
PL-U4101C1 PcWaveForm Archi_1 Script 演算関数仕様編

2022 年 4 月

Revision 1.18

お断り

記載されている会社名および製品名はその会社の所有する商標です。
記載された内容については事前のお断りなく変更させていただく場合がございます。
記載された内容は 2022 年 4 月現在のものです。
ご使用にあたっては、本取扱説明書の内容を十分お読みいただけますようお願い申し上げます。

本取扱説明書は、PDF 形式でプログラム CD の中に入っています。

株式会社 デイシー

〒198-0024

東京都青梅市新町 9-2190

電話: 0428-34-9860

メール: info@deicy.co.jp

© Copyright 2006- DEICY Corporation

※ Archi_1 Script とは PcWaveForm 上で記述し、任意の解析を行うことのできる機能を持った Script 言語体系の名称です。

改定履歴

発行日	Revision	内容
2010年03月20日	1.00	<p>初版</p> <p>演算関数増加に伴い PcWaveForm 解析機能操作編より演算関数部分を別冊としました。 同時に、新規関数の追加および従来の機能強化を行いました。</p> <p><追加された演算関数></p> <ol style="list-style-type: none"> 1. データ番号データ値埋め戻し関数 SBV(k,m,X) 2. 閾値通過データ番号関数 2 DTM(k,m,j,n,X) 3. 度数分布関数 TRC(k,m,X) 4. 度数分布統計値関数 TRX(k,m,X,Y,a) 5. データ検索関数 FND(k,X,Y)(新規関数) <p><機能強化した演算関数></p> <ol style="list-style-type: none"> 1. 最大値関数 MAX(k,X) 2. 最小値関数 MIN(k,X) 3. 平均値関数 MEA(k,X) 4. 合計値関数 SUM(k,X) 5. 実効値関数 EFF(k,X) 6. 標準偏差関数 STD(k,X) 7. データ個数関数 LEN(k,X) 8. ISO262. 14. /EN12299:2009 B 補正フィルタ WBB(k,X)
2010年06月21日	1.01	<p><機能強化した演算関数></p> <ol style="list-style-type: none"> 1. F-V 関数 FVC(k,X,m) 2. 度数分布統計値関数 TRX(k,m,X,Y,a) 3. 最大値位置データ番号検索関数 MPX(k,X) 4. 最小値位置データ番号検索関数 MNP(k,X) 5. 閾値通過データ番号関数 2 DTM(k,m,j,n,X) 6. 一定値区間データ番号関数 SWP(m,n,k,X) 7. 区間最大値/最小値データ番号関数 PMP(j,k,m,X) 8. MARK 番号データ番号関数 MRK(k) <p><新規追加した演算関数></p> <ol style="list-style-type: none"> 1. 倫理遷移データ番号検索関数 LST(k,X) 2. パルス差分値合成関数 ASS(X,Y)
2010年08月25日	1.02	<p><機能強化した演算関数></p> <ol style="list-style-type: none"> 1. 閾値通過データ番号関数 2 DTM(k,m,j,n,X,o)
2011年03月31日	1.03	<p><機能強化した関数></p> <ol style="list-style-type: none"> 1 移動実効値関数 RRV(k,X,n) フラグ追加 <p><新規追加した関数></p> <ol style="list-style-type: none"> 1. 聴感補正フィルタ C カーブ WCC(X) 2. 時間重み付け移動実効値関数 RTT(k,X,m) <p><修正した関数></p> <ol style="list-style-type: none"> 1 ロゼッタゲージ応力方向関数 ETH(a,b,c)最大応力方向を示す補正式修正
2011年04月28日	1.04	<p><機能強化した関数></p> <ol style="list-style-type: none"> 1. $X > Y$ 比較関数 GTE(X,Y,k) 2. $X < Y$ 比較関数 LTE(X,Y,k) 3. $X = Y$ 比較関数 EQU(X,Y,k) 4. $X < > Y$ 比較関数 NEQ(X,Y,k) <p>kは省略可 k=0の時、論理値を戻し、k<>0の時、成立したXのIndexを戻す。</p> <p><新規追加した関数></p> <ol style="list-style-type: none"> 1. $X > Y$ 比較関数 GT_(X,Y,k) 2. $X < Y$ 比較関数 LT_(X,Y,k)
2011年06月09日	1.05	<p><機能強化した関数></p> <ol style="list-style-type: none"> 1. MXP 関数、MNP 関数 最大値検索範囲指定を Index だけでなく、論理値数列でも可能とした 2. DAG 関数 直流数列の生成を可能とした。又、同時に引数並びを変更した。 <p><変更した関数></p> <ol style="list-style-type: none"> 1. TCP 関数 戻り値を 1000 と 0 からなる数列から "1" と "0" からなる論理値数列の変更 2. TBL 関数 引数記述方法変更 <p><Script 文字列操作文から演算関数化、新設></p> <ol style="list-style-type: none"> 1. ELM(&m) 文字列配列データ個数取得関数 ← char num_element 2. CLN(&m) 文字列配列文字数取得関数 ← char length 3. CEXT(k,n,&m) 文字列指定範囲切り出し関数 ← char extract 4. CFD(k,&n,&m) 文字列指定文字検索関数 ← char find 5. CREC(k,n,&m) 文字列配列再構成関数 ← char recomposition 6. CDEL(k,n,&m) 文字列指定範囲削除関数 ← char delete 7. CINS(k,&n,&m) 文字列挿入関数 ← char insert 8. CREP(k,&n,&s,&m) 文字列置換関数 ← char replace <p><Script 入出力文から演算関数化、新設></p> <ol style="list-style-type: none"> 1. NCH() 解析対象ファイル収録チャンネル数取得関数 ← read num_ch 2. CFNM() 解析対象ファイル名取得関数 ← read file_name 3. CHS() 解析対象ファイル収録チャンネル番号取得関数 ← read ch_series 4. CMRM(k) 解析対象ファイルマークメモ取得関数 ← read mark_memo
2011年10月15日	1.06	<p><新規追加した関数></p> <ol style="list-style-type: none"> 1. MED(X) 中央値関数 2. THO(k,X) 1/3 オクターブ解析関数 3. COTF(k,X) 1/3 オクターブ解析周波数取得関数 4. DBC(k,X,Y) dB 変換関数

