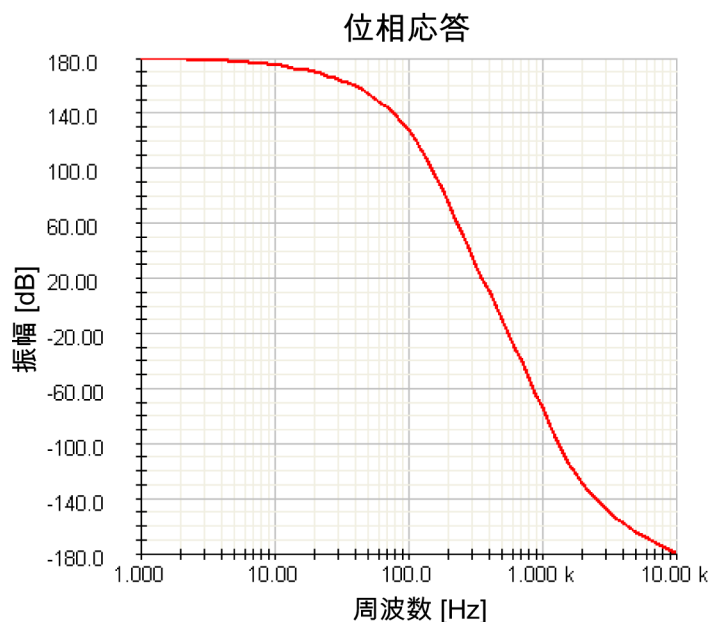


イラスト 4.15: 位相応答 - FilterBesselBP

以下も参照してください。

"@FilterBesselBS" ページ 77、"@FilterBesselHP" ページ 73 および
"@FilterBesselLP" ページ 71

4.24 @FilterBesselBS

関数

入力信号を直接型 IIR バンドストップベッセルフィルタでフィルタリングします。

構文

@FilterBesselBS(信号; 次数; 低域 F_c ; 高域 F_c)

@FilterBesselBS(信号; 次数; 低域 F_c ; 高域 F_c ; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
低域 F_c	番号：ヘルツによる低域カットオフ周波数
高域 F_c	番号：ヘルツによる高域カットオフ周波数
フェーズレス	番号：フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR バンドストップのベッセルフィルタリングを行います。

カットオフ周波数は、応答の振幅が-3 dB の周波数です。

詳細については、ページ 188 の IIR フィルタの章を参照してください。

例

```
Signal = @FilterBesselBS(Formula.Signal; 2; 200; 1000)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

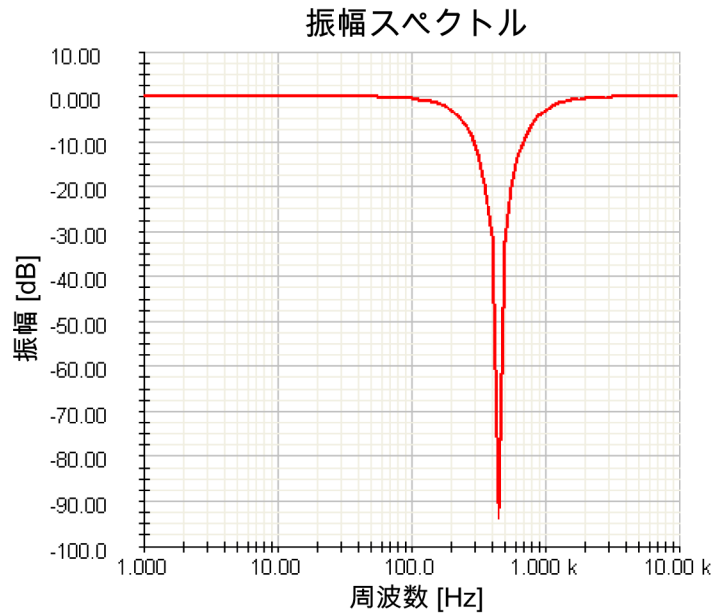


イラスト 4.16: 振幅スペクトル - FilterBesselBS

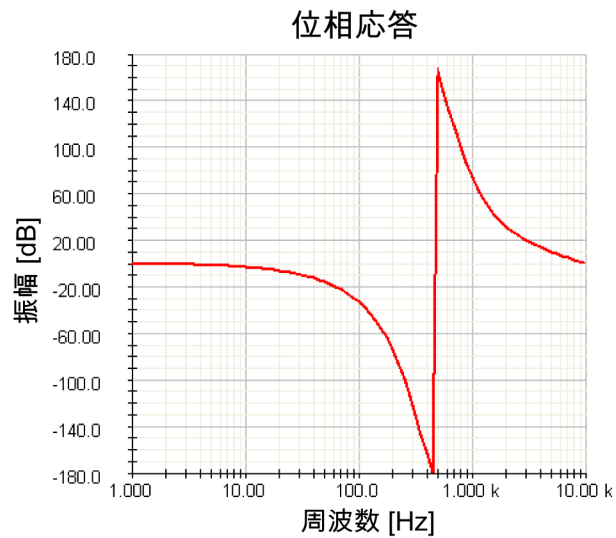


イラスト 4.17: 位相応答 - FilterBesselBS

以下も参照してください。

"@FilterBesselBP" ページ 75、"@FilterBesselHP" ページ 73 および
"@FilterBesselLP" ページ 71

4.25 @FilterChebyshevLP

関数

入力信号を直接型 IIR 低パスチエビシェフフィルタでフィルタリングします。

構文

@FilterChebyshevLP(信号; 次数; F_c)

@FilterChebyshevLP(信号; 次数; F_c ; リプル)

@FilterChebyshevLP(信号; 次数; F_c ; リプル; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
F_c	番号：ヘルツによるカットオフ周波数
リプル	番号：デシベルによる阻止帯域リプルの振幅で、デフォルト値は 1 dB です。
フェーズレス	番号：フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR 低パスのチエビシェフフィルタリングを行います。

カットオフ周波数は、バターワースおよびベッセルフィルタにおいて定義される -3 dB ポイントではなく、応答の振幅が最大振幅未満で指定されたリプルに等しくなる最大周波数です。例えば、リプルが 2 dB と定義され F_c が 200 Hz の場合、200 Hz での振幅は -2 dB となります。イラスト 4.18 を参照ください。

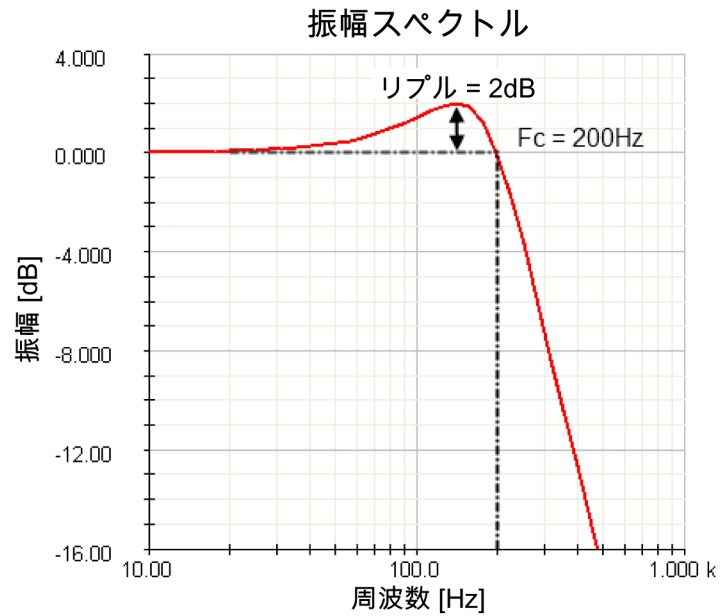


イラスト 4.18: 振幅スペクトル - FilterChebyshevLP (リップル)

詳細については、ページ 188 の IIR フィルタの章を参照してください。

例

```
Signal = @FilterChebyshevLP
        (Formula.Signal; 2; 200; 2)
```


公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

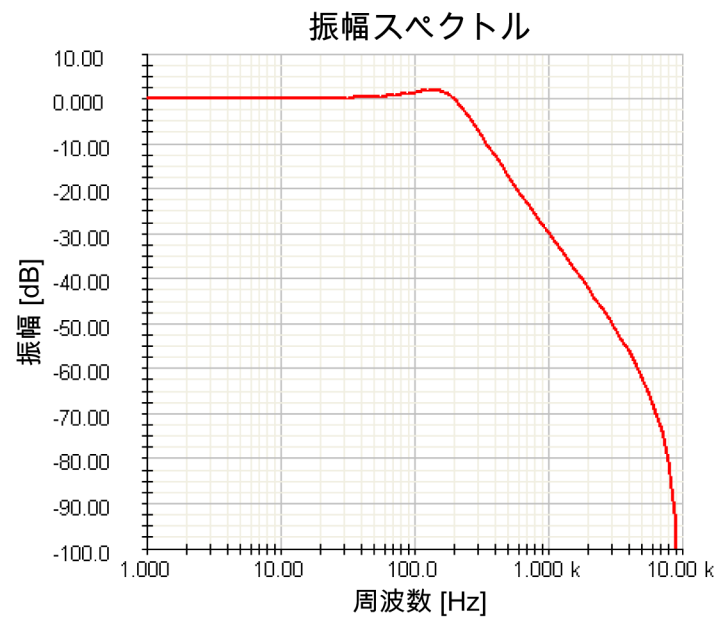


イラスト 4.19: 振幅スペクトル - FilterChebyshevLP

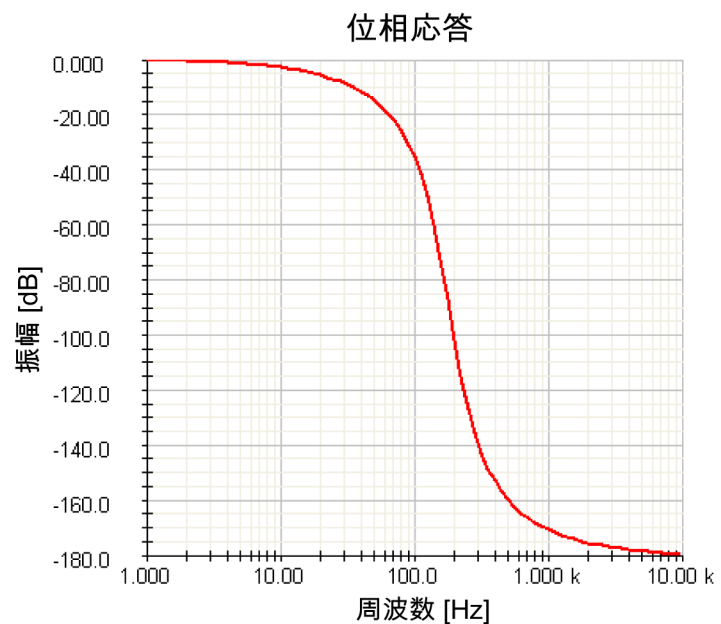


イラスト 4.20: 位相応答 - FilterChebyshevLP

以下も参照してください。

"@FilterChebyshevBP" ページ 84、"@FilterChebyshevBS" ページ 86 および
"@FilterChebyshevHP" ページ 82

4.26 @FilterChebyshevHP

関数

入力信号を直接型 IIR 高パスチエビシエフフィルタでフィルタリングします。

構文

@FilterChebyshevHP (信号; 次数; Fc)

@FilterChebyshevHP (信号; 次数; Fc; リプル)

@FilterChebyshevHP (信号; 次数; Fc; リプル; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
Fc	番号：ヘルツによるカットオフ周波数
リプル	番号：デシベルによる阻止帯域リプルの振幅で、デフォルト値は 1 dB です。
フェーズレス	番号：フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR 高パスのチエビシエフフィルタリングを行います。

カットオフ周波数は、バターワースおよびベッセルフィルタにおいて定義される -3 dB ポイントではなく、応答の振幅が最大振幅未満で指定されたリプルに等しくなる最小周波数です。例えば、リプルが 2 dB と定義され Fc が 200 Hz の場合、200 Hz での振幅は -2 dB となります。詳細については、ページ 188 の **IIR フィルタ** の章を参照してください。

例

```
Signal = @FilterChebyshevHP(Formula.Signal; 2; 200)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

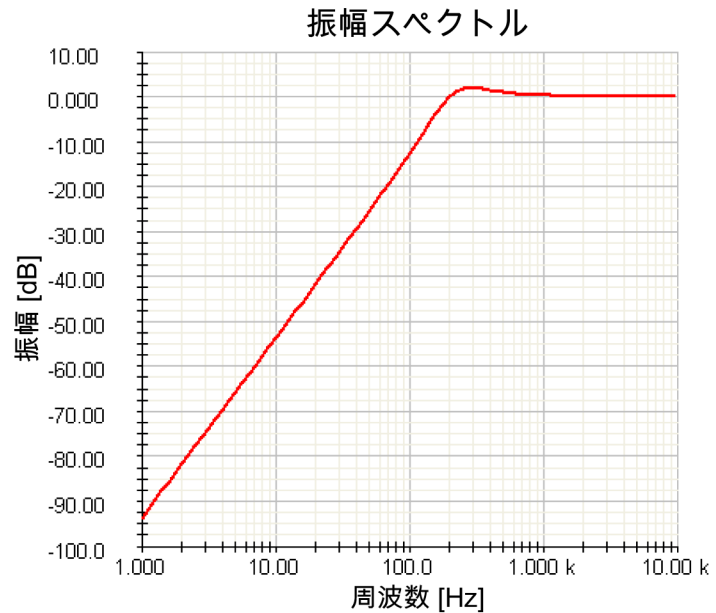


イラスト 4.21: 振幅スペクトル - FilterChebyshev

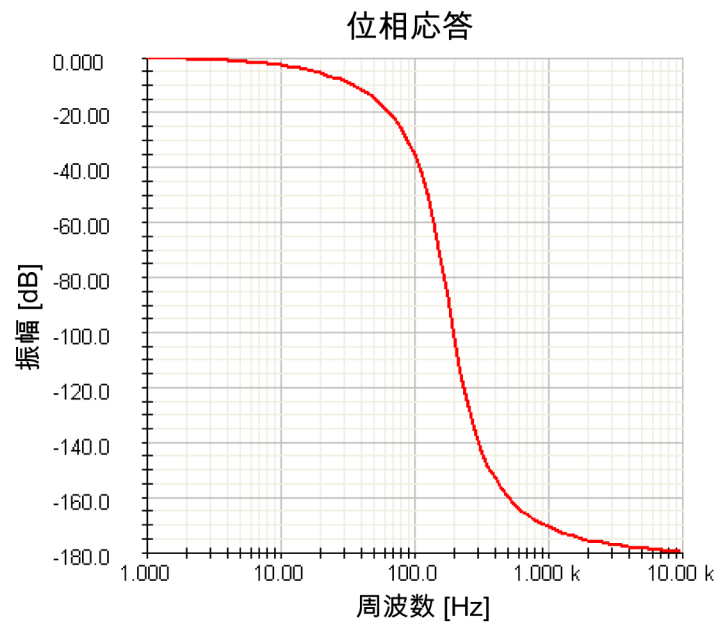


イラスト 4.22: 振幅スペクトル - FilterChebyshev

以下も参照してください。

"@FilterChebyshevBP" ページ 84、"@FilterChebyshevBS" ページ 86 および
"@FilterChebyshevLP" ページ 79

4.27 @FilterChebyshevBP

関数

入力信号を直接型 IIR バンドパスチェビシェフフィルタでフィルタリングします。

構文

@FilterChebyshevBP (信号; 次数; 低域 F_c ; 高域 F_c)

@FilterChebyshevBP (信号; 次数; 低域 F_c ; 高域 F_c ; リプル)

@FilterChebyshevBP (信号; 次数; 低域 F_c ; 高域 F_c ; リプル; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
低域 F_c	番号：ヘルツによる低域カットオフ周波数
高域 F_c	番号：ヘルツによる高域カットオフ周波数
リプル	番号：デシベルによる阻止帯域リプルの振幅で、デフォルト値は 1 dB です。
フェーズレス	番号：フィルタ方法 0 フィルタはフェーズレスで機能しません (デフォルト) 1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR バンドパスのチェビシェフフィルタリングを行います。カットオフ周波数は、バターワースおよびベッセルフィルタにおいて定義される -3 dB ポイントではなく、応答の振幅が最大振幅未満で指定されたリプルに等しくなる最小または最大周波数です。例えば、リプルが 2 dB と定義され低域 F_c が 200 Hz の場合、200 Hz での振幅は -2 dB となります。

詳細については、ページ 188 の IIR フィルタの章を参照してください。

例

```
Signal = @FilterChebyshevBP (Formula.Signal; 2; 200; 1000; 2)
```

公式. 信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のように
なります。

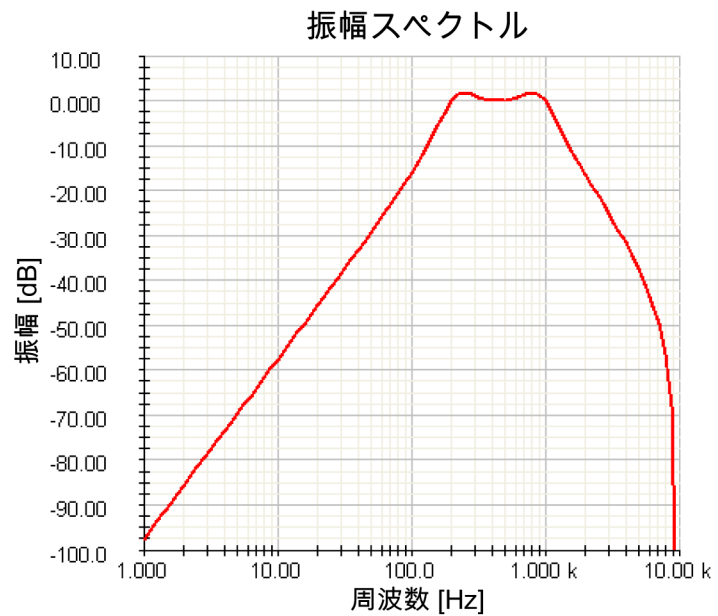


イラスト 4.23: 振幅スペクトル - FilterChebyshevBP

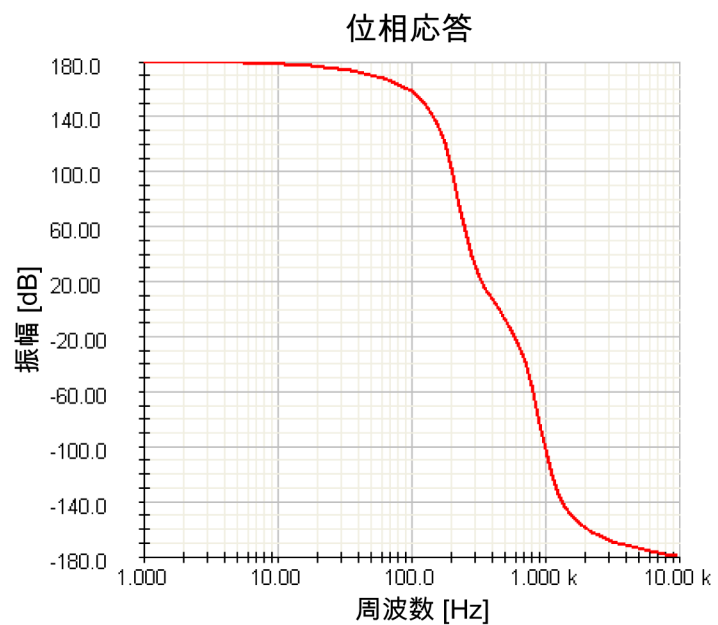


イラスト 4.24: 位相応答 - FilterChebyshevBP

以下も参照してください。

"@FilterChebyshevBS" ページ 86、"@FilterChebyshevHP" ページ 82 および
"@FilterChebyshevLP" ページ 79