		<ul style="list-style-type: none"> 5. TRX(X,Y) 積和演算関数 6. VAR(X,Y) 共分散関数 7. CRR(X,Y) 相関係数関数 8. LSM(X,Y) 最小二乗法係数演算関数 9. GMM(X) ガンマ関数 10. UNQ(X) 数列同値除去関数 11. CNV(k,m,n,X,f) セル番号/セル代表値演算関数 12. PL() π関数 13. CRR(X,Y) 相関係数演算関数 14. CEQ(&m,&n,K)文字列一致検査関数 15. CEC(k,n,&n,&m)文字列指定文字削除関数 18. ATS(k,n,m)1,2,5 刻み生成関数 19. CMP(k,X,Y)存在 Index 取得関数 20. RTI(k,m) R10 系列 Index 取得関数 21. RTF(X) R10 公称周波数変換関数 22. CTN(&m) 文字列数値変換関数 23. CXUT() サンプリング単位取得関数 24. CSDT() 収録開始年月日取得関数 25. CSTM() 収録開始時刻取得関数 26. XOF() 収録開始 X 軸オフセット取得関数 27. NDT() 収録データ個数取得関数 28. CCMT(k) 解析対象ファイルコメント取得関数 29. CFPN() カレントフォルダパス名取得関数 30. RFC(&m) ファイルステータス取得関数 31. MAC(X,Y)積和演算関数 32. CERP(k,&n,&m)文字列要素置換関数 33. FTI(X)R10 系列 Index 変換関数 34. CBSC(&m,k)文字列大文字小文字変換 35. CPTV(k,&m)文字列 Index 指定抽出関数 37. CDEC(k,n,&n,&m)文字列指定文字削除関数 38. CSEP(k,&m)文字列分離関数 39. CNCT(k,&m)文字列結合関数、 40. CNDT()現在年月日取得関数 41. CNTM()現在時刻取得関数 42. CVTA(k,X)数値 ASCII 文字列変換関数
2012年1月13日	1.07	誤字訂正
2012年3月12日	1.08	<p><説明追加></p> <ul style="list-style-type: none"> 1. RVS(k,X,Y) 戻り値選択関数 2. ITP((k,n,m,X) テーブル補間関数 <p><新規追加関数></p> <ul style="list-style-type: none"> 1. DEL(k,X) 指定 Index 数列削除関数 2. FIR(k,X) FIR フィルタ実行関数 3. VSN(n,m,X) 矢弦関数
2012年7月2日	1.09	<p><新規追加した関数></p> <ul style="list-style-type: none"> 1. CFT(k,X,m) クレストファクタ演算関数 2. WBF(X) ISO2631-1_Wf 補正フィルタ関数 3. WBJ(X) ISO2631-1_Wj 補正フィルタ関数 4. OCT(k,m,X) 1/3 オクターブバンドフィルタ関数 5. FFT(k,n,m,j,X) フーリエ変換関数 6. PCX(X) 極座標変換関数 7. UNW(X) 位相角アンラップ関数 8. PDT(k,m,X,f) PEAK 検出関数 9. CCH() カレントチャンネル番号取得関数 10. INL(X1,X2,..,X10) 飛越連結関数 <p><機能強化した関数></p> <ul style="list-style-type: none"> 1. MAX(k,X) 最大値関数 2. MIN(k,X) 最小値関数 3. EFF(k,X) 実効値関数 4. MEA(k,X) 平均値関数 5. SUM(k,X) 合計値関数 6. STD(k,X) 標準偏差関数 7. LMT(k,X,m) 著大値除去関数 8. RVS(k,X,Y) 戻り値選択関数 9. AMP(k,X,m) 数列振幅値変換関数 10. MGR(X,Y,k) 数列縫合関数 11. VRP(k,n,m,X) 数列生成関数
2012年8月31日	1.10	<p><新規追加した関数></p> <ul style="list-style-type: none"> 1. FFQ(k,m) FFT 周波数列取得関数 2. TR2(k,m,X,i,n,Y) 二次元時間率頻度解析関数 3. RFM(k,m,n,X) レインフロー法頻度解析関数 4. PVM(k,m,n,X,f) PVM(k,m,n,X,f) 5. MMM(k,m,n,X) 最大/最小法頻度解析関数 6. LCR(k,m,n,X) レベルクロス法頻度解析関数 7. FLD(X) 折り畳み関数 8. MCS(k,X) 行列の列合計演算

2012年10月13日	1.11	<新規追加した関数> 1. CMM(k,m,n,X) 閾値区間最大値/最小値 Index 検索関数 <機能強化した関数> 1. MAX(k,X) 最大値関数 2. MIN(k,X) 最小値関数 3. EFF(k,X) 実効値関数 4. MEA(k,X) 平均値関数 5. SUM(k,X) 合計値関数 6. STD(k,X) 標準偏差関数 7. MED(k,X) 中央値関数
2012年11月12日	1.12	<新規追加した関数> 1. CSTR(k,&m) 文字列繰り返し生成半数 2. CLNK(&m,&n,...) 文字列配列連結関数 3. CCON(&,&n,...) 文字列接続関数 4. CNSC(k,m,n,X) 文字列数値変換関数 5. CADJ(k,&m) 文字列文字数整合関数 <機能強化した関数> 1. CREP(k,&s,&n,&m) 文字列置換関数 2. CEXT(k,n,&m) 文字列切り出し関数
2018年10月13日	1.13	逆余弦関数戻り値誤記訂正
2018年10月31日	1.14	演算関数一覧表のWBD,WBK関数の対象軸を修正(個別の関数説明部分は修正なし)
2019年1月16日	1.15	移動積分関数フラグ説明追加
2019年7月2日	1.16	積和演算関数記述例誤記修正
2021年3月8日	1.17	移動平均関数記述例誤記修正
2022年4月7日	1.18	時間重み付け移動実効値関数 引数説明体裁修正