4.28 @FilterChebyshevBS

関数

入力信号を直接型 IIR バンドストップチエビシェフフィルタでフィルタリングします。

構文

@FilterChebyshevBS(信号; 次数; 低域 F_c ; 高域 F_c)

@FilterChebyshevBS(信号; 次数; 低域 F_c ; 高域 F_c ; リプル)

@FilterChebyshevBS(信号; 次数; 低域 F_c ; 高域 F_c ; リプル; フェーズレス)

パラメータ

信号	入力波形
順序	番号：フィルタ次数
低域 F_c	番号：ヘルツによる低域カットオフ周波数
高域 F_c	番号：ヘルツによる高域カットオフ周波数
リプル	番号：デシベルによる阻止帯域リプルの振幅で、デフォルト値は 1 dB です。
フェーズレス	番号：フィルタ方法
	0 フィルタはフェーズレスで機能しません (デフォルト)
	1 フィルタはフェーズレスで機能します

出力

出力はフィルタリングされた波形になります。

記述

この関数は直接型 IIR バンドストップのチエビシェフフィルタリングを行います。カットオフ周波数は、バターワースおよびベッセルフィルタにおいて定義される -3 dB ポイントではなく、応答の振幅が最大振幅未満で指定されたリプルに等しくなる最小または最大周波数です。例えば、リプルが 2 dB と定義され低域 F_c が 200 Hz の場合、200 Hz での振幅は -2 dB となります。

例えば、リプルが 2 dB と定義され F_c が 200 Hz の場合、200 Hz での振幅は -2 dB となります。

詳細については、ページ 188 の **IIR フィルタ** の章を参照してください。

例

```
Signal = @FilterChebyshevBS(Formula.Signal; 2; 200; 1000)
```

公式.信号を 20 kHz でサンプリングした場合、周波数と位相応答は以下のようになります。

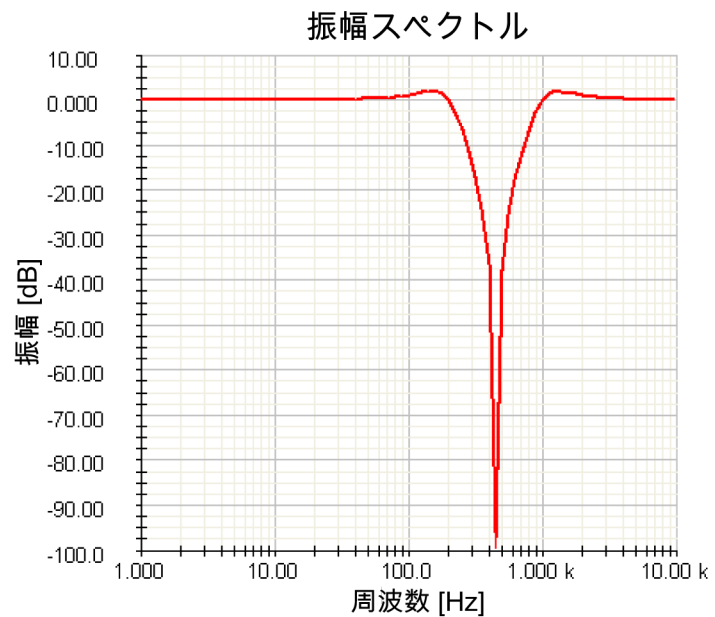


イラスト 4.25: 振幅スペクトル - FilterChebyshevBS

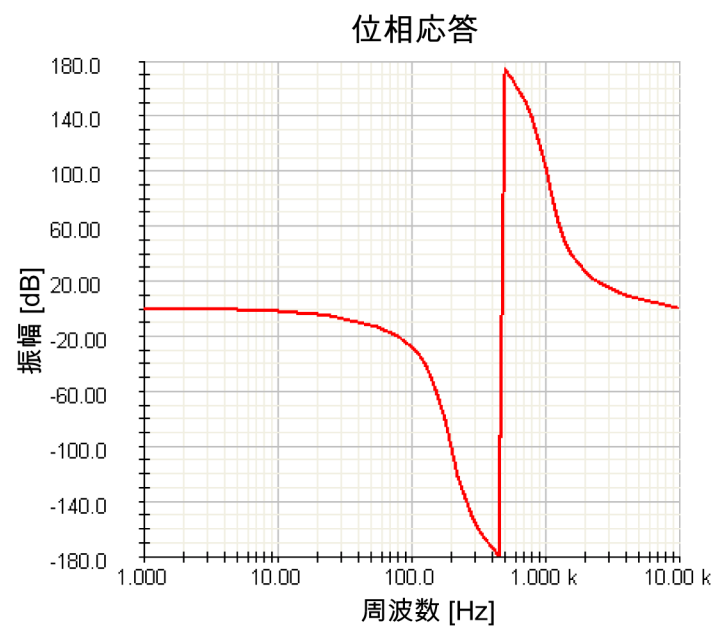


イラスト 4.26: 位相応答 - FilterChebyshevBS

以下も参照してください。

"@FilterChebyshevBP" ページ 84、"@FilterChebyshevHP" ページ 82 および
"@FilterChebyshevLP" ページ 79

4.29 @Frequency

関数

波形の周波数を決定します。

構文

@Frequency (波形)

@Frequency (波形; 開始)

@Frequency (波形; 開始; 終了)

パラメータ

波形	周波数が決定される入力波形
開始	番号：セグメントの開始。
終了	番号：セグメントの終了。

出力

出力は数値です。開始と終了のパラメータが設定されているとき、この関数は「記録が行われる間」にも機能します。結果は、用意されているデータから継続的に計算されます。

記述

波形または波形セグメントの周波数が決められます。セグメントの限界（開始と終了）を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート 開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として（時間など）指定されなければなりません。

周波数は以下のように決定されます。

- 最初に、波形の最小値と最大値が決められます。これらの値の平均値が0（ゼロ）レベルであると見なされ、開始から終了までの範囲におけるゼロ交差の数が計算されます。交差は、最初のものと同じ方向にあるべきです。交差は、ノイズによる影響を抑えるために0（ゼロ）レベルの周囲 $\pm 5\%$ のヒステリシスを使って決定されます。
- 同じ方向の最初と最後のゼロ交差の間の時間差、およびそれらの間に存在するゼロ交差の数に応じて、波形の期間が決定されます。周波数は期間の逆数です。

例

以下の例は、50 Hz 信号の周波数を決定します。

```
Signal = @SineWave(20k; 1000; 50)
```

```
Freq   = @Frequency(Formula.Signal)
```

以下も参照してください。

"@Period" ページ 132

4.30 @GreaterEqualThan

関数

この関数は、2つの数値入力パラメータについて `greater-than-or-equal-to` (以上) (\geq) の評価を実施します。

構文

`@GreaterEqualThan (以上) (パラメータ 1; パラメータ 2)`

パラメータ

パラメータ 1 番号: 評価のために使用される最初のパラメータ

パラメータ 2 番号: 評価のために使用される 2 つ目のパラメータ

出力

出力は 1 または 0 です。

記述

`GreaterEqualThan` (以上) の関数は、入力パラメータについて「`greater-than-or-equal-to` (以上)」の評価を実施します。

パラメータ 1 \geq パラメータ 2 であるとき戻り値は 1 (= 真) となり、その他の場合は 0 (= 偽) となります。

一般的に `GreaterEqualThan` (以上) の関数は、IIF 関数と組み合わせて使用されます。

例

以下の例では入力パラメータの比較を行い、それに応じて結果を出します。

```
GETExam1    = @GreaterEqualThan(5; 5) => 0 (false)
GETExam2    = @GreaterEqualThan(12; 10) => 1 (true)
IIFExample  = @IIF(Formula.GETExam2; "TRUE"; "FALSE")
```

以下も参照してください。

"@EqualTo" ページ 57、"@GreaterThan" ページ 92、"@IIF" ページ 95、"@LessEqualThan" ページ 106 および"@LessThan" ページ 107

4.31 @GreaterThan

関数

この関数は、2つの数値入力パラメータについて `greater-than (超える) (>)` の評価を実施します。

構文

`@GreaterThan (パラメータ 1; パラメータ 2)`

パラメータ

パラメータ 1 番号：評価のために使用される最初のパラメータ

パラメータ 2 番号：評価のために使用される 2 つ目のパラメータ

出力

出力は 1 または 0 です。

記述

`GreaterThan (超える)` の関数は、入力パラメータについて「`greater-than (超える)`」の評価を実施します。

パラメータ 1 > パラメータ 2 であるとき戻り値は 1 (真) となり、その他の場合は 0 (偽) となります。

一般的に `GreaterThan (超える)` の関数は、IIF 関数と組み合わせて使用されます。

例

以下の例では入力パラメータの比較を行い、それに応じて結果を出します。

```
GTEexam1      = @GreaterThan ( 5;      => 0 ( false )
                    100 )
GTEexam2      = @GreaterThan ( 12;   => 1 ( true )
                    10 )
IIFExample    = @IIF ( Formula.GTEexam2; "TRUE"; "FALSE" )
```

以下も参照してください。

"@EqualTo" ページ 57、"@GreaterEqualThan" ページ 91、"@IIF" ページ 95、"@LessEqualThan" ページ 106 および"@LessThan" ページ 107

4.32 @Histogram

関数

振幅ヒストグラムを計算します。

構文

@Histogram (波形; Y 下位; Y 上位; 数)

パラメータ

波形	ヒストグラムの計算が行われる入力波形
Y 下位	番号 : ヒストグラムに含める最低振幅
Y 上位	番号 : ヒストグラムに含める最高振幅
番号	領域の数

出力

振幅ヒストグラムを含む波形

記述

ヒストグラムの関数は、各振幅領域において発生する振幅値 (波形値) の数を決定します。振幅領域は、Y 下位および Y 上位の間の範囲を数の領域に分割することで指定されます。各領域の幅は (Y 上位 - Y 下位) / 数となります。波形の各値は、これらの領域のいずれかに分類されます。ヒストグラムは、各領域に分類された値の個数を示します。領域の最大許容数は 4096 です。通常、100 から 1000 までの間の値が使用されます。

パラメータ Y 下位と Y 上位が、ヒストグラムに含まれる振幅の範囲を決定します。この範囲に該当しない値はすべて破棄されます。範囲に該当しない値がヒストグラムの領域に含まれるようにするため、ヒストグラムを計算する前に @Clip 関数を使用して振幅値を限定してください。

例

以下の例では、11 の領域を使用して -5.5 V と +5.5 V の間の振幅のヒストグラムを計算します。この場合、各領域の幅は 1 V となります。

```
Signal = 5 * @SineWave(5000; 1000; 10)
Histo = @Histogram(Formula.Signal; -5.5; 5.5; 11)
```

以下の例では、公式データベース変数の信号を想定することで、最小と最大の波形の間に 100 の領域を持つヒストグラムを計算します。

```
MinSignal = @Min(Formula.Signal)
MaxSignal = @Max(Formula.Signal)
```

```
Histo      = @Histogram(Formula.Signal;  
                        Formula.MinSignal;  
                        Formula.MaxSignal; 100)
```

以下も参照してください。

"@Clip" ページ 45、"@Max" ページ 111 および"@Min" ページ 118

4.33 @IIF

関数

この関数は Immediate If (IIF)(緊急的なもしも) を実行し、ある条件が TRUE (真) であると評価されるある値を戻し、FALSE (偽) であると評価されると別の値を戻します。

構文

@IIF (パラメータ; もしも真であるとき; もしも偽であるとき)

パラメータ

パラメータ 評価される波形/数/文字列

もしも真であるパラメータの評価が真である場合の波形/数/文字列の出力
とき

もしも偽であるパラメータの評価が偽である場合の波形/数/文字列の出力
とき

出力

出力は波形、数、または文字列となることができます。

記述

IIF 関数が最初の入力パラメータについて評価を行う場合、評価の種類はこのパラメータの種類によって異なり、詳細については以下の表の中で説明します。評価が行われ、結果は真または偽となります。結果が真である場合、2つ目のパラメータが IIF 関数の出力として戻され、それ以外の場合は3つ目のパラメータが戻されます。

以下の表は、最初のパラメータの評価について説明しています。

パラメータの種類	真	偽
数値	0 以外	0 に等しい
文字列	空ではない	なし
波形	0 以外	0

例

以下の例では、生成された正弦波を使用します。信号が表示という名称のディスプレイに表示され、有効なカーソルの位置における波形の値が 0.5 を超えると IIF 関数は文字列「0.5 を超える」を戻し、これ以外の場合、文字列「0.5 未満」が戻されます。

GreaterThan (超える) 関数を使って、有効なカーソルの位置における波形の値と 0.5 の値の比較が行われます。値が 0.5 を超えると結果は 1 (真) となり、それ以外の場合、結果は 0 (偽) になります。この結果は、IIF 関数の最初のパラメータとなります。

```
Signal      = @SineWave(8000; 8001; 5)
GreaterHalf = @GreaterThan
              (Display.Display.ActiveCursor.YValue;
               0.5)
IIFExample  = @IIF(Formula.GreaterHalf; "Above 0.5";
                  "Below 0.5")
```

以下も参照してください。

"@EqualTo" ページ 57、"@GreaterEqualThan" ページ 91、"@GreaterThan" ページ 92、"@LessEqualThan" ページ 106 および"@LessThan" ページ 107

4.34 @Integrate

関数

波形を積分します。

構文

@Integrate (波形)

パラメータ

波形 積分する波形

出力

積分された波形

記述

積分は、すべてのサンプル値の累積和を計算します。

$$i(1) = y(1)$$

$$i(n) = i(n-1) + y(n) \cdot \Delta x \text{ for } n = 2, \dots, N$$

積分は、波形の開始地点から終了地点までに至る曲線の下領域を計算します。

例

以下の例では、瞬時のパワーを積分して時間の関数としてのエネルギーを得ます。

```
Signal = @SineWave(10k; 1000; 50)
Power  = (Formula.Signal * Formula.Signal) / 600
Energy = @Integrate(Formula.Power)
```

以下も参照してください。

"@Diff" ページ 53 および "@Energy" ページ 55

4.35 @IntLookUp

関数

波形からのデータを変換ルックアップテーブル (CLUT) へのインデックス (ポインタ) として使用します。 CLUT は個別のファイルに保存されます。

構文

@IntLookUp (波形; CLUT)

@IntLookUp (波形; CLUT; インデックスオフセット)

パラメータ

波形 入力波形

CLUT 文字列 : CLUT ファイルの名前と固有の場所を示すフルパス
インデックスオ番号 : インデックスポインタへのオフセット
フセット

出力

変換された波形

記述

この関数は波形からのデータを変換ルックアップテーブルへのインデックスとして使用します。 データは 16 ビットの整数であることができます。

変換ルックアップテーブル (CLUT) は ASCII (テキスト) ファイルに保存されます。

このファイルの各行には、浮動小数点の数が一つ含まれます。 行番号は CLUT のインデックスとして使用されます。

最初の行はデフォルトでインデックス 0 に関連しています。 しかし、関数がオプションの「インデックスオフセット」パラメータを使用する場合、最初の行はインデックスオフセットと関連しています。 これにより、マイナス値を使った作業が可能になります。

16 ビットの整数によるデータを使用する場合、CLUT は $2^{16} = 65536$ ポイントを含んでいる必要があります。 データに記号 (+ または -) が付いているときは、32768 のオフセットを使用します。 これで、CLUT の最初の地点は - 32768 と関連します。

例

以下のルックアップテーブルを想定します。

行番号	値
1	11.3
2	21.4
3	31.5
4	41.4
5	51.2
6	61.3
7	71.5
8	81.6
9	91.2
10	101.4
11	111.2

元々の波形が以下の通りであると想定します。

6, 2, 7, 8

最初の値 6 は行 7 に関連しています (インデックスは 0 ベースです)。CLUT テーブルの対応する値は 71.5 です。

変換された出力波形は以下のようになります。

71.5, 31.5, 81.6, 91.2

オプションの「インデックスオフセット」パラメータが使用され、2 に設定された場合 :

最初の値 6 は行 5 に関連しています (インデックスは 6 - 2 は 0 ベースです)。CLUT テーブルの対応する値は 51.2 です。

変換された出力波形は以下のようになります。

51.2, 11.3, 61.3, 71.5

入力波形に、CLUT に保存された最大インデックスよりも大きいインデックスが含まれる場合、CLUT の最後の値が使用されます。入力波形に、CLUT に保存された最小インデックスよりも小さいインデックスが含まれる場合、CLUT の最初の値が使用されます。

ディレクトリ「C:\CalibData」に存在する外部のルックアップテーブル「Lookup.asc」を使用する場合 :

```
Signal = @Ramp(1000; 2001; 0; 4096; 11)
Calib   = @IntLookUp
        (Formula.Signal; "C:\CalibData\Lookup.asc")
```

以下も参照してください。
"@IntLookUp12" ページ 101

4.36 @IntLookUp12

関数

波形からのデータを変換ルックアップテーブル (CLUT) へのインデックス (ポインタ) として使用します。

構文

@IntLookUp12 (波形; Clut)

パラメータ

波形	変換する波形
Clut	変換ルックアップテーブル

出力

変換された波形

記述

この関数は波形からのデータを変換ルックアップテーブルへのインデックス (ポインタ) として使用します。

ノート この関数は 12 ビットの整数データのために最適化されています。

変換ルックアップテーブルには 4096 個の値が含まれています。

元々の波形の各値は CLUT へのポインタとして使用されます。CLUT での対応する値は、出力を生成するために使用されます。

CLUT を作成することができます。

- 高位のプログラミング言語と Perception CSI オプションを使用します。
- 公式データベース機能 (の組み合わせ) を共に使用します。

次の図は、この関数の使用例を示しています。

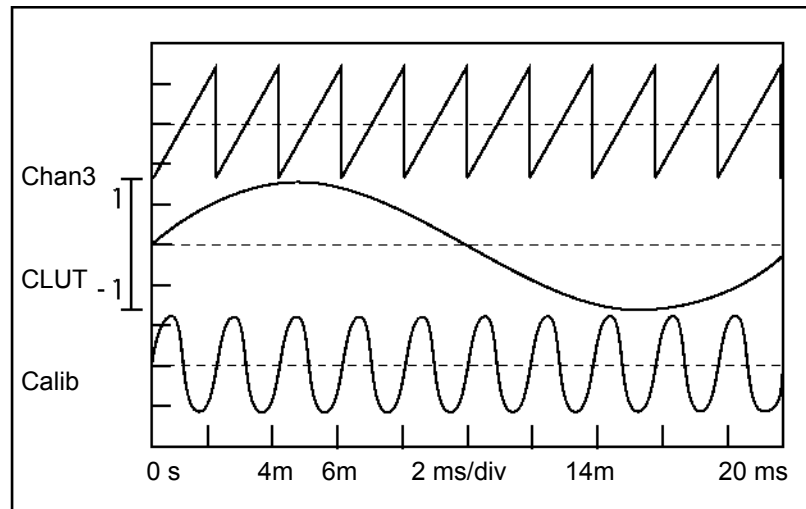


イラスト 4.27: 図 - @IntLookUp12 関数

- 頂部のトレースは元々の記録を示します。
- 中間部のトレースは CLUT の内容を示します。
- 底部のトレースは変換されたデータを示します。

この図を作成するために、以下の公式が使用されます。

```
CLUT = @SineWave(200k; 4096; 48.8)
Org   = @Ramp(2M; 40960; 4095; 10)
Calib = @IntLookUp12(Formula.Org; Formula.CLUT)
```

最初に、周波数 48.8 Hz の 4096 の地点で正弦波が生成されます。200 kHz のサンプルレートで、一つのサイクルが波形に組み込まれます。

生成された波形 Formula.Org を取り入れ、この波形を CLUT にインデックスとして使用することで、「Calib」という結果が生成されます。

ノート 元々のデータは、解像度が 12 ビットである -32767 ~ +32768 である整数データ (2 バイト) であるべきです。データが使用される前に、整数値は 16 で除算されて正しいインデックスが取得されます。したがって、14 ビットまたは 16 ビットのデータも使用することができますが、これらを使用すると最も重要でないビットが失われ、予想できない、あるいは予期しない結果が生成される可能性があります。

以下も参照してください。

"@IntLookUp" ページ 98

4.37 @Join

関数

複数の波形を連結することで結合します。

構文

@Join (波形 1; ...; 波形 N)

パラメータ

波形 1	最初の波形
波形 N	N >= 2 での最後の波形

出力

連結された波形

記述

この関数は複数の波形を連結して、新しい波形を作ります。波形の終端を次の波形の開始点に連結することで、波形の結合が行われます。水平方向の軸は最初の波形によって決まります。出力波形の最初のサンプルの x 座標は、最初の波形の最初のサンプルの x 座標に等しくなります。その他の波形からのサンプルは、単に最初の波形のサンプルに連結されます。出力される波形の長さは、パラメータとして指定される波形の長さの和になります。

入力波形が様々な時間ベースを持つ場合、出力波形にも複数の時間ベースが含まれます。

例

以下の例に示すように、この関数はシミュレートした信号を生成するために特に有用です。

```

Seg1    = @Ramp(1k; 100; 0; 0)
Seg2    = @Ramp(1k; 50; 0; 1)
Seg3    = @Ramp(1k; 100; 1; 1)
Signal  = @Join(Formula.Seg1; Formula.Seg2;
                Formula.Seg3)

```

Join (結合) 関数は、波形の最初の有効なデータポイントを前の波形に連結します。このため、連結された 2 つの波形の元々の終了時間と開始時間が異なる場合があります。しかし、出力波形ではサンプルインターバルの部分が一つしか存在しないこととなります。

以下の例では、このことについて説明しています。

```
Seg1 = @Ramp(1k; 100; 0; 0)  
Seg2 = @Ramp(1k; 100; 0; 0)
```

@XShift を使ってセグメント 2 を水平方向に 100 単位シフトします。

```
Seg3 = @XShift(Formula.Seg2; 100)  
Signal = @Join(Formula.Seg1; Formula.Seg3)
```

以下も参照してください。
"@Cut" ページ 49

4.38 @Length

関数

一つの波形に含まれる**サンプルの数**を返します。

構文

@Length (*波形*)

パラメータ

波形 入力波形

出力

出力は数値です。

記述

この関数は、指定された入力波形の中で使用することが可能なサンプルの数を返します。この値は、他の関数のための入力パラメータとして使用することが可能です。

例

以下の例では、1000 個のサンプルで構成される正弦波を生成します。変数 N サンプルは、この信号の長さ (1000) を受信します。

```
Signal = 25 * @SineWave(20k; 1000; 50)
NSamples = @Length(Formula.Signal)
```

以下も参照してください。

"@XDelta" ページ 175、"@XFirst" ページ 178 および"@XLast" ページ 179

4.39 @LessEqualThan

関数

この関数は、2つの数値入力パラメータについて `less-than-or-equal-to` (以下) (\leq) の評価を実施します。

構文

`@LessEqualThan` (以下) (パラメータ 1; パラメータ 2)

パラメータ

パラメータ 1 番号: 評価のために使用される最初のパラメータ

パラメータ 2 番号: 評価のために使用される 2 つ目のパラメータ

出力

出力は 1 または 0 です。

記述

`LessEqualThan` (以下) の関数は、入力パラメータについて「`less-than-or-equal-to` (以下)」の評価を実施します。

パラメータ 1 \leq パラメータ 2 であるとき戻り値は 1 (真) となり、その他の場合は 0 (偽) となります。

一般的に `LessEqualThan` (以下) の関数は、IIF 関数と組み合わせて使用されません。

例

以下の例では入力パラメータの比較を行い、それに応じて結果を出します。

```

LETEXam1    = @LessEqualThan(5; 5)    => 1 (true)
LETEXam2    = @LessEqualThan(12; 10) => 0 (false)
IIFExample  = @IIF(Formula.LETEXam2; "true"; "false")

```

以下も参照してください。

"@EqualTo" ページ 57、"@GreaterEqualThan" ページ 91、"@GreaterThan" ページ 92、"@IIF" ページ 95 および"@LessThan" ページ 107

4.40 @LessThan

関数

この関数は、2つの数値入力パラメータについて **less-than (未満)** ($<$) の評価を実施します。

構文

@LessThan (パラメータ 1; パラメータ 2)

パラメータ

パラメータ 1 番号: 評価のために使用される最初のパラメータ

パラメータ 2 番号: 評価のために使用される 2 つ目のパラメータ

出力

出力は 1 または 0 です。

記述

LessThan (未満) の関数は、入力パラメータについて「less-than (未満)」の評価を実施します。パラメータ 1 < パラメータ 2 であるとき戻り値は 1 (真) となり、その他の場合は 0 (偽) となります。

一般的に LessThan (未満) の関数は、IIF 関数と組み合わせて使用されます。

例

以下の例では入力パラメータの比較を行い、それに応じて結果を出します。

```

LessThanExamp11 = @LessThan(5; => 1 (true)
                    100)
LessThanExamp12 = @LessThan(12; => 0 (false)
                    10)
IIFExample      = @IIF(Formula.LessThanExamp12;
                    "TRUE"; "FALSE")

```

以下も参照してください。

"@EqualTo" ページ 57、"@GreaterEqualThan" ページ 91、"@IIF" ページ 95 および "@LessEqualThan" ページ 106

4.41 @Ln

関数

特定の数値の自然対数、(または底「e」への対数)、 $\ln(x)$ または $\log_e(x)$ として表記：数値を得るために底 (e) を累乗するために必要な累乗指数。

構文

@Ln (Par)

パラメータ

Par 入力波形または数値

出力

入力の自然対数を含む波形または数値

記述

対数の底 e が計算されます ($e = 2.718\dots$)。この対数は自然対数と呼ばれます。波形パラメータが使用されると、各サンプルについて対数関数が個別に計算されます。

数値が 1E-30 よりも小さい場合、値は $\ln(1E-30) = -69.08$ に設定されます。

形式上、 $\ln(a)$ は 1 から a までの $1/x$ のグラフ (積分) の下の領域として定義することができます。すなわち：

$$\ln(a) = \int_1^a \frac{1}{x} dx.$$

この関数は、@Exp の逆関数です。

例

以下の例は、システム変数 System.Constants.e の自然対数が 1 に等しいことを示しています。

```
One = @Ln(System.Constants.e)
```

以下の例は、自然対数を波形において使用する方法を示しています。

```
Wave        = @Ramp(10k; 1k; 1E-5; 10)
WaveLn      = @Ln(Formula.Wave)
```

以下も参照してください。
"@Exp" ページ 58 および "@Log" ページ 110

4.42 @Log

関数

底 10 への特定数値の対数、 $\log_{10}(x)$ として表記：数値を得るために底 (10) を累乗するために必要な累乗指数。

構文

@Log (Par)

パラメータ

Par 入力波形または数値

出力

入力の対数を含む波形または数値

記述

対数の底 10 が計算されます。波形パラメータが使用されると、各サンプルについて対数関数が個別に計算されます。

数値が 1E-30 よりも小さい場合、値は $\log(1E-30) = -30$ に設定されます。

例

$10^x = 100$ である場合、 $x = \log_{10}(100)$ となります。

```
x = @Log(100)
```

結果は $x = 2$ となります。

以下も参照してください。

"@Ln" ページ 108 および "@Pow" ページ 134

4.43 @Max

関数

波形の最大値を決定します。

構文

@Max (波形)

@Max (波形; 開始)

@Max (波形; 開始; 終了)

パラメータ

波形	最大値が判断される入力波形
開始	番号 : セグメントの開始。
終了	番号 : セグメントの終了。

出力

出力は数値です。開始と終了のパラメータが設定されているとき、この関数は「記録が行われる間」にも機能します。結果は、用意されているデータから継続的に計算されます。

記述

波形または波形セグメントの最大値が決められます。セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート 開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。

例

以下の例では、50 Hz の正弦波を生成します。信号の最初の 20 ms (50 Hz のある期間) の最大値が計算され、(ピーク) 振幅が決定されます。

```
Signal = 25 * @SineWave(20k; 1000; 50)
Ampl   = @Max(Formula.Signal; 0; 20m)
```

最大値の位置を決めるには、代わりに @MaxPos 関数を使います。そして見つかった位置における最大値は、@Value 関数を使って得ることができます。

```
TMax = @MaxPos(Formula.Signal; 0; 20m)
```

`Ampl = @Value(Formula.Signal; Formula.TMax)`

以下も参照してください。

"@MaxPos" ページ 114、"@Min" ページ 118 および"@MinPos" ページ 121

"@Value" ページ 174

4.44 @MaxNum

関数

複数の数値のうちの最大値を判断します。

構文

@MaxNum (パラメータ 1; ...; パラメータ N)

パラメータ

パラメータ 1 最初の数値

パラメータ N 最後の数値、N >= 2

出力

すべての数値のうちの最大値

記述

この関数は、指定されたすべてのパラメータのうちの最大値を戻します。各パラメータは、有効な値を含む数値でなければなりません。いずれかのパラメータに不正な種類が含まれている場合、あるいは値が含まれていない場合、関数は値を戻しません。

例

以下の例では、一つの波形の最小値を計算しますが、0 (ゼロ) 未満であった場合は値をクリップします。波形は実際の信号であると想定されます。

```
Signal      = Active.Group1.Recorder_A.Ch_A1
MinValue    = @Min(Formula.Signal)
ClippedMin  = @MaxNum(Formula.MinValue; 0)
```

以下の例では、10 を戻します。

```
Ten = @MaxNum(-10; 3; 2.4; 10.0; -9.9)
```

以下も参照してください。

"@MinNum" ページ 120

4.45 @MaxPos

関数

波形の**最大値の位置**を判断します。

構文

@MaxPos (波形)

@MaxPos (波形; 開始)

@MaxPos (波形; 開始; 終了)

パラメータ

波形	最大値の位置が決定される入力波形
開始	番号：セグメントの開始。
終了	番号：セグメントの終了。

出力

出力は最大値の x 位置です。開始と終了のパラメータが設定されているとき、この関数は「記録が行われる間」にも機能します。結果は、用意されているデータから継続的に計算されます。

記述

波形または波形セグメントの最大値の x 位置が決められます。セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート *開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。*

ノート *戻される値は、サンプル数ではなく x 位置に対する波形の x 軸における位置です。この数値の単位は波形の x 単位です。*

例

以下の例では、圧力が最大値に到達するときの時間を判断します。実際の最大圧力は、この時間のために @Value 関数を使用することで判断されます。

圧力は実際の信号であると想定されます。

```
Pressure = Active.Group1.Recorder_A.Ch_A1
```

```
TimeOfMax = @MaxPos (Formula.Pressure)
MaxPress  = @Value (Formula.Pressure;
                  Formula.TimeOfMax)
```

以下も参照してください。

"@Max" ページ 111、"@Min" ページ 118 および"@MinPos" ページ 121

4.46 @Mean

関数

波形の算術的平均値を計算します。

構文

@Mean (波形)

@Mean (波形; 開始)

@Mean (波形; 開始; 終了)

パラメータ

波形	平均値の計算が行われる入力波形
開始	番号：セグメントの開始。
終了	番号：セグメントの終了。

出力

出力は数値です。開始と終了のパラメータが設定されているとき、この関数は「記録が行われる間」にも機能します。結果は、用意されているデータから継続的に計算されます。

記述

平均値は、以下の公式を使って計算されます。

$$\text{Mean} = \frac{1}{N} \sum_{n=n_1}^{n_2} y(n) \text{ with } N = (n_2 - n_1 + 1)$$

セグメントの限界である開始 (n1) と終了 (n2) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート 開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。

例

以下の例では、ハムノイズが加えられた信号 (定数値 100) を生成します。信号の最初の 20 ms (50 Hz のある期間) の平均値が計算されます。

```
Signal = 100 + 0.5 * @SineWave(20k; 1000; 50) +  
         @Noise(20k; 1000)  
Avg     = @Mean(Formula.Signal; 0; 20m)
```

アブレイジングが行われているため、平均値は波形に含まれる個々のサンプルと比較して精度がかなり高い信号の読取値です。

以下も参照してください。

"@RMS" ページ 156 および "@StdDev" ページ 165

4.47 @Min

関数

波形の最小値を決定します。

構文

@Min (波形)

@Min (波形; 開始)

@Min (波形; 開始; 終了)

パラメータ

波形	最小値が判断される入力波形
開始	番号：セグメントの開始。
終了	番号：セグメントの終了。

出力

出力は数値です。開始と終了のパラメータが設定されているとき、この関数は「記録が行われる間」にも機能します。結果は、用意されているデータから継続的に計算されます。

記述

波形または波形セグメントの最小値が決められます。セグメントの限界（開始と終了）を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート *開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として（時間など）指定されなければなりません。*

例

以下の例では、DC 値の上に 50 Hz の正弦波を生成します。信号の最初の 20 ms (50 Hz のある期間) の最大値と最小値が計算され、50 Hz コンポーネントの (ピーク間) 振幅が決定されます。

```
Signal = 100 + 25 * @SineWave(20k; 1000; 50)
MinAmpl = @Min(Formula.Signal; 0; 20m)
MaxAmpl = @Max(Formula.Signal; 0; 20m)
Vpp      = Formula.MaxAmpl - Formula.MinAmpl
```

最小値の位置を決めるには、代わりに @MinPos 関数を使います。そして見つかった位置における最小値は、@Value 関数を使って得ることができます。

```
TMin = @MinPos(Signal; 0; 20m)
Minim = @Value(Formula.Signal; Formula.TMin)
```

以下も参照してください。

"@Max" ページ 111、"@MaxPos" ページ 114 および

"@MinPos" ページ 121

4.48 @MinNum

関数

複数の数値のうちの最小値を計算します。

構文

@MinNum (パラメータ 1; ...; パラメータ N)

パラメータ

パラメータ 1 最初の数値

パラメータ N 最後の数値 N >= 2

出力

すべての数値のうちの最小値

記述

この関数は、指定されたすべてのパラメータのうちの最小値を戻します。各パラメータは、有効な値を含む数値でなければなりません。いずれかのパラメータに不正な種類が含まれている場合、あるいは値が含まれていない場合、関数は値を戻しません。

例

以下の例では、一つの波形の最大値を計算しますが、100 を超える場合は値をクリップします。波形は実際の信号であると想定されます。

```
Signal      = Active.Group1.Recorder_A.Ch_A1
MaxValue    = @Max(Formula.Signal)
ClippedMax  = @MinNum(Formula.MaxValue; 100)
```

以下の例では、-10 を戻します。

```
MinusTen = @MinNum(45.0; 10; -3.3; 20; -10.0; 5)
```

以下も参照してください。

"@MaxNum" ページ 113

4.49 @MinPos

関数

波形の**最小値の位置**を判断します。

構文

@MinPos (波形)

@MinPos (波形; 開始)

@MinPos (波形; 開始; 終了)

パラメータ

波形	最小値の位置が決定される入力波形
開始	番号 : セグメントの開始。
終了	番号 : セグメントの終了。

出力

出力は最小値の x 位置です。

記述

波形または波形セグメントの最小値の x 位置が決められます。セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート **開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。**

ノート **戻される値は、サンプル数ではなく x 位置に対する波形の x 軸における位置です。この数値の単位は波形の x 単位です。**

例

以下の例では、圧力が最小値に到達するときの時間を判断します。実際の最小圧力は、この時間のために @Value 関数を使用することで判断されます。変数圧力は実際の信号であると想定されます。

```
Pressure = Active.Group1.Recorder_A.Ch_A1
TimeOfMin = @MinPos (Formula.Pressure)
MinPress = @Value (Formula.Pressure; Formula.TimeOfMin)
```

以下も参照してください。

"@Max" ページ 111、"@MaxPos" ページ 114 および"@Min" ページ 118

4.50 @NextHillPos

関数

この関数は、波形の次のローカル最大値の位置を検索します。

構文

@NextHillPos (波形; 開始位置; ヒステリシス)

パラメータ

波形 次のローカル最大値の位置が決定される入力波形
開始位置 番号： (前方への) 検索が開始するべき x 単位の開始位置です。
ヒステリシス 番号： ノイズ抑制ヒステリシス値