適用

本書(Archi_1 Script 演算関数仕様編)は、PcWaveForm プログラムの解析機能操作編に記述したチャンネル間演算機能で使用する演算関数、Archi_1 Script 言語仕様編に記述した演算式で使用する演算関数の仕様について説明します。

本書の記述内容は、下記のプログラムバージョンに対応します。

- PcWaveForm Ver.7.09 ~
- PcWaveFormFANA Ver.5.39 ~
- PcWaveFormWBV+ Ver.2.19 ~

PcWaveForm プログラムの取扱説明書構成は、下記の5部構成となっています。

- 基礎編
- 解析機能操作編
- Archi_1 Script 演算関数仕様編(本書)
- Arch_1 Script 言語仕様編
- Archi_1 Script 記述方法編

の5部構成となっています。

PcWaveForm プログラムの基本操作については基礎編を、解析機能の操作については解析機能操作編を参照下さい。

なお、Archi_1 Script 言語の仕様については Archi_1 言語仕様編を、記述方法については Archi_1 Script 記述方法編を参照下さい。

ご注意

- 本書は万全を記して作成しておりますが、万一、ご不明なことや誤りなどお気づきのことがありましたらご連絡下さい。
- 本書の実行結果から生じるお客様の損害や不利益については、それが直接的、あるいは間接的を問わず一切責任を負いかねますのでご了承下さい。
- 本書は、改良のため予告なしに変更する場合があります。
- 本書の一部または全部を無断で複製または転載することは禁止されています。
- 本書に記載された会社名や製品名は、各社の登録商標である場合がございます。

目次

1. 演算関数一覧表	1
2. 演算関数仕様	6
2. 1. 三角関数	6
2. 1. 1. 正弦関数	6
2. 1. 2. 余弦関数	6
2. 1. 3. 正接関数	6
2. 1. 4. 逆正弦関数	6
2. 1. 5. 逆余弦関数	6
2. 1. 6. 逆正接関数	7
2. 1. 7. デグリー変換関数	7
2. 1. 8. ラジアン変換関数	7
2. 1. 9. π 関数	7
2. 2. 算術関数	8
2. 2. 1. 絶対値関数	8
2. 2. 2. 累積関数	8
2. 2. 3. 変則累積関数	9
2. 2. 4. 差分関数	9
2. 2. 5. 微分関数	10
2. 2. 6. 包絡線関数	10
2. 2. 7. 波形上制限関数	11
2. 2. 8. 波形下制限関数	11
2. 2. 9. 積分関数	12
2. 2. 10. 移動積分関数	13
2. 2. 11. 逆数関数	13
2. 2. 12. 最小公倍数関数	14
2. 2. 13. 常用対数関数	14
2. 2. 14. 自然対数関数	14
2. 2. 15. 指数関数	14
2. 2. 16. 負領域化関数	15
2. 2. 17. 正領域化関数	15
2. 2. 18. 整数化関数	15
2. 2. 19. 丸め関数	16
2. 2. 20. 実効値関数	16
2. 2. 21. 時間重み付け移動実効値関数	17
2. 2. 22. 移動実効値関数	18
2. 2. 23. 符号反転関数	18
2. 2. 24. 平方根関数	19
2. 2. 25. リサンプリング関数	19