出力

出力は次のローカル最大値の x 位置です。

記述

波形における次のローカル最大値の x 位置が決定されます。

検索は、開始位置により指定される x 位置から右に向かって開始します。開始位置が波形の開始地点よりも前にある場合、波形の開始地点は検索のための開始位置として使用されます。

ヒステリシスの値を使って、信号へのノイズの影響が抑制されます。例えば、信号に 100 mV のピーク間ノイズが存在する場合、200 mV をヒステリシス値として指定することで、アルゴリズムが小さいノイズピークをローカル最大値として判断することを防止することができます。

ノート 戻される値は、サンプル数ではなく x 位置に対する波形の x 軸における位置です。この数値の単位は波形の x 単位です。

例

以下の例では、40 mV のヒステリシスを使用して 500 ms の前方への検索を開始することで、10 mV のノイズがある 1 V の正弦波でローカル最大値の位置を決めます。

```
Signal = @SineWave(10k; 10000; 50 ) + 0.01 * @Noise(10k; 10000)
NHPos  = @NextHillPos (Formula.Signal; 500m; 40m)
```

以下も参照してください。

"@NextValleyPos" ページ 127、"@PrevHillPos" ページ 135 および
"@PrevValleyPos" ページ 139

4.51 @NextLvlCross

関数

指定された信号レベルで波形の次の交差の位置を決定します：

構文

@NextLvlCross (波形; 開始位置; レベル)

@NextLvlCross (波形; 開始位置; レベル; 傾き)

パラメータ

波形	次の平面交差の位置が決定される入力波形
開始位置	番号： 検索が開始するべき x 単位の開始位置です。
レベル	番号： 検索する対象である振幅レベル
傾き	番号： 傾きの方向 (-1, 0, 1)

出力

出力は次の平面交差の x 位置です。

記述

指定されたレベルと方向で、波形における次の平面交差の x 位置が決定されます。検索は、開始位置により指定される x 位置から右側の正の時間方向に向かって開始します。開始位置が波形の開始地点よりも前にある場合、波形の開始地点は検索のための開始位置として使用されます。

傾きのパラメータは、検索対象の平面交差のタイプを管理します。

- 傾き = 1： 正の平面交差 (下のレベルから上のレベルへ)
- 傾き = -1： 負の平面交差 (上のレベルから下のレベルへ)
- 傾き = 0： あらゆる交差 (正または負)

ノート

戻される値は、サンプル数ではなく x 位置に対する波形の x 軸における位置です。この数値の単位は波形の x 単位です。

例

以下の例では、“Active.Group1.Recorder_A.Ch_A1”における実際の信号として使用可能であると想定される TTL 信号に存在する最初の TTL パルスの開始と終了の時間を決定します。

```
TTSignal = Active.Group1.Recorder_A.Ch_A1
TimeStart = @NextLvlCross (Formula.TTSignal; -1E20;
                          2.5; 1)
```

```
TimeEnd = @NextLvlCross (Formula.TTSignal;  
                        Formula.TimeStart; 2.5; -1)
```

以下も参照してください。
"@PrevLvlCross" ページ 137

4.52 @NextValleyPos

関数

この関数は、波形の次のローカル最小値の位置を検索します。

構文

@NextValleyPos (波形; 開始位置; ヒステリシス)

パラメータ

波形 次のローカル最小値の位置が決定される入力波形
開始位置 番号： (前方への) 検索が開始するべき x 単位の開始位置です。
ヒステリシス 番号： ノイズ抑制ヒステリシス値

出力

出力は次のローカル最小値の x 位置です。

記述

波形における次のローカル最小値の x 位置が決定されます。

検索は、開始位置により指定される x 位置から右に向かって開始します。開始位置が波形の開始地点よりも前にある場合、波形の開始地点は検索のための開始位置として使用されます。

ヒステリシスの値を使って、信号へのノイズの影響が抑制されます。例えば、信号に 100 mV のピーク間ノイズが存在する場合、200 mV をヒステリシス値として指定することで、アルゴリズムが小さいマイナスのノイズピークをローカル最小値として判断することを防止することができます。

ノート 戻される値は、サンプル数ではなく x 位置に対する波形の x 軸における位置です。この数値の単位は波形の x 単位です。

例

以下の例では、40 mV のヒステリシスを使用して 500 ms の前方への検索を開始することで、10 mV のノイズがある 1 V の正弦波でローカル最小値の位置を決めます。

```
Signal = @SineWave(10k; 10000; 50) + 0.01 * @Noise(10k; 10000)
NVPos  = @NextValleyPos(Formula.Signal; 500m; 40m)
```

以下も参照してください。

"@NextHillPos" ページ 123、"@PrevHillPos" ページ 135 および"@PrevValleyPos"
ページ 139

4.53 @Noise

関数

ノイズを含む波形を生成します。

構文

@Noise (*F* サンプリング; *N* サンプル)

パラメータ

F サンプリング番号 : サンプリング周波数

N サンプル サンプルの数

出力

ノイズを含む波形

記述

この関数は、振幅が - 1 から 1 までの間のランダムノイズを含む波形を生成します。サンプリング周波数、および生成するサンプルの数を指定しなければなりません。

ノイズを生成できることは、測定された、または生成された波形の分析においてノイズの影響を調査するために非常に有用である可能性があります。

例

以下の例では、10 V のピーク間のノイズに隠された 5 V の振幅の正弦波を生成します。

```
Signal = 5 * @SineWave(10k; 1000; 50) + 10 *
          @Noise(10k; 1000)
```

以下も参照してください。

"@Pulse" ページ 141、"@Ramp" ページ 145、"@SineWave" ページ 160 および "@SquareWave" ページ 164

4.54 @Not

関数

この関数は、入力パラメータについて論理 NOT の評価を実施します。

構文

@Not (パラメータ 1)

パラメータ

パラメータ 1 NOT 評価に使用される数

出力

出力は 1 または 0 です。

記述

@Not 関数は、入力パラメータについて論理 NOT の評価を実施します。評価の内容に応じて、結果 1 または 0 になります。0 に等しくない数値は論理「True (真)」に対応し、数値 0 は論理「False (偽)」に対応します。

NOT 関数の真の表は以下の通りです。

パラメータ 1	結果
真	偽
偽	真

例

以下に例とその戻り値の一覧を示します。

```

NotExamp11 = @Not(1)      => 0 (= false)
NotExamp12 = @Not(4)      => 0 (= false)
NotExamp13 = @Not(0)      => 1 (= true)

```

以下も参照してください。

"@And" ページ 39 および "@Or" ページ 131

4.55 @Or

関数

この関数は、入力パラメータについて論理 OR の評価を実施します。

構文

@Or (パラメータ 1; ...; パラメータ N)

パラメータ

パラメータ 1 番号 : OR 評価のために使用される最初のパラメータ

パラメータ N 番号 : OR 評価のために使用される最後のパラメータ

N >= 2

出力

出力は 1 または 0 です。

記述

@Or 関数は、入力パラメータについて論理 OR の評価を実施します。評価の内容に応じて、結果 1 または 0 になります。0 に等しくない数値は論理「True (真)」に対応し、数値 0 は論理「False (偽)」に対応します。

OR 関数の真の表は以下の通りです。

パラメータ 1	パラメータ 2	結果
真	真	真
真	偽	真
偽	真	真
偽	偽	偽

例

以下に例とその戻り値の一覧を示します。

OrExamp11 = @Or(1; 1) => 1 (= true)

OrExamp12 = @Or(1;4;10) => 1 (= true)

OrExamp13 = @Or(1;4;0;6) => 1 (= true)

OrExamp14 = @Or(0;0) => 0 (= false)

以下も参照してください。

"@And" ページ 39 および "@Not" ページ 130

4.56 @Period

関数

波形の期間を決定します。期間は周波数の逆数です。

構文

@Period (波形)

@Period (波形; 開始)

@Period (波形; 開始; 終了)

パラメータ

波形	期間が判断される入力波形
開始	番号：セグメントの開始。
終了	番号：セグメントの終了。

出力

出力は数値です。

記述

波形または波形セグメントの期間が決められます。セグメントの限界（開始と終了）を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート *開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として（時間など）指定されなければなりません。*

期間は、底と最上部の大きさのアルゴリズムを用いて決められます。これらの大きさの平均は 0（ゼロ）レベルであると見なされます。開始から終了までの範囲に存在するゼロ交差の数が計算されます。交差は、最初のものと同じ方向にあるべきです。交差は、ノイズによる影響を抑えるために 0（ゼロ）レベルの周囲のヒステリシスを使って決定されます。ヒステリシスは、最上部と底の間の +/- 5%の差となります。

同じ方向の最初と最後のゼロ交差の間の時間差、およびそれらの間に存在するゼロ交差の数に応じて、波形の期間が決定されます。

例

以下の例は、50 Hz 信号の期間を決定します。

```
Signal = @SineWave(10k; 10k; 50) + 0.01 * @Noise(10k;  
10k)  
Period = @Period(Formula.Signal)  
Freq    = 1 / Formula.Period
```

以下も参照してください。
"@Frequency" ページ 89

4.57 @Pow

関数

累乗法、数学演算、書き込まれた a^n 、以下の 2 つパラメータを含む：底「a」および指数「n」

構文

@Pow (底; 指数)

パラメータ

底	底の波形または数値
指数	指数の波形または数値

出力

底および指数の累乗法を含む波形または数値

記述

指数関数は、累乗指数で累乗された底を計算します。

底と指数が数値であると、結果も数値になります。

底または指数が波形であると、数値と組み合わせて各サンプルについて指数関数が個別に計算されます。結果は波形になります。

ノート *底と指数の両方に波形を使用することは不可能です。*

例

以下の結果は等しくなります。

```

Input      = 2 * @SineWave(10k; 10k; 10)
Result1    = @Exp(Formula.Input)
Result2    = @Pow(System.Constants.e; Formula.Input)

```

以下も参照してください。

"@Exp" ページ 58

4.58 @PrevHillPos

関数

この関数は、波形の前のローカル最大値の位置を検索します。

構文

@PrevHillPos (波形; 開始位置; ヒステリシス)

パラメータ

波形 前のローカル最大値の位置が決定される入力波形
開始位置 番号 : (後方への) 検索が開始されるべき x 単位の開始位置です。
ヒステリシス 番号 : ノイズ抑制ヒステリシス値

出力

出力は前のローカル最大値の x 位置です。

記述

波形における前のローカル最大値の x 位置が決定されます。

検索は、開始位置により指定される x 位置から左に向って開始します。開始位置が波形の終了地点よりも後にある場合、波形の終了地点は検索のための開始位置として使用されます。

ヒステリシスの値を使って、信号へのノイズの影響が抑制されます。例えば、信号に 100 mV のピーク間ノイズが存在する場合、200 mV をヒステリシス値として指定することで、アルゴリズムが小さいノイズピークをローカル最大値として判断することを防止することができます。

ノート 戻される値は、サンプル数ではなく x 位置に対する波形の x 軸における位置です。この数値の単位は波形の x 単位です。

例

以下の例では、40 mV のヒステリシスを使用して 500 ms の後方への検索を開始することで、10 mV のノイズがある 1 V の正弦波でローカル最大値の位置を決めます。

```
Signal = @SineWave(10k; 10000; 50) + 0.01 * @Noise(10k; 10000)
PHPos = @PrevHillPos(Formula.Signal; 500m; 40m)
```

以下も参照してください。

"@NextHillPos" ページ 123、"@NextValleyPos" ページ 127 および
"@PrevValleyPos" ページ 139

4.59 @PrevLvlCross

関数

指定された信号レベルで波形の前の交差の位置を決定します：

構文

@PrevLvlCross (波形; 開始位置; レベル)

@PrevLvlCross (波形; 開始位置; レベル; 傾き)

パラメータ

波形	前の平面交差の位置が決定される入力波形
開始位置	番号： (後方への) 検索が開始されるべき x 単位の開始位置です。
レベル	番号： 検索する対象である振幅レベル
傾き	番号： 傾きの方向 (-1, 0, 1)

出力

出力は前の平面交差の x 位置です。

記述

指定されたレベルと方向で、波形における前の平面交差の x 位置が決定されます。検索は、開始位置により指定される x 位置から左側の負の時間方向に向かって開始します。

傾きのパラメータは、検索対象の平面交差のタイプを管理します。

- 傾き = 1： 正の平面交差 (下のレベルから上のレベルへ)
- 傾き = -1： 負の平面交差 (上のレベルから下のレベルへ)
- 傾き = 0： あらゆる交差 (正または負)

ノート

戻される値は、サンプル数ではなく x 位置に対する波形の x 軸における位置です。この数値の単位は波形の x 単位です。

例

以下の例では、TTL 信号に存在する最後の TTL パルスの開始と終了の時間を決定します。この信号は実際の信号であると想定されます。

```
TTLSignal = Active.Group1.Recorder_A.Ch_A1
TimeEnd   = @PrevLvlCrossing( Formula.TTLSignal; 1E20;
                             2.5; 1)
```

```
TimeStart = @PrevLvlCrossing(Formula.TTLSignal;  
                             Formula.TimeEnd; 2.5; -1)
```

以下も参照してください。
"@NextLvlCross" ページ 125

4.60 @PrevValleyPos

関数

この関数は、波形の前のローカル最小値の位置を検索します。

構文

@PrevValleyPos (波形; 開始位置; ヒステリシス)

パラメータ

波形 前のローカル最小値の位置が決定される入力波形
開始位置 番号： (後方への) 検索が開始されるべき x 単位の開始位置です。
ヒステリシス 番号： ノイズ抑制ヒステリシス値

出力

出力は前のローカル最小値の x 位置です。

記述

波形における前のローカル最小値の x 位置が決定されます。

検索は、開始位置により指定される x 位置から左に向って開始します。開始位置が波形の終了地点よりも後にある場合、波形の終了地点は検索のための開始位置として使用されます。

ヒステリシスの値を使って、信号へのノイズの影響が抑制されます。例えば、信号に 100 mV のピーク間ノイズが存在する場合、200 mV をヒステリシス値として指定することで、アルゴリズムが小さいマイナスのノイズピークをローカル最小値として判断することを防止することができます。

ノート 戻される値は、サンプル数ではなく x 位置に対する波形の x 軸における位置です。この数値の単位は波形の x 単位です。

例

以下の例では、40 mV のヒステリシスを使用して 500 ms の後方への検索を開始することで、10 mV のノイズがある 1 V の正弦波でローカル最小値の位置を決めます。

```
Signal = @SineWave(10k; 10000; 50) + 0.01 * @Noise(10k; 10000)
PVPos = @PrevValleyPos(Formula.Signal; 500m; 40m)
```

以下も参照してください。

"@NextHillPos" ページ 123、"@NextValleyPos" ページ 127 および"@PrevHillPos"
ページ 135

4.61 @Pulse

関数

単一パルスを含む波形を生成します。

構文

@Pulse(*F* サンプリング; *N* サンプル; パルス開始)

@Pulse(*F* サンプリング; *N* サンプル; パルス開始; パルス幅)

パラメータ

F サンプリング番号 : サンプリング周波数

N サンプル サンプルの数

パルス開始 番号 : パルスが開始されるサンプル

パルス幅 番号 : デフォルト値が 1 に設定されたサンプルにおけるパルス幅。

出力

単一パルスを含む波形。

記述

この関数は単一パルスの波形を生成します。単一パルスを生成するサンプリング周波数、サンプルの数を指定することができます。波形の長さには制限はありません。パルス波の振幅は 1 V です。

パルス関数を使って、フィルタ関数のインパルスおよびステップ応答を決定することができます。

この関数を使用して、様々な波形を合成することができます。このデータを、他の分析関数のための入力として使用することができます。

例

```
SignalPulse      = @Pulse(80k; 80k; 20k)
ImpulseResponse = @FilterButterworthLP
                  (Formula.SignalPulse; 2; 200)

SignalStep       = @Pulse(80k; 80k; 20k; 10k)
StepResponse     = @FilterButterworthLP
                  (Formula.SignalStep; 2; 200)
```

信号の一部のみにのみ関心がある場合、パルス関数を使用して、指定された間隔の範囲外にあるすべての値をゼロにすることができます。この間隔は、パルス関数のパルス開始およびパルス幅パラメータで定義することができます。

```
SignalInterval = Formula.SignalStep *  
                Active.Group1.Recorder_A.Ch_A1
```

以下も参照してください。

"@Noise" ページ 129、"@SineWave" ページ 160、"@SquareWave" ページ 164
および"@Ramp" ページ 145

4.62 @PulseWidth

関数

波形におけるパルスのパルス幅を決定します。

構文

@PulseWidth (波形)

@PulseWidth (波形; 開始)

@PulseWidth (波形; 開始; 終了)

パラメータ

波形	パルスを含む入力波形
開始	番号 : セグメントの開始。
終了	番号 : セグメントの終了。

出力

出力は数値です。

開始と終了のパラメータが設定されているとき、この関数は「記録が行われる間」に機能します。結果は、開始と終了の間のデータが使用可能である場合に計算されます。

記述

パルス幅は、波形 (波形セグメント) におけるパルスの 1 番目と 2 番目の 50% の大きさの移行の間での時間差を使って計算されます。

大きさのパーセンテージ交差は、底と最上部の大きさを用いて計算されます。底と最上部の大きさは、波形 (波形セグメント) の大きさに関するヒストグラムに示される 2 つの母集団を検索し、これらの 2 つの母集団の平均値を得ることによって判断されます。

セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。

開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート 開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。

例

以下の例ではパルスを生成し、50% ~ 50%のパルス幅を計算します。結果は150 ms となります。

```

Sig1      = @Ramp(1k; 100; 0; 0)
Sig2      = @Ramp(1k; 51; 0; 10)
Sig3      = @Ramp(1k; 99; 10; 10)
Sig4      = Sig4 x
Signal    = @Join(Formula.Sig1; Formula.Sig2;
                  Formula.Sig3; Formula.Sig4; Formula.Sig1)
Width     = @PulseWidth(Formula.Signal)

```

以下も参照してください。

"@FallTime" ページ 61 および "@RiseTime" ページ 154

4.63 @Ramp

関数

線形ランプを一つ以上含む波形を生成します。

構文

@Ramp (*F* サンプリング; *N* サンプル; *Y* 開始; *Y* 終了)

@Ramp(*F* サンプリング; *N* サンプル; *Y* 開始; *Y* 終了; カウント)

パラメータ

F サンプリング番号 : サンプリング周波数

N サンプル サンプルの数

Y 開始 番号 : 最初のサンプルの値

Y 終了 番号 : 最後のサンプルの値

カウント ランプの数

出力

線形ランプを一つ以上含む波形

記述

この関数は、*Y* 開始から開始し、*Y* 終了で終了する線形ランプを一つ以上含む波形を生成します。サンプリング周波数、および生成するサンプルの数も指定することができます。波形の長さは 1 ギガサンプルに制限されます。

オプションのパラメータカウントを指定する場合、指定された数のランプが生成されます。カウント値を使用すると、のこぎり波形が生成されます。カウントが指定されていないときは、値 0 (ゼロ) が使用されます。この場合、のこぎり波形ではない単一のランプが生成されます。

@Joint 機能およびその他の波形生成機能と共にランプ機能を生成できることを利用して、様々な波形を合成することができます。シミュレーションしたデータを、他の分析関数のための入力として使用することができます。

例

以下の例では、4 V のピーク間のノイズが追加された 10 V の振幅の擬似パルスを生成します。

```
Sig1 = @Ramp(1k; 100; 0; 0)
Sig2 = @Ramp(1k; 51; 0; 10)
Sig3 = @Ramp(1k; 100; 10; 10)
Sig4 = @Ramp(1k; 51; 10; 0)
```

```
Signal = @Join(Formula.Sig1; Formula.Sig2;  
              Formula.Sig3; Formula.Sig4) + 2 * @Noise(1k;  
              302)
```

以下も参照してください。

"@Noise" ページ 129、"@Pulse" ページ 141、"@SineWave" ページ 160 および
"@SquareWave" ページ 164

4.64 @ReadAsciiFile

関数

ASCII (テキスト) ファイルから波形データをインポートします。

構文

@ReadAsciiFile (ファイル名)

パラメータ

ファイル名 文字列：ASCII ファイルの名前と固有の場所を示すフルパス

出力

出力は波形になります。

記述

この関数は、ASCII ファイルから波形データをインポートします。インポートされたデータのサンプルは、等距離であると解釈されます。

ASCII ファイルにはヘッダとデータ部分が含まれていなければなりません。短いバージョンと長いバージョンという 2 種類のヘッダがサポートされています。

短いヘッダ：

線	記述	例
1	ヘッダ行数	5
2	データの区切り文字 (点、カンマ、タブ、またはセミコラム)	
3	データペアの数	n
4	x と y (x; y) のスケール係数	5.0E-8;2.44E+0
5	x と y (x; y) の単位	s;A

長いヘッダ：

線	記述	例
1	ヘッダ行数	12
2	データの区切り文字 (点、カンマ、タブ、またはセミコラム)	
3	データペアの数	n
4	データ生成の日付	17.03.00
5	データ生成の時間	23:59

線	記述	例
6	データの作成者に関する特別な情報。 この行は空白にすることができます。 互換性上の理由のみ。	TDG 0.5
7	コメント (最大 80 文字)	最初の例: テスト 1;
8	x と y (x; y) のスケール係数	5.0E-8;2.44E+0
9	x と y (x; y) の単位	s;A
10	スケールされた単位 (x; y) の名前	時間; 電流
11	単位ビットによる y データの分解能	12
12	単位%によるダイナミックレンジの使用	80

データ :

データはヘッダの後に存在し、ヘッダの大きさにより 6 行目または 13 行目から開始します。

各データの行には、x と y のペアが含まれています。x 値と y 値の間のセパレータは、ヘッダの 2 行目に定義されます。

インポートされたデータのサンプルは、等距離であると解釈されます。

最初のデータの行には、最初の (最も小さい) x 値が含まれています。

例

以下の例には、ファイル「ReferenceCurve.asc」が読み込まれる方法を示します。

```
Signal = @ReadAsciiFile("d:\data\ReferenceCurve.asc")
```

4.65 @Reduce

関数

再度サンプリングを行って波形に存在するサンプルの数を減らします。

構文

@Reduce (波形; 率)

パラメータ

波形	再サンプリングする波形
率	番号 : 再サンプリング率

出力

再サンプリングされた波形

記述

波形を再度サンプリングしてデータの量を減らします。再サンプリング率は最も近い整数 N に四捨五入し、 $2, 3, \dots$ (波形の長さ) / 2 の範囲で示されるべきです。出力の波形は、利用可能なサンプルがなくなるまで、サンプル 1、サンプル $1 + N$ 、サンプル $1 + 2N$ という具合にサンプルを受け取ります。出力される波形の長さは、入力される波形の長さの N 分の 1 になります。出力される波形のサンプリング間隔は、入力される波形のサンプリング間隔の N 倍になります。最初のサンプルの x 座標は変更されません。

波形に高い周波のコンポーネントが存在する場合、再サンプリングによってエイリアシングの効果が引き起こされる可能性があります。この場合、再サンプリングを行う前にデータの低パスフィルタリングを目的とした円滑化の @Smooth 関数を使用してください。

例

以下の例では、正弦波を 10 の係数で再サンプリングし、10 の係数によってサンプリング率を効果的に低下させます。サンプルの数は、1000 個から 100 個に減らされます。

```
Sine1000 = @SineWave(10k; 1000; 50)
Sine100   = @Reduce(Formula.Sine1000; 10)
```

以下も参照してください。

"@Res2" ページ 153 および "@Smooth" ページ 161

4.66 @RefCheck

関数

波形または波形セグメントを 1 つまたは 2 つのリファレンス波形と比較します。

構文

@RefCheck (波形; 上部範囲)

@RefCheck (波形; 上部範囲; 下部範囲)

@RefCheck (波形; 上部範囲; 下部範囲; 開始)

@RefCheck (波形; 上部範囲; 下部範囲; 開始; 終了)

パラメータ

波形	比較対象の入力波形
上部範囲	上部範囲を含む波形
下部範囲	下部範囲を含む波形
開始	番号 : 比較対象のセグメントの開始位置
終了	番号 : 比較対象のセグメントの終了位置

出力

(リファレンス波形間の) 成功または失敗を示す数

記述

この関数を使って、波形を 1 つまたは 2 つのリファレンス波形と比較します。出力は、上部または下部の波形を持つ入力波形の最初の平面交差の x 位置となります。

入力波形が上部と下部の波形の間にあると、戻り値は「数ではない」(Nan) 値になります。

セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。

開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

例

以下の例では、1つの正弦波と2つの「範囲」の正弦波を生成します。ノイズを使って成功または失敗を生成します。より高いノイズ値によって失敗が生じます。

```
Signal = @SineWave(8000; 8001; 5)
Upper  = Formula.Signal + 0.1
Lower  = Formula.Signal - 0.1
Noise  = 0.1 * @Noise(8000; 8001)
Signal2 = Formula.Signal + Formula.Noise
Check  = @RefCheck(Formula.Signal2; Formula.Upper;
                  Formula.Lower)
Result = @IIF(Formula.Check; "NOT OK"; "OK")
```

4.67 @RemoveGlitch

関数

指定された開始時間と終了時間の間の新しいデータと共に、これらの二つのポイントの間にある直線を利用して、波形を戻します。

構文

@RemoveGlitch (*波形*; *開始*; *終了*)

パラメータ

<i>波形</i>	グリッチが除去される入力波形
<i>開始</i>	番号：グリッチ開始時間
<i>終了</i>	番号：グリッチ終了時間

出力

出力は、元々の波形と同じ長さの波形になります。

記述

結果として生じる波形は、開始時間と終了時間の間に直線を生成します。元々の波形の残りの部分は変化しません。

ノート *記録の間に公式を使用してグリッチを除去する場合、開始と終了のパラメータは使用可能なデータの区域の中に存在していなければなりません。開始時間または終了時間が波形の中に対応するデータ値を持っていない場合、RemoveGlitch 関数の計算を行うことはできません。*

例

以下の例では、2秒間の長さを持つ50 Hzの正弦波を生成します。そしてこの関数は、1秒から1.5秒までの間隔においてサンプルが直線と置き換えられる長さが2秒間の新しい波形を生成します。

```
Signal = @SineWave(1k; 2000; 50)
Cropped = @RemoveGlitch(Formula.Signal; 1000m; 1500m)
```


4.68 @Res2

関数

長さが 2 の累乗数になるように波形を再サンプリングします。

構文

@Res2 (波形)

パラメータ

波形 再サンプリングする波形

出力

再サンプリングされた波形

記述

この関数は、新しいサンプルの数量が元々の長さよりも 2 の累乗分大きくなるか、元々の長さと同しくなるように波形を再サンプリングします。この関数を効率的に使用して、希望される波形セグメントのスペクトルを計算することができます。そうしたセグメントに含まれるサンプルの数は通常、FFT アルゴリズムによって求められる 2 の累乗と等しくありません。最初にこのセグメントにおいて @Res2 関数を使用することで、同じ信号を含む新しい波形が生成されますが、サンプルの数が 2 の累乗になるようにより速い率でサンプリングが行われます。波形を切り詰めることもゼロ詰めすることもなく、再サンプリングされたこの波形を FFT の計算のために使用することができます。

再サンプリング率は通常、整数ではないため、2 つのサンプルの間の波形の値が生成されなければなりません。これは線形補間を使って行うことができます。再サンプリング率のために、波形のサンプリング間隔が修正されます。

例

以下の例は 2500 個のサンプルの正弦波を生成し、4096 個のサンプルを持つ波形に対してそれを再サンプリングします。長さの値は 4096 になります。

```
Signal      = @SineWave(10k; 2500; 100)
Resampled   = @Res2( Formula.Signal )
Length      = @Length( Formula.Resampled )
```

以下も参照してください。

"@Reduce" ページ 149

4.69 @RiseTime

関数

波形におけるパルスの立ち上がり時間を決定します。

構文

@RiseTime (波形)

@RiseTime (波形; 開始)

@RiseTime (波形; 開始; 終了)

パラメータ

波形	パルスの前縁を含む入力波形
開始	番号：セグメントの開始。
終了	番号：セグメントの終了。

出力

出力は数値です。

開始と終了のパラメータが設定されているとき、この関数は「記録が行われる間」に機能します。結果は、開始と終了の間のデータが使用可能である場合に計算されます。

記述

立ち上がり時間は、波形 (波形セグメント) でのパルスの最初の前縁における近位地点 (10% の大きさの移行) と遠位地点 (90% の大きさの移行) との間での時間差を使って計算されます。

大きさのパーセンテージ交差は、底と最上部の大きさを用いて計算されます。底と最上部の大きさは、波形 (波形セグメント) の大きさに関するヒストグラムに示される 2 つの母集団を検索し、これらの 2 つの母集団の平均値を得ることによって判断されます。

セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート 開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。

例

以下の例ではパルスを生成し、前縁の 10% ~ 90% の立ち上がり時間を計算します。
結果は 40 ms となります。

```
Sig1 = @Ramp(1k; 100; 0; 0)
Sig2 = @Ramp(1k; 51; 0; 10)
Sig3 = @Ramp(1k; 100; 10; 10)
Sig4 = @Ramp(1k; 51; 10; 0)
Signal = @Join(Formula.Sig1; Formula.Sig2;
               Formula.Sig3; Formula.Sig4; Formula.Sig1)
Rise = @RiseTime(Formula.Signal)
```

以下も参照してください。

"@FallTime" ページ 61 および "@PulseWidth" ページ 143

4.70 @RMS

関数

波形の二乗平均平方根を計算します。

構文

@RMS (波形)

@RMS (波形; 開始)

@RMS (波形; 開始; 終了)

パラメータ

波形	RMS の計算が行われる入力波形
開始	番号 : セグメントの開始。
終了	番号 : セグメントの終了。

出力

出力は数値です。

記述

RMS は、以下の公式を使って計算されます。

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{n=n_1}^{n_2} y^2(n)} \text{ with } N = (n_2 - n_1 + 1)$$

セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート 開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。

例

以下の例では振幅が 10 V である 50 Hz の正弦波を生成し、完全な波形を使用して RMS 値を計算します。

```
Signal = 10.0 * @SineWave(20k; 1000; 50)
RMSAmpl = @RMS(Formula.Signal)
```

以下も参照してください。

"@Energy" ページ 55、"@Mean" ページ 116 および"@StdDev" ページ 165

4.71 @SAEJ211Filter

関数

SAE J211 の基準に従って入力信号をフィルタリングします。

構文

@SAEJ211Filter (信号; CFC)

パラメータ

信号	入力波形
CFC	番号：チャンネル周波数クラス

出力

出力はフィルタリングされた波形になります。

記述

この関数はデジタル低パスフィルタリングを行って、生データから高い周波数のノイズを除去します。フィルタは、4端子フェーズなし低パスのバターワースフィルタです。フィルタのタイプは、自動車業界において広く採用されているSAE J211 の基準に準拠しています。

カットオフ周波数 (Fc) とチャンネル周波数クラス (CFC) の間の関係は、以下のように定義されます。

$$F_c = CFC * 1.67$$

カットオフ周波数は、サンプリング周波数の 4 分の 1 を下回るべきではありません。

例

以下の例では、高い周波数のノイズをある程度含む正弦波を生成します。信号のフィルタリングは、SAE J211 の基準に準拠したフィルタリング関数を使って行われます。

```
Signal    = 5 * @SineWave(80000; 1000; 400)
          + @Noise(80000; 1000)
Filtered  = @SAEJ211Filter(Formula.Signal; 300)
```

カットオフ周波数は、 $F_c = 1.67 * 300 = 500$ Hz となります。

4.72 @Sin

関数

入力パラメータの正弦を計算します。

構文

@Sin (*Par*)

パラメータ

Par 入力波形または数値

出力

入力の正弦を含む波形または数値

記述

三角関数の正弦は、入力パラメータがラジアン角度であると想定された上で計算されます。波形パラメータが使用されると、各サンプルについて正弦が個別に計算されます。

例

以下の例では、度数で指定された変数「角度」の正弦を計算します。

```
Angle      = 36
AngleRad   = System.Constants.Pi * Formula.Angle / 180
SinAngle   = @Sin(Formula.AngleRad)
```

以下も参照してください。

"@ATan" ページ 42、"@Cos" ページ 46 および"@Tan" ページ 168

4.73 @SineWave

関数

正弦関数によって説明される波形を生成します。

構文

@SineWave (*F* サンプリング; *N* サンプル; *F* 信号)

パラメータ

F サンプリング番号 : サンプリング周波数

N サンプル サンプルの数

F 信号 番号 : 信号周波数。

出力

正弦波を含む波形。

記述

この関数は正弦波を生成します。正弦波の周波数を生成するサンプリング周波数およびサンプルの数を指定することができます。波形の長さに制限はありません。正弦波の振幅は 1 V です。

この関数を使用して、様々な波形を合成することができます。このデータを、他の分析関数のための入力として使用することができます。

例

以下の例では、4 V のピーク間ノイズを追加した状態で 10 kHz でサンプリングした、波形が 10 V であるシミュレートした 50 Hz の正弦波を含む 100 ms の波形セグメントを生成します。

```
Signal = 10 * @SineWave(10k; 1000; 50) + 2 *
          @Noise(10k; 1000)
```

より複雑な波形については、複数の正弦波を追加することが可能です。

```
Signal2 = 5 * @SineWave(10k; 1000; 50)
          + 7 * @SineWave(10k; 1000; 60)
          + 12 * @SineWave(10k; 1000; 90)
```

以下も参照してください。

"@Noise" ページ 129、"@Pulse" ページ 141、"@Ramp" ページ 145 および
"@SquareWave" ページ 164

4.74 @Smooth

関数

移動平均フィルタを使用して、波形を円滑化します。

構文

@Smooth (波形; N)

パラメータ

波形	円滑化する波形
N	円滑化率のサンプルの数

出力

円滑化した波形

記述

波形の円滑化は、以下の公式に従って指定されたサンプルの数の移動重みなし平均を求めて行うことができます。

$$\text{Output}(n) = \frac{1}{N} \sum_{k=n-N'}^{k=n+N'} \text{Input}(k)$$

このとき、

N = 円滑化率

N' = (N-1) / 2

n = 1 から「波形の終了」までの範囲におけるサンプルの数

円滑化率 N は、平均を出すために使用されるサンプルの数をコントロールします。N が奇数の整数ではないとき、最初に最も近い奇数の整数に四捨五入されます。値が 3 を下回る場合、3 に設定されます。値が 1001 を超える場合、1001 に設定されます。N の値が高ければ、円滑化の効果はより顕著になります。

円滑化によって、高い周波数のコンポーネントが効果的にフィルタリングされます。サンプルの数が妨害の期間の整数の倍数分に等しい場合に、最良のフィルタリングが行われます。円滑化は対称性を備えているため、波形の特性の位置は変わりません。

出力される波形の長さは、入力される波形の長さに等しくなります。エッジ付近での円滑化のために、関数は波形がエッジと同じデータ値で拡張されたものと想定します。

三角重み付き円滑化を実施することが求められる場合、波形の円滑化を 2 回実施することでこれを行うことができます。高い周波数の妨害のみを抑えるには、多数のサンプルを使用したときよりも少ない数のサンプルを使用することで、@Smooth 関数を複数回使用するとより効果的です。

例

以下の例では、元の信号から円滑化した信号を差し引くことで、50 Hz の正弦波から高い周波数のノイズを抽出します。

```

Mains = 220 * @SineWave(200k; 4096; 50)
        + 10 * @Noise(200k; 4096)
Smol  = @Smooth(Formula.Mains; 55)
Noise = Formula.Mains - Formula.Smol

```

以下も参照してください。

"@Integrate" ページ 97

4.75 @Sqrt

関数

特定のパラメータの平方根を計算します。

構文

@Sqrt (*Par*)

パラメータ

Par 入力波形または数値

出力

入力の平方根を含む波形または数値

記述

この関数は、入力されたパラメータの平方根を計算します。

波形パラメータが使用されると、各サンプルについて平方根関数が個別に計算されます。波形に含まれる負の値は、絶対値の負の平方根となります。平方根の数値については、負の値は存在しません。

例

可能な平方根の演算を以下に示します。

```
Signal = 4 * @SineWave(10k; 1000; 50)
Result = @Sqrt(Formula.Signal)
Five   = @Sqrt(25)
NoValue = @Sqrt(-1)
```

以下も参照してください。

"@Pow" ページ 134

4.76 @SquareWave

関数

平方根を含む波形を生成します。

構文

@Syntax (*F* サンプルング; *N* サンプル; *N* 周波数)

パラメータ

F サンプルング番号 : 波形のサンプル周波数。

N サンプル : 波形に含まれるサンプルの数。

N 周波数 : 番号 : 生成された方形波の周波数。

出力

方形波を含む波形。

記述

この関数は方形波を生成します。 サンプルング周波数、生成するサンプルの数、および方形波の周波数を指定することができます。 波形の長さに制限はありません。 方形波の振幅は 1 V です。

方形波関数を生成できることを利用して、様々な波形を合成することができます。 シミュレーションしたデータを、他の分析関数のための入力として使用することができます。

例

以下の例では、2 V のピーク間ノイズを追加した状態で 10 kHz でサンプルングした、波形が 10 V であるシミュレートした 50 Hz の方形波を含む 100 ms の波形セグメントを生成します。

```
Signal = 10 * @SquareWave(10k;1000;50)
        + 2 * @Noise(10k;1000)
```

以下も参照してください。

"@Noise" ページ 129、"@Pulse" ページ 141、"@Ramp" ページ 145 および "@SineWave" ページ 160

4.77 @StdDev

関数

波形の標準偏差を計算します。

構文

@StdDev (波形)

@StdDev (波形; 開始)

@StdDev (波形; 開始; 終了)

パラメータ

波形	標準偏差が計算される入力波形
開始	番号 : セグメントの開始。
終了	番号 : セグメントの終了。

出力

出力は数値です。

記述

標準偏差は、波形の一つの統計学的な特性です。これはほぼ、波形の平均値からの波形におけるサンプルの二乗平均平方根の偏差です。すなわち、値のスプレッドの測定値です。

標準偏差は、以下の公式を使って計算されます。

$$\text{StdDev} = \sqrt{\frac{1}{N-1} \sum_{n=n_1}^{n_2} (y(n) - \bar{y})^2}$$

$$\text{in which: } \bar{y} = \frac{1}{N} \sum_{n=n_1}^{n_2} y(n)$$

$$\text{with } N = (n_2 - n_1 + 1)$$

セグメントの限界 (開始と終了) を使用して一定の範囲のサンプルが選択されます。セグメントの限界が指定されていない場合、波形全体が使用されます。

開始のみが指定されている場合、波形の開始から終了までの波形セグメントが使用されます。

ノート **開始と終了のパラメータはサンプルとしてではなく、水平方向の軸を単位として (時間など) 指定されなければなりません。**

例

以下の例では、100 の DC コンポーネントに加えて 1 の RMS 振幅を持つ 50 Hz の正弦波を生成します。この信号から標準偏差が判断されます。

```
Signal = 100 + @Sqrt(2) * @SineWave(20k; 2000; 50)
StdDev = @StdDev(Formula.Signal)
```

以下も参照してください。

"@Energy" ページ 55、"@Mean" ページ 116 および"@RMS" ページ 156

4.78 @Sweep

関数

マルチスイープの記録から特定のスイープを選択します。

構文

@Sweep (信号; N スイープ)

パラメータ

信号 波形：マルチスイープの記録。
N スイープ 選択されるスイープの整数

出力

選択されたスイープのみを含む波形

記述

レコーダのハードウェアとソフトウェアでいくつかの設定を行うことで、マルチスイープの記録を行うことが可能になります。有効なそれぞれのトリガイベントが新しいスイープを生成します。マルチスイープの記録からのチャンネルの波形は、複数のスイープを含んでいます。

@Sweep 関数は、波形から特定のスイープを 1 つ選択することを可能にします。N スイープは、1 からスイープの数までの範囲の数値です。

ノート 選択された波形には、スイープが一つだけ含まれています。

例

この例では、記録の 2 番目のスイープにおける圧力の最大値を計算します。

```
PressureSweep = @Sweep
                (Active.Group1.Recorder_A.Ch_A2; 2)
MaxPressure   = @Max (Formula.PressureSweep)
```

4.79 @Tan

関数

入力パラメータの正接を計算します。

構文

@Tan (*Par*)

パラメータ

Par 入力波形または数値

出力

入力の正接を含む波形または数値

記述

三角関数の正接は、入力パラメータがラジアンのものであると想定された上で計算されます。波形パラメータが使用されると、各サンプルについて正接が個別に計算されます。

関数@ATan は、関数@Tan の逆三角関数です。

例

以下の例では、度数で指定された変数「角度」の正接を計算します。

```
Angle      = 72
AngleRad  = System.Constants.Pi * Formula.Angle / 180
TanAngle  = @Tan (Formula.AngleRad)
```

以下も参照してください。

"@ATan" ページ 42、"@Cos" ページ 46 および"@Sin" ページ 159

4.80 @TriggerTime

関数

選択されたスイープのトリガ位置を戻します。

構文

@TriggerTime (波形)

@TriggerTime (波形; N スイープ)

パラメータ

波形 スイープを 1 つ以上含む波形
N スイープ 番号 : 選択されたスイープ、デフォルト = 1。

出力

X 単位でトリガ位置を含む数。

記述

レコーダのハードウェアとソフトウェアでいくつかの設定を行うことで、マルチスイープの記録を行うことが可能になります。有効なそれぞれのトリガイベントが新しいスイープを生成します。マルチスイープの記録からのチャンネルの波形は、各スイープのためのトリガ時間に関する情報を含んでいます。

@TriggerTime 関数は、指定されたスイープの数のための X 単位によるトリガ時間を取得することを可能にします。

この関数は、「SweepReview」表示などと組み合わせて使用することができます。こうした種類の表示には、単一のスイープが 1 つだけ示されます。表示は各スイープに移動することができます。表示されるスイープのトリガ位置は、常に 0 (ゼロ) を基準にします。また、カーソル値もスイープのトリガ位置を基準にします。カーソルの位置に関して計算を実施することが必要な場合、対応するトリガ時間を使用して、元の波形におけるカーソルの実際の位置を計算する必要があります。

例

この例は、マルチスイープの記録から 2 つ目のスイープのトリガ位置における電圧の最大値 20 ms の計算を示しています。

```
Signal      = Active.Group1.Recorder_A.Ch_A2
TPos       = @TriggerTime( Formula.Signal; 2)
MaxVoltage = @Max( Formula.Signal; Formula.TPos - 10m;
                  Formula.TPos + 10m)
```

以下も参照してください。
"@TriggerTimeToText" ページ 171

4.81 @TriggerTimeToText

関数

時間 - 日付という形式の文字列で、選択されたスイープのトリガ位置を戻します。

構文

@TriggerTimeToText (波形)

@TriggerTimeToText (波形; N スイープ)

@TriggerTimeToText (波形; N スイープ; 小数)

@TriggerTimeToText (波形; N スイープ; 小数; 時間形式)

@TriggerTimeToText (波形; N スイープ; 小数; 時間形式; データ形式)

パラメータ

波形	スイープを 1 つ以上含む波形	
N スイープ	番号 : 選択されたスイープ、デフォルト = 1	
小数	0 から 9 までの範囲である秒の分数を表示する小数の数。デフォルト = 3	
時間フォーマット	「相対」(デフォルト)	記録開始からの時間
	「ローカル」	ローカル時間
	「UTC」	協定世界時 (UTC)
日付形式	「なし」 (デフォルト) 日付情報なし	
	「短い」	短い日付文字列の表示
	「長い」	長い日付文字列の表示

出力

出力はトリガ位置を示す文字列です。

記述

レコーダのハードウェアとソフトウェアで様々な設定を行うことで、マルチスイープの記録を行うことが可能になります。有効なそれぞれのトリガイイベントが新しいスイープを生成します。マルチスイープの記録からのチャンネルの波形は、各スイープのためのトリガ時間に関する情報を含んでいます。

@TriggerTimeToText 関数は、指定されたスイープのトリガ時間をテキスト文字列として取得することを可能にします。

この関数は、Perception の SweepReview 表示などと組み合わせて使用することができます。この表示には、単一のスイープが 1 つ示されます。表示は各スイープに移動することができます。スイープのトリガ位置は常に 0 (ゼロ) を基準とします。しかし、TriggerTimeToText 関数を使って、他の関係および様々な形式でトリガ時間を示すことが可能です。

日付と時間の形式は Windows の地域での設定に関連付けられているため、使用する機器によって異なる可能性があります。詳細については、Windows の Help を参照してください。

ノート この関数は内部の時間ベースを使って記録されたデータにのみ作用します。これは、x 軸の次元が時間を示していることを意味します。時間と日付の形式による文字列は、大文字と小文字を区別しません。戻り値は、(Excel などの)他のソフトウェアも文字列として使用、および解釈することができる文字列です。

例

この例では、UTC 時間での 2 番目のスイープのトリガ時間を示します。

```
Signal      = Active.Group1.Recorder_A.Ch_A2
TriggerUTC  = @TriggerTimeToText
              (Formula.Signal; 2; 6; "UTC"; "Short")
```

出力される文字列は、以下のようになります。"3-3-2008 10:40:45.301455"

時刻のみが必要な場合：

```
TriggerTime = @TriggerTimeToText
              (Formula.Signal; 2; 0; "Local"; "none")
```

結果は以下のようになります。"11:40:45"

以下も参照してください。

"@TriggerTime" ページ 169

4.82 @TrueRMS

関数

サイクル数毎に真の RMS (二乗平均平方根) を示す波形を戻します。

構文

@TrueRMS (波形)

@TrueRMS (波形; サイクル; 開始; 終了)

パラメータ

波形	RMS の計算のための波形
サイクル	RMS の計算のために使用されるサイクルの数。
開始	番号 : セグメントの開始
終了	番号 : セグメントの終了

出力

求められるサイクル数での RMS 値を含む波形。

記述

この関数は、元の信号と同じサンプル率を持つ新しい波形を生成します。生成される波形の実際の値は、供給される波形の真の RMS 値となります。デフォルト値は 1 サイクルです。他の値は 0.5 サイクルの倍数に変換されます。この関数はゼロ交差を使用してサイクルを決定します。サイクルのパラメータを使用して、計算のためにサイクルの数を 0.5 の倍数に設定します。

例

以下の例では、各サイクルにおける電圧の真の RMS 値を示す波形を生成します。完全な信号における平均の RMS の計算が行われます。

```
RMS_Signal = @TrueRMS
              (Active.Group1.Recorder_Voltage.Voltage;1)
AverageRMS = @Mean(Formula.RMS_Signal)
```

以下も参照してください。

"@RMS" ページ 156

4.83 @Value**関数**

指定された x 位置での波形の振幅値を戻します。

構文

@Value (*波形; XPos*)

パラメータ

波形 指定された x 位置での値が決定される波形。
X 位置 番号 : 波形の値が決定される x 位置。

出力

指定された x 位置での波形の振幅値。

記述

指定された x 位置における波形の値が決定されます。 x 位置はサンプル数ではなく、 x 座標として指定されます。 x 座標が 2 つのサンプリング地点の間にあるとき、線形補間を使ってこの x 位置での値が決定されます。

x 位置が波形データセットの最初のサンプルの前、または最後のサンプルの後に存在する場合、関数は値を戻しません。

例

以下の例では、レベル 2.5 の上昇する平面交差のためのマーカ (Marker) という信号を検索し、変数時間マーカ (TimeMarker) に見つかった時間を保存します。そして、この位置で別の信号の値が決定されます。実際の信号が想定されます。

```
Marker           = Active.Group1.Recorder_A.Ch_A1
OtherSignal      = Active.Group1.Recorder_A.Ch_A2
TimeMarker       = @NextLvlCross (Formula.Marker; 0; 2.5; 1)
ValueAtMark     = @Value (Formula.OtherSignal;
                           Formula.TimeMarker)
```

4.84 @XDelta

関数

波形のサンプリング間隔(サンプル間に経過する時間)を戻します :

構文

@XDelta (波形)

パラメータ

波形 入力波形。

出力

出力は数値です。

記述

この波形は、波形のサンプリング間隔 (期間) を戻します。 サンプリング間隔はサンプリング周波数の逆数で、隣接するサンプルの間の差を単位 x で示します。

例

以下の例では、サンプリング周波数が 1 kHz である正弦波を生成します。 変数 dx はこの信号のサンプリング間隔 (1 ms) を受け取ります。 変数 sf は、以下のサンプリング周波数を受け取ります。

```
Signal = 25 * @SineWave(1k; 1000; 50)
dx      = @XDelta(Formula.Signal)
sf      = 1.0 / @XDelta(Formula.Signal)
```

以下も参照してください。

"@Length" ページ 105、"@XFirst" ページ 178 および"@XLast" ページ 179、

4.85 @XDeltaHigh

関数

波形の最大のサンプリング間隔を決定します。

構文

@XDeltaHigh (波形)

パラメータ

波形 最大のサンプリング間隔が決定される波形。

出力

出力は数値です。単位 x による 2 つのサンプル間の差。

記述

マルチ時間ベースの波形について作業を行う場合、波形の最大のサンプリング間隔が決定されます。サンプリング間隔はサンプリング周波数の逆数で、隣接するサンプルの間の差を単位 x で示します。

例

この例では、2 つの異なるサンプル率を含む波形を生成します。XDelta 関数を使ってサンプリング周波数を決定します。

```
SignalLow = @SineWave(10k;10k;50)
SignalHigh = @SineWave(100k;10k;50)
Signal_LHL = @Join(Formula.SignalLow;
                    Formula.SignalHigh; Formula.SignalLow)
XDelta      = @XDelta(Formula.Signal_LHL)
XHigh       = @XDeltaHigh(Formula.Signal_LHL)
XLow        = @XDeltaLow(Formula.Signal_LHL)
```

以下も参照してください。

"@XDelta" ページ 175 および "@XDeltaLow" ページ 177

4.86 @XDeltaLow

関数

波形の最小のサンプリング間隔を決定します。

構文

@XDeltaLow (波形)

パラメータ

波形 最大のサンプリング間隔が決定される波形。

出力

出力は数値です。単位 x による 2 つのサンプル間の差。

記述

マルチ時間ベースの波形について作業を行う場合、波形の最小のサンプリング間隔が決定されます。サンプリング間隔はサンプリング周波数の逆数で、隣接するサンプルの間の差を単位 x で示します。

例

この例では、2 つの異なるサンプル率を含む波形を生成します。XDelta 関数を使ってサンプリング周波数を決定します。

```
SignalLow = @SineWave(10k;10k;50)
SignalHigh = @SineWave(100k;10k;50)
Signal_LHL = @Join(Formula.SignalLow;
                    Formula.SignalHigh; Formula.SignalLow)
XDelta     = @XDelta(Formula.Signal_LHL)
XHigh     = @XDeltaHigh(Formula.Signal_LHL)
XLow      = @XDeltaLow(Formula.Signal_LHL)
```

以下も参照してください。

"@XDelta" ページ 175 および "@XDeltaHigh" ページ 176

4.87 @XFirst

関数

波形における最初のサンプルの x 座標を戻します。

構文

@XFirst (波形)

パラメータ

波形 入力波形。

出力

出力は数値です。

記述

最初のサンプルの x 座標が戻されます。 x 座標は、波形の水平方向の単位に関する値です (通常は時間)。

例

以下の例では、正弦波を生成します。生成された信号は、常に時間 0 から開始します。波形は、水平方向に 100 ms シフトします。このシフトした波形の最初のサンプルの x 座標が決定される場合、100 ms となります。

```
Signal = @SineWave(10k; 1000; 50)
Shifted = @XShift(Formula.Signal; 100m)
Start = @XFirst(Formula.Shifted)
```

以下も参照してください。

"@Length" ページ 105、"@XDelta" ページ 175、"@XLast" ページ 179 および "@XShift" ページ 180

4.88 @XLast

関数

波形における最後のサンプルの x 座標を戻します。

構文

@XLast (波形)

パラメータ

波形 入力波形。

出力

出力は数値です。

記述

最後のサンプルの x 座標が戻されます。 x 座標は、波形の水平方向の単位に関する値です (通常は時間)。

例

以下の例では、正弦波を生成します。生成された信号は、常に時間 0 から開始します。この例では、波形における最後のサンプルの x 座標を決定するための 2 つの方法を示しています。

```
Signal = @SineWave(10k; 1000; 50)
XEnd1  = @XFirst(Formula.Signal) +
         (@Length(Formula.Signal) - 1) *
         @XDelta(Formula.Signal)
XEnd2  = @XLast(Formula.Signal)
```

以下も参照してください。

"@Length" ページ 105、"@XDelta" ページ 175、"@XFirst" ページ 178 および "@XShift" ページ 180

4.89 @XShift

関数

波形を、指定された時間で水平にシフトします。

構文

@XShift (波形, シフト)

パラメータ

波形 水平方向にシフトする波形。
シフト 番号：シフトの規模。

出力

時間でシフトした元の波形。

記述

この関数は波形の内容を変更しませんが、波形が指定された時間（またはその他の単位）で水平方向にシフトするように、水平方向のスケーリングに関する情報を変更します。正の値は波形を右方向にシフトさせ、負の値は波形を左方向にシフトさせます。

例

以下の例では、波形の水平方向の軸が常に 0（ゼロ）から開始します。実際の信号が想定されています。

```
Signal      = Active.Group1.Recorder_A.Ch_A1
Shift       = -1 * @XFirst(Formula.Signal)
CorrSignal = @XShift(Formula.Signal; Formula.Shift)
```

以下も参照してください。

"@XDelta" ページ 175、"@XFirst" ページ 178 および"@XLast" ページ 179

4.90 @XYArray

関数

二次元配列を使用して波形を生成します。

構文

@XYArray (X1; Y1; ...; ...; XN; YN)

パラメータ

X1	最初のサンプルの X 値。
Y1	最初のサンプルの Y 値
XN	最後のサンプルの X 値; N>= 2.
YN	最後のサンプルの Y 値; N>= 2.

出力

配列のすべての地点を含む波形。

記述

この関数は、パラメータ X1、Y1、X2、Y2 などが通過するすべての地点を含む波形を生成します。入力された地点は、時間で整列されるべきです。

出力される波形は、等距離のデータ地点を含んでいます。この関数は入力 X 値を評価し、可能となる最良の新しいサンプル率を定義します。サンプル率として、1、2、5 という一連の数が使用されます。

X1 = 0 ms、X2 = 30 ms、X3 = 50 ms を想定します。これにより、サンプリング間隔 20 ms が生じます。

X1 = 25 ms、X2 = 30 ms、X3 = 50 ms を想定します。これにより、サンプリング間隔 5 ms が生じます。

X1 = 2 s、X2 = 12 s、X3 = 15 s を想定します。これにより、サンプリング間隔 2 s が生じます。

ノート 出力されるサンプルは等距離です。サンプルの数は、入力されるサンプルの数よりも大きくなる場合があります。

例

以下の例では、5 つの地点のみが入力される一方で、8 つのデータ地点を含む波形が生成されます。XY 配列の関数によって 3 つの新しい地点が生成されます。

```
NewWave = @XYArray(0; 0; 1; 2; 2; 1; 5; -3; 7; 0)
```

以下も参照してください。
"@YArray" ページ 183

4.91 @YArray

関数

この関数は、単一次元配列をデータ入力として使用して波形を生成します。

構文

@YArray (*F* サンプルング; *Y1*; ...; *YN*)

パラメータ

F サンプルング番号 : サンプルング周波数

Y1 番号 : 最初のデータポイントの振幅 (*Y* 値)

YN 番号 : 最後のデータポイントの振幅 (*Y* 値) $N \geq 2$

出力

出力は、配列のすべてのデータポイントを含む波形となります。

記述

この関数は、パラメータ *Y1* から *YN* までが通過するすべての地点を含む波形を生成します。最初のパラメータはサンプルング周波数 (= $1 /$ サンプルング期間) を定義します。

例

以下の例では、サンプルング周波数が 100 Hz である 9 つのデータポイントを含む波形を生成します。

```
Signal = @YArray(100; 0; 1; 2; 2; 1; -1; -3; 2; 0)
```

以下も参照してください。

"@XYArray" ページ 181

5 パルス測定および分析

5.1 一般事項

IEEE 標準パルスの用語と定義

本章の内容は、IEEE 標準 194-1977 と IEEE 標準 181-1977 に基づいています。

詳細については、IEEE 標準協会のウェブサイトを参照してください。

standards.ieee.org/

波、パルス、および移行

- **波**：ある媒体の物理的な状態の変化で、一つまたは複数の妨害によって時間の関数として媒体で広まります。
- **パルス**：最初の公称の状態から発生し、2つ目の公称の状態を得て、最終的に最初の公称の状態に戻ります。この資料の残りの部分においては、パルスという用語は波という用語に含まれるものとします。
- **移行**：最初の公称の状態の間における波またはパルスの一部。この資料の残りの部分においては、移行という用語はパルスと波という用語に含まれるものとします。

単一パルス波形の概要については、次の図を参照してください。

単一パルス

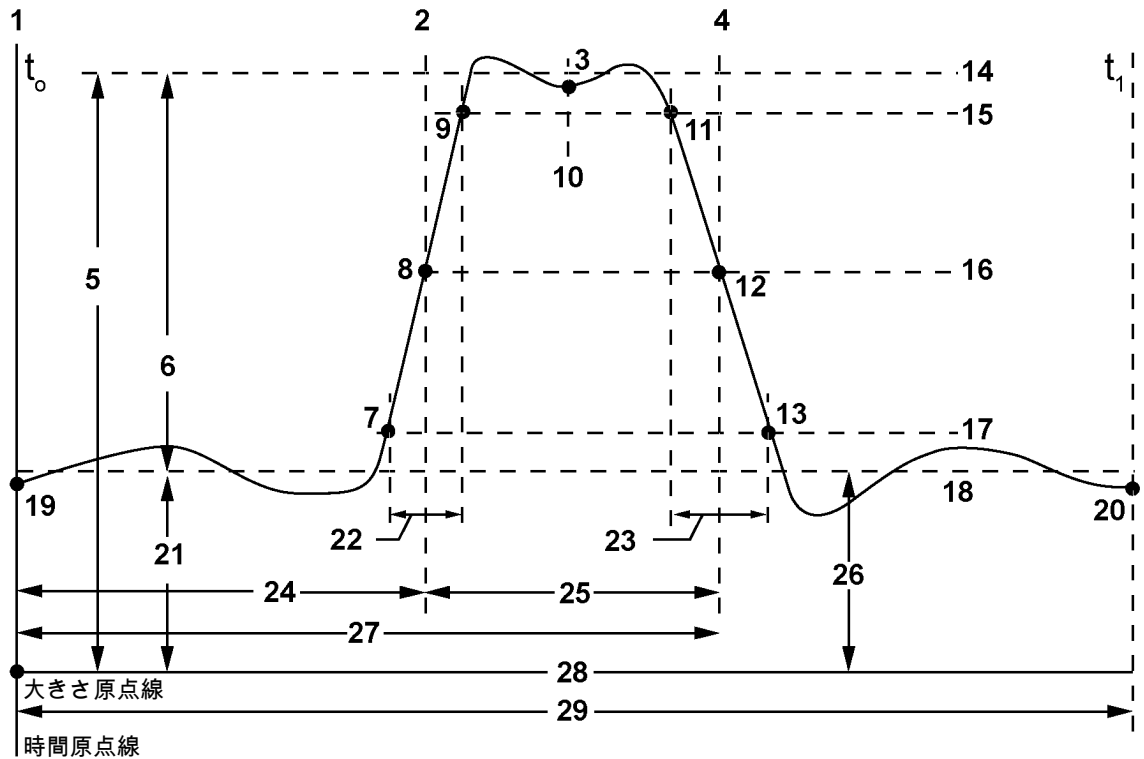


イラスト 5.1: 単一パルス波形の図

- | | | |
|---|--------------|--|
| 1 | 時間原点線 | 定数、および他に指定がなければ 0 (ゼロ) に等しい時間を持ち波形期間の最初のデータ時間 t_0 を過ぎる指定された時間の線。 |
| 2 | パルス開始線 | パルス開始線。パルス開始時間での時間リファレンス線。 |
| 3 | 上心点 | パルス波形上において指定された時間基準地点または大きさ基準地点。地点が指定されていない場合、上心点はパルス波形と上心線の交点における時間基準地点となります。 |
| 4 | パルス停止線 | パルス停止線。パルス停止時間での時間リファレンス線。 |
| 5 | 最上部の大きさ | 指定された手順またはアルゴリズムによって得られた最上部の大きさ。 |
| 6 | パルス振幅 | 最上部の大きさと底の大きさの間の代数的な差。 |
| 7 | 近位 (最初の移行地点) | 波形と近位線の交点における大きさの基準地点。 |
| 8 | 近心 (最初の移行地点) | 波形と近心線の交点における大きさの基準地点。 |

- | | | |
|----|----------------|--|
| 9 | 遠位 (最初の移行地点) | 波形と遠位線の交点における大きさの基準地点。 |
| 10 | 上心線 | パルス開始時間とパルス停止時間の平均での時間リファレンス線。 |
| 11 | 遠位 (最後の移行地点) | 波形と遠位線の交点における大きさの基準地点。 |
| 12 | 近心 (最後の移行地点) | 波形と近心線の交点における大きさの基準地点。 |
| 13 | 近位 (最後の移行地点) | 波形と近位線の交点における大きさの基準地点。 |
| 14 | 最上線 | 最上部の大きさに存在する大きさリファレンス線。 |
| 15 | 遠位線 | パルス波形の遠位領域で指定された大きさにおける大きさリファレンス線。他に指定がなければ、遠位線は 90%のリファレンスの大きさに存在します。 |
| 16 | 近心線 | パルス波形の近心領域で指定された大きさにおける大きさリファレンス線。他に指定がなければ、近心線は 50%のリファレンスの大きさに存在します。 |
| 17 | 近位線 | パルス波形の近位領域で指定された大きさにおける大きさリファレンス線。他に指定がなければ、近位線は 10%のリファレンスの大きさに存在します。 |
| 18 | 基線 | 底の大きさに存在する大きさリファレンス線。 |
| 19 | 最初のベースポイント | 他に指定がなければ、パルス期間における最初のデータ地点。 |
| 20 | 最後のベースポイント | 他に指定がなければ、パルス期間における最後のデータ地点。 |
| 21 | 底の大きさ | 指定された手順またはアルゴリズムによって得られた底の大きさ。他に指定がなければ、底の両方の部分は手順またはアルゴリズムに含まれます。 |
| 22 | 最初のトランジットの継続時間 | パルス波形における最初の移行波形のトランジットの継続時間。 |
| 23 | 最後のトランジットの継続時間 | パルス波形における最後の移行波形のトランジットの継続時間。 |
| 24 | パルス開始時間 | パルス波形の最初の移行における大きさ基準地点により指定される瞬間。他に指定がなければ、パルス開始時間は最初の移行の近心地点にあります。 |
| 25 | パルス継続時間 | パルス開始時間とパルス停止時間の間の継続時間。 |

- | | | |
|----|------------------|---|
| 26 | Offset (オフセット) | 指定された2つの大きさリファレンス線の間の代数的な差。他に指定がなければ、2つの大きさリファレンス線は波形基線および大きさ原点線となります。 |
| 27 | パルス停止時間 | パルス波形の最後の移行における大きさ基準地点により指定される瞬間。他に指定がなければ、パルス停止時間は最後の移行の近心地点にあります。 |
| 28 | 大きさ原点線 | 他に指定がなければ0 (ゼロ) に等しい大きさを持ち波形期間にかけて伸びる指定された大きさの線。 |
| 29 | パルス波形期間 | 波形データが知られる、または知られる可能性がある期間。等式によって明示される波形期間は - 無限から + 無限に延びる可能性があり、またすべての波形データのように最初のデータ時間から t_0 2 目のデータ時間 t_1 に延びる可能性があります。 |

6 IIR フィルタ

6.1 はじめに

Perception の公式データベースには一連の IIR フィルタが含まれています。これらのフィルタは、フィードバックおよびフィードフォワード係数によって指定される **直接型 IIR フィルタ** を使用して入力シーケンスをフィルタリングするクラスを実装しています。IIR とは、Infinite Impulse Response (無限インパルス応答) の略です。出力信号と入力信号との関係はイラスト 6.1 の公式のようになります。

$$y(n) = \sum_{k=0}^N b_k x(n-k) - \sum_{k=1}^N a_k y(n-k)$$

イラスト 6.1: Infinite Impulse Response (無限インパルス応答) – 出力、入力

ここで、

N = フィルタ次数

b_i = フィードフォワードフィルタ係数、ゼロ数とも呼ばれます。

a_i = フィードバックフィルタ係数、極数とも呼ばれます。

$x(n)$ = 入力信号

$y(n)$ = 出力信号

ブロック図はイラスト 6.2 のようになります。

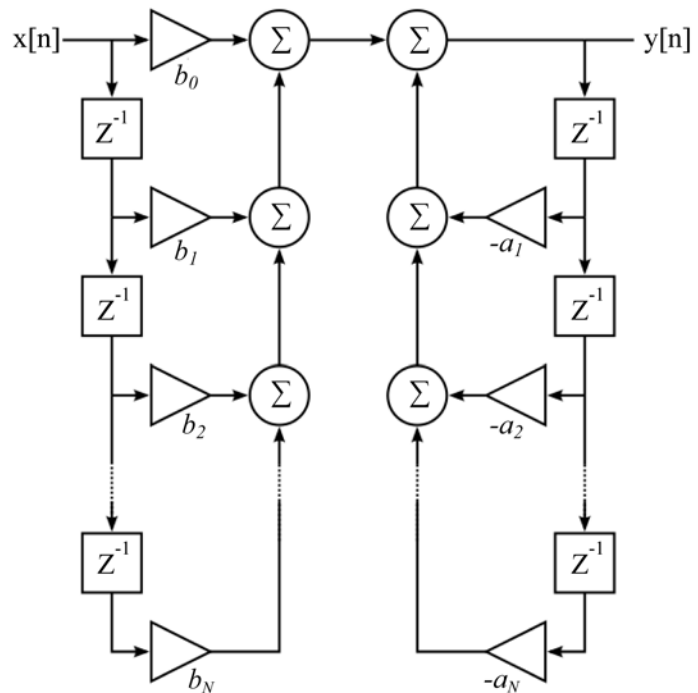


イラスト 6.2: 無限インパルス応答 – ブロック図

フィルタの公式はすべて「@Filter」で始まります。フィルタ関数の一覧を以下に示しています。

@FilterButterworthLP
 @FilterButterworthHP
 @FilterButterworthBP
 @FilterButterworthBS
 @FilterBesselLP
 @FilterBesselHP
 @FilterBesselBP
 @FilterBesselBS
 @FilterChebyshevLP
 @FilterChebyshevHP
 @FilterChebyshevBP
 @FilterChebyshevBS

フィルタには以下の 3 種類があります。

- ベッセル
- バターワース
- チェビシエフ

各種類に対して、低パス (LP)、高パス (HP)、バンドパス (BP) およびバンドストップ (BS) が実装されています。

6.1.1 ベッセル

ベッセルフィルタは、群遅延が最大限平坦な（位相応答が最大限線形な）線形フィルタタイプです。ベッセルフィルタは非常に少ないオーバーシュートで良好なステップ応答を実現しますが、バターワースフィルタよりも周波数応答が劣ります。ベッセルフィルタの位相応答はほぼ線形になります。ベッセルは位相関係が重要な場合に有用性があります。

このフィルタは時間領域における使用に最も適しています。

フィルタリングする信号に方形波形やノイズなど正弦曲線以外の信号が含まれており、フィルタリングした信号をタイミングの目的で使用したい場合にも、ベッセルフィルタを使用することをお勧めします。この場合、元の信号のさまざまな周波数コンポーネントのすべてでフィルタリング後にほぼ同じ遅延が生じることから、ベッセルは最適な選択肢です。

利点：

最良のステップ応答、非常に少ないオーバーシュートまたはリングング。

欠点：

通過帯域を超える減衰の初速度がバターワースより遅い。

6.1.2 バターワース

バターワースフィルタは、通過帯域における周波数応答が可能な限り平坦になるよう設計された信号処理フィルタタイプで、振幅が最大限平坦なフィルタとも呼ばれます。リップルのない良好な周波数応答を実現しますが、特に高次フィルタの場合に位相応答が極めて非線形になる場合があります。

このフィルタは周波数領域における使用に最も適しています。

フィルタリングする信号が正弦曲線信号の場合、バターワースは多くの場合に最適な選択肢です。入力信号における1つの優位周波数のみ処理するため、出力信号の変形は最小限になることから、非線形の位相応答はそれほど問題ではありません。

利点：

- 通過帯域における最大限平坦な振幅応答。
- 総合的に良好なパフォーマンス。
- チェビシェフより良好なパルス応答。
- ベッセルより優れた減衰率。

欠点：

- ステップ応答におけるある程度のオーバーシュートとリングング。

6.1.3 チェビシエフ (I 型)

チェビシエフフィルタは、バターワースフィルタよりロールオフが急峻で、通過帯域のリプルが大きく (I 型) なります。チェビシエフフィルタは、フィルタ範囲にわたって理想的なフィルタ特性と実際のフィルタ特性との間の誤差を最小限にするという性質を有していますが、通過帯域においてリプルが発生します。

バターワースフィルタと比較すると、チェビシエフフィルタは、より低次のフィルタで通過帯域と阻止帯域との間のより急峻な遷移を実現できます。

このフィルタは周波数領域における使用に最も適しています。

利点：

- 通過帯域を超える減衰率がバターワースより良好。

欠点：

- 通過帯域におけるリプル。
- ステップ応答においてバターワースより大幅にリングングが多い。

6.1.4 振幅スペクトル

以下のイラスト 6.3 は、3 種類の 2 次低パス IIR フィルタの振幅スペクトルを示しています。ベッセルフィルタは阻止帯域における周波数応答が最も劣ります。チェビシエフはロールオフが最も急峻ですが、通過帯域においてリプルが発生しています。

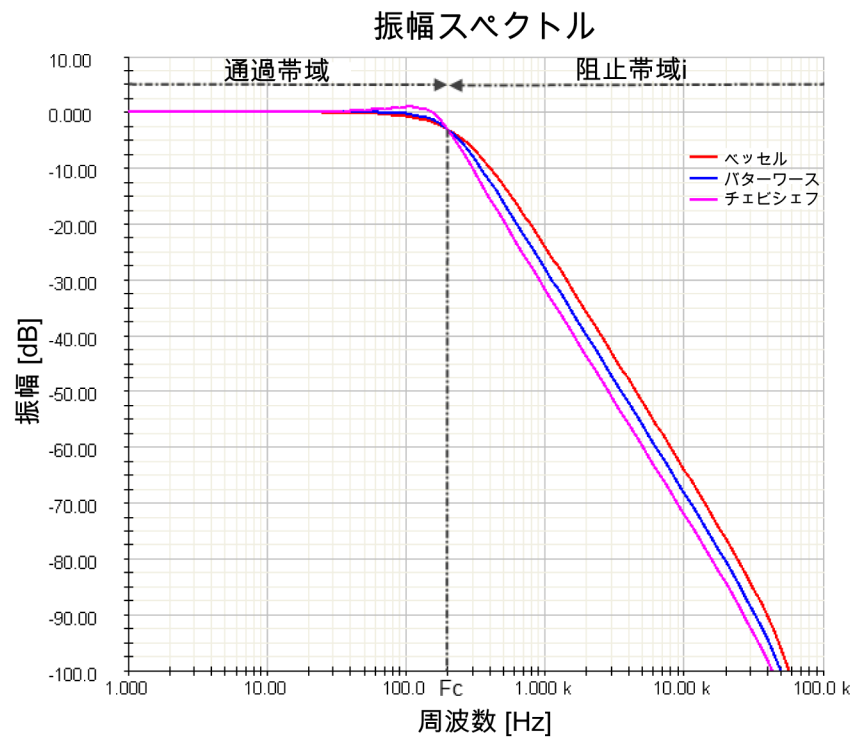


イラスト 6.3: 低パスフィルタの振幅スペクトル

イラスト 6.4 は、3 種類のフィルタの通過帯域を示しています。

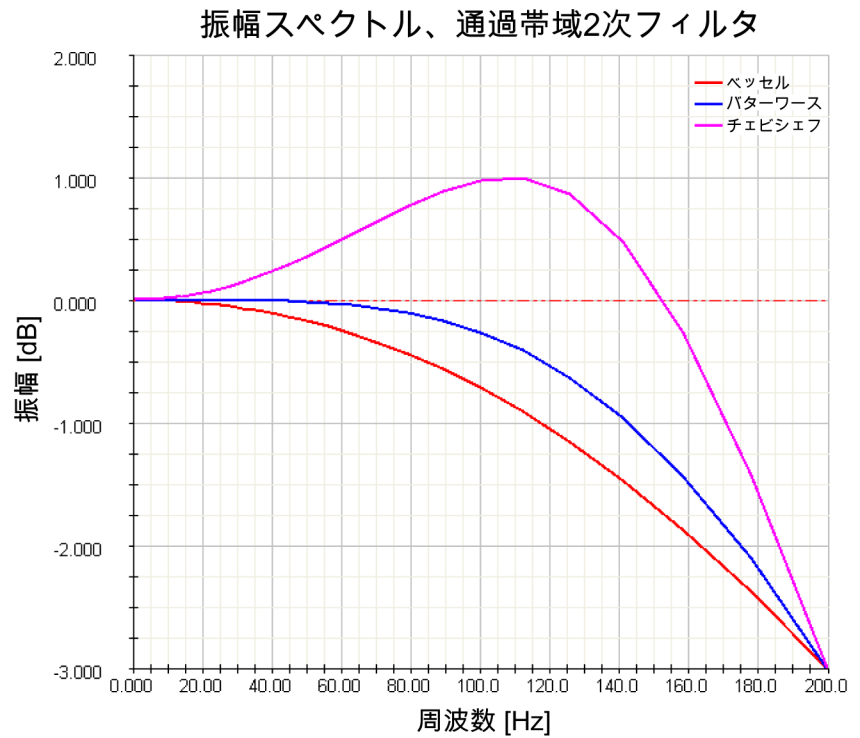


イラスト 6.4: 振幅スペクトル、フィルタの通過帯域

イラスト 6.5 は、3 種類のフィルタの阻止帯域を示しています。

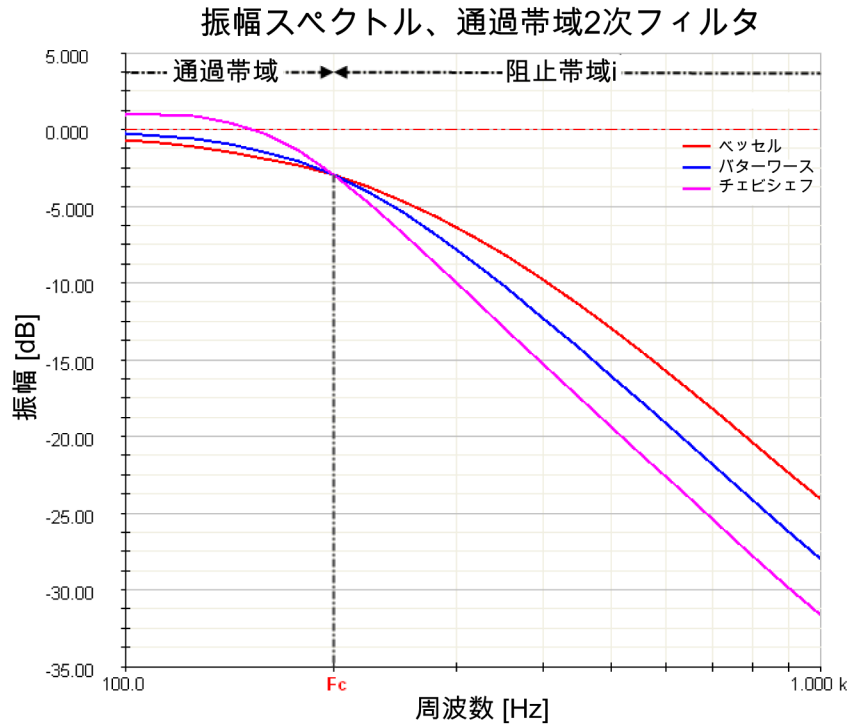


イラスト 6.5: 振幅スペクトル – 阻止帯域

6.1.5 インパルス応答

以下のイラスト 6.6 は、3 種類の 2 次低パス IIR フィルタのインパルス応答を示しています。ベッセルフィルタが最良のインパルス応答を示しています。

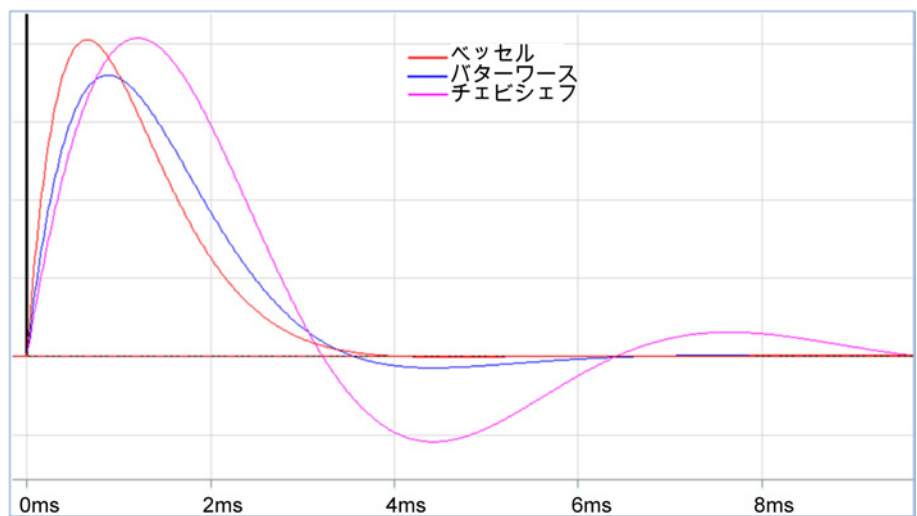


イラスト 6.6: 低パスフィルタのインパルス応答

6.1.6 ステップ応答

以下のイラスト 6.7 は、3 種類の 2 次低パス IIR フィルタのステップ応答を示しています。

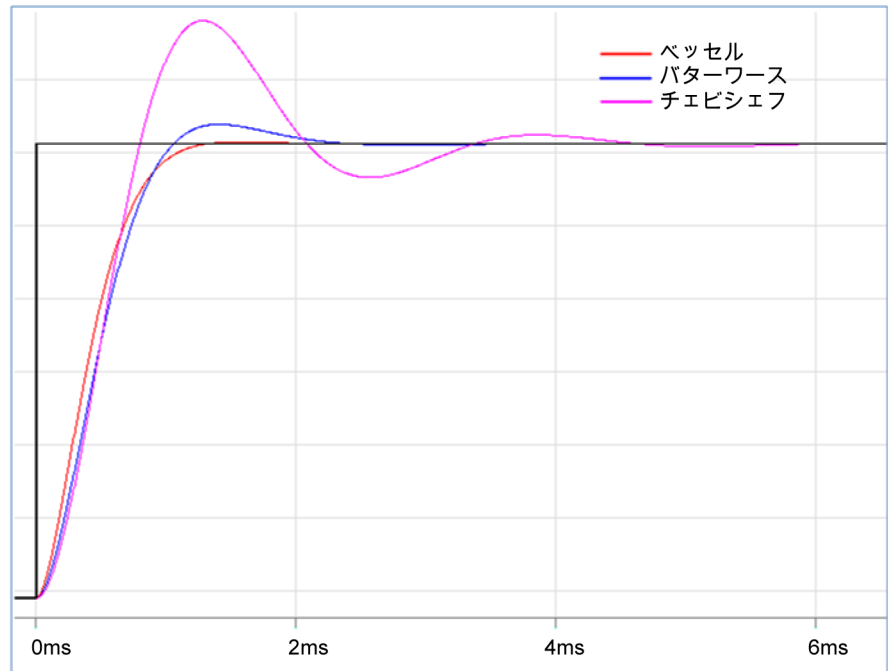


イラスト 6.7: 低パスフィルタのステップ応答

これについても、ベッセルフィルタが最良のステップ応答を示しています。上記のインパルス応答とステップ応答は以下の公式を使用して再現することができます。

Num	Name	Formula	Units
1	Pulse	@Pulse(80k; 80k; 20k)	
2	ButherworthPR	@FilterButterworthLP(Formula.Pulse; 2; 200)	
3	BesselPR	@FilterBesselLP(Formula.Pulse; 2; 200)	
4	ChebyshevPR	@FilterChebyshevLP(Formula.Pulse; 2; 200; 3)	
5			
6	Step	@Pulse(80k; 80k; 20k; 10k)	
7	ButherworthSR	@FilterButterworthLP(Formula.Step; 2; 200)	
8	BesselSR	@FilterBesselLP(Formula.Step; 2; 200)	
9	ChebyshevSR	@FilterChebyshevLP(Formula.Step; 2; 200; 3)	
10			

イラスト 6.8: フィルタの応答

上記の公式を使用して、Perception の既存のフィルタ公式で作成できる任意のフィルタのステップ応答とインパルス応答を確認することもできます。

6.1.7 フェーズレスフィルタリング

フェーズレスフィルタリングは、逆順序テクニックを使用して行われます。このテクニックでは信号を2回フィルタリングします。使用されるフィルタの次数は、入力した次数の値の半分 (次数/2) になります。最初の段階で信号は通常どおりにフィルタリングされ、2番目の段階で、フィルタリングされた出力が同じフィルタを使用して再度フィルタリングされますが、データポイントはフィルタを逆の順序で通過します。フィルタを2回使用するため、全体的なフィルタ次数は入力した次数の値 (次数) と等しくなります。入力した値が奇数の値の場合、フィルタ次数はこの奇数値より小さい最初の偶数値になります。例えば、入力した値が7の場合、フィルタの次数の値は6になります。フィルタリングされた信号の位相シフトはゼロになります。イラスト 6.9 を参照ください。

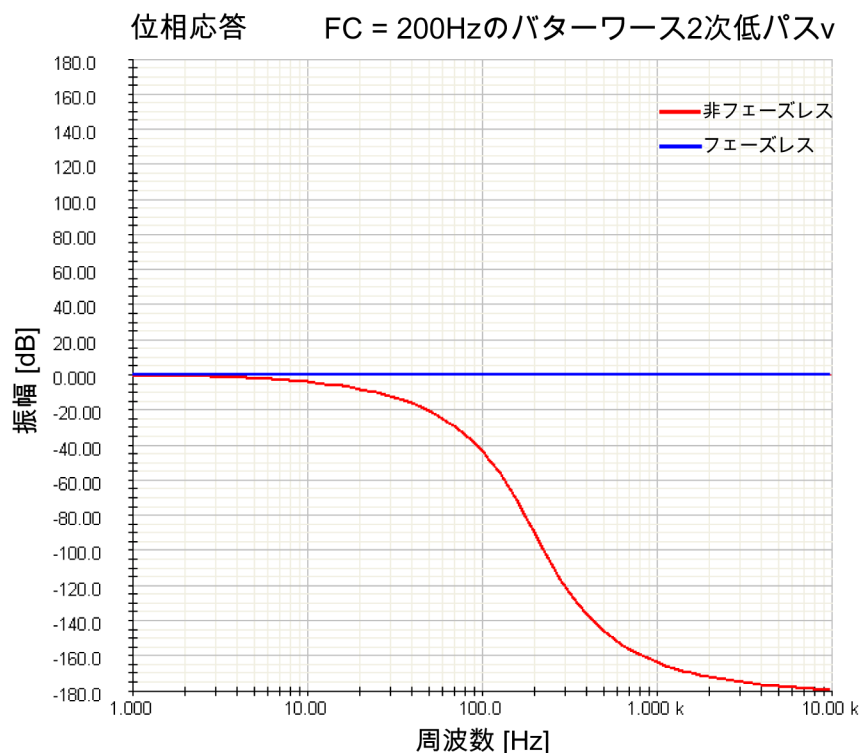


イラスト 6.9: フィルタの位相応答

フェーズレスと非フェーズレスフィルタの振幅スペクトルの間には誤差が発生します。この誤差は、フェーズレスフィルタが、半分の次数のフィルタで2回フィルタリングを行っているで、このためカットオフ周波数における減衰は-3 dBではなく、-6 dBになります。イラスト 6.10 を参照ください。

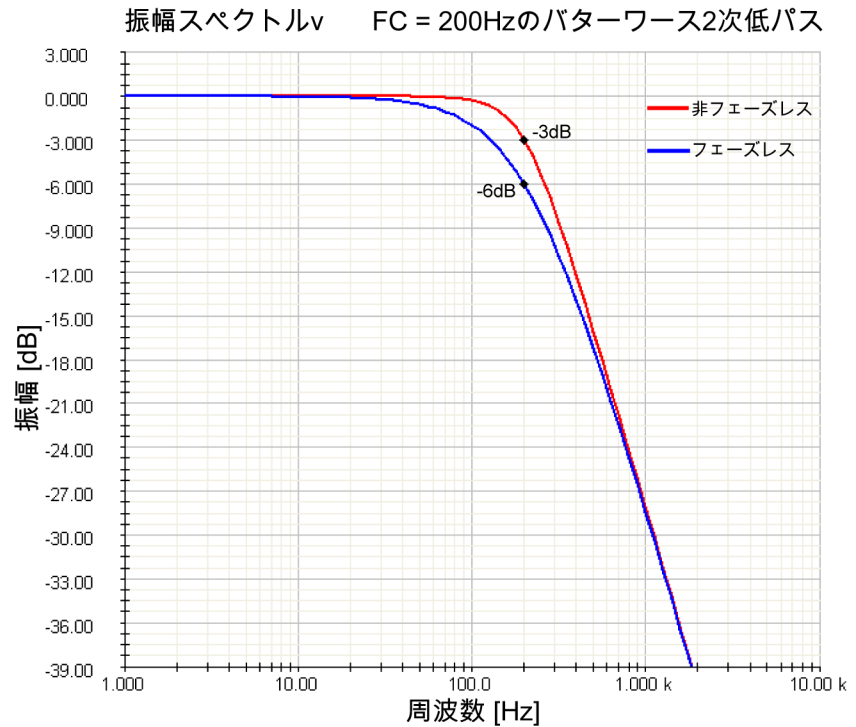


イラスト 6.10: 振幅ベクトル - 減衰

6.1.8 サンプリング率とカットオフ周波数の重要性

フィルタリングする個々の信号のサンプリング率とフィルタのカットオフ周波数の影響を認識しておくことが重要です。サンプリング率は、カットオフ周波数の2倍以上でなければなりません。これはナイキスト周波数としても知られています。

フィルタの振幅曲線もサンプリング周波数によって影響を受けます。2次フィルタのロールオフは 12 dB/オクターブまたは 40 dB/ディケードですが、サンプリング率の半分に近い周波数の場合、フィルタの挙動は異なります。以下のイラスト 6.11 を参照ください。デジタルフィルタリングとアナログフィルタリングを比較する場合はこの違いを認識していなければなりません。アナログフィルタは、阻止帯域全体に対して一定の傾斜でロールオフします。

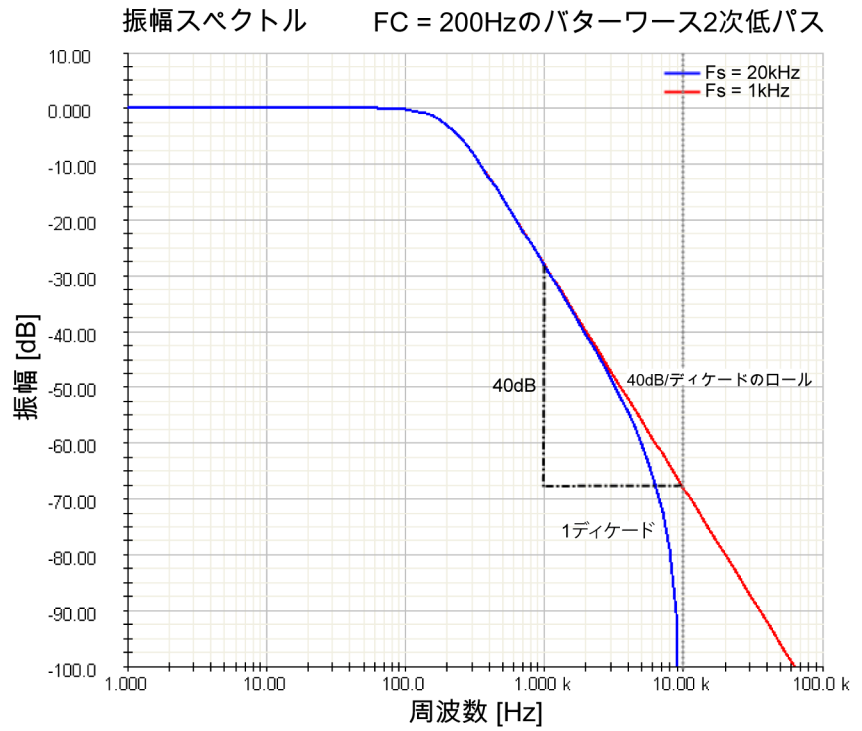


イラスト 6.11: 半分のサンプリング率での周波数

索引

A	
Abs	38
And	39
ASCII ファイルの読み込み	147
ATan	42
E	
EqualTo (等しい)	57
ExpWave	59
F	
FilterBesselBP	75
FilterBesselBS	77
FilterBesselHP	73
FilterBesselLP	71
FilterButterworthBP	67
FilterButterworthBS	69
FilterButterworthHP	65
FilterButterworthLP	63
FilterChebyshevBP	84
FilterChebyshevBS	86
FilterChebyshevLP	79
G	
GreaterEqualThan (以上)	91
GreaterThan (超える)	92
I	
IEEE 標準パルスの用語と定義	184
IIF	95
IIR Filters	
チエビシェフ (I 型)	191
IIR フィルタ	188
インパルス応答	194
カットオフ周波数	197
サンプリング率	197
ステップ応答	195
パワーワース	190
フェーズレスフィルタリング	196
ベッセル	190
振幅スペクトル	191
IntLookUp	98
IntLookUp12	101
J	
Join (結合)	103
L	
LessEqualThan (以下)	106
LessThan (未満)	107
Ln	108
Log (対数)	110
N	
NextHillPos	123
NextValleyPos	127
Not	130
O	
Or	131
P	
Pow	134
PrevHillPos	135
PrevLvlCross	137
PrevValleyPos	139
Pulse	141
R	
RemoveGlitch	152
RMS	156
S	
SAEJ211	158
T	
TriggerTimeToText	171

X		加算	29
XDelta	175	単一パルス	185
XDelta 低	177	印刷	
XDelta 高	176	公式	21
XY 配列	181	周波数	89
X シフト	180	平均	116
X 最初	178	平方根	163
X 最後	179	微分	53
		情報シート, 拡張済み (オプション)	
		ツール	11
Y		指数	58
Y 配列	183	方形波	164
		曲線フィッティング	47
		最大	111
		最大位置	114
		最大数値	113
		最小	118
		最小値の位置	121
		最小数値	120
		期間	132
		概要	24
		標準偏差	165
		次の平面交差	125
		正弦	159
		正弦波	160
		正接	168
		波、パルス、および移行	184
		減算	31
		発行人	2
		真の RMS	173
		積分	97
		立ち上り時間	154
		立ち下り時間	61
		長さ	105
		関数の概要	24
*			
エネルギー	55		
エリア	40		
クリップ	45		
サイクル	51		
スイープ	167		
トリガ時間	169		
ノイズ	129		
パルス幅	143		
ヒストグラム	93		
ブロック FFT	43		
ランプ	145		
リファレンス確認	150		
一般事項	23		
乗算	33		
低減	149		
余弦	46		
使用許諾契約	3		
例外	23		
保証	3		
値	174		
円滑化	161		
再サンプリング 2	153		
分析	9		
ツール	11		
レイアウトの修正	16		
作用素	11		
公式の保存	21		
公式の印刷	21		
公式の読み込み	20		
公式シート	11		
変数	14		
定数	14		
関数	14		
切り取り	49		

Head Office

HBM

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

France

HBM France SAS

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

UK

HBM United Kingdom

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

USA

HBM, Inc.

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

PR China

HBM Sales Office

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

© Hottinger Baldwin Messtechnik GmbH. All rights reserved.
All details describe our products in general form only.
They are not to be understood as express warranty and do
not constitute any liability whatsoever.

measure and predict with confidence