2. 2. 26. クレストファクタ演算関数	19
2. 3. 複素数取扱関数	21
2. 3. 1. フーリエ変換関数	21
2. 3. 2. 極座標変換関数	22
2. 3. 3. 位相角アンラップ関数	22
2. 3. 4. FFT 周波数列取得関数	22
2. 4. 統計関連関数	23
2. 4. 1. 相関関数	23
2. 4. 2. 移動相関値関数	24
2. 4. 3. 移動振幅頻度関数	25
2. 4. 4. 最大値関数	25
2. 4. 5. 最小値関数	26
2. 4. 6. 平均値関数	26
2. 4. 7. 中央値関数	27
2. 4. 8. 合計値関数	27
2. 4. 9. 標準偏差関数	28
2. 4. 10. 積和演算関数	28
2. 4. 11. 共分散関数	29
2. 4. 12. 相間係数関数	29
2. 4. 13. 最小二乗法係数関数	29
2. 4. 14. ガンマ関数	30
2. 4. 15. セル番号/セル代表値演算関数	30
2. 5. 頻度解析関連関数	31
2. 5. 1. 度数分布関数	31
2. 5. 2. 度数分布統計値関数	32
2. 5. 3. 二次元時間率頻度解析関数	32
2. 5. 4. レインフロー法頻度解析関数	33
2. 5. 5. 極大/極小法頻度解析関数	34
2. 5. 6. 最大/最小法頻度解析関数	35
2. 5. 7. レベルクロス法頻度解析関数	36
2. 5. 8. 折り畳み関数	36
2. 6. 解析補助関数	37
2. 6. 1. 行列の列合計演算	37
2. 7. 解析補助関数	38
2. 7. 1. 非線形補正テーブル関数	38
2. 7. 2. 1,2,5 刻み数列生成関数	38
2. 7. 3. R10 系列 Index 変換関数	39
2. 7. 4. R10 系列 Index 取得関数	39
2. 7. 5. R10 公称周波数変換関数	39
2. 8. 論理演算関数	40

2. 8. 1. 論理積関数.....	40
2. 8. 2. 否定論理積関数	40
2. 8. 3. 論理和関数.....	41
2. 8. 4. 排他的論理和関数.....	41
2. 8. 5. 否定論理関数	41
2. 8. 6. 部分否定論理関数.....	42
2. 8. 7. $X=Y$ 比較演算関数.....	42
2. 8. 8. $X<>Y$ 比較演算関数	43
2. 8. 9. $X>Y$ 比較演算関数.....	43
2. 8. 10. $X>=Y$ 比較演算関数.....	44
2. 8. 11. $X<Y$ 比較演算関数	44
2. 8. 12. $X<=Y$ 比較演算関数.....	45
2. 8. 13. 論理値化関数.....	45
2. 8. 14. Bit 論理関数	46
2. 9. パルス列取扱関数.....	47
2. 9. 1. パルス波形差分値合成関数	47
2. 9. 2. パルス計数関数	48
2. 9. 3. パルス周波数関数.....	49
2. 9. 4. パルス逡倍関数	50
2. 9. 5. 回転パルス生成関数	51
2. 9. 6. パルス間引き整形関数	51
2. 10. フィルタ関数.....	52
2. 10. 1. 4次バターワース・ハイパスフィルタ関数	52
2. 10. 2. 4次バターワース・ローパスフィルタ関数	52
2. 10. 3. 積分フィルタ.....	53
2. 10. 4. 6次チェビシェフ・ローパスフィルタ関数	53
2. 10. 5. 移動平均関数.....	54
2. 10. 6. FIR フィルタ実行関数.....	55
2. 10. 7. 矢弦関数.....	55
2. 10. 8. 1/3 オクターブバンドフィルタ関数.....	56
2. 11. 補正フィルタ関数.....	57
2. 11. 1. ISO5349-1 手腕振動知覚補正フィルタ	57
2. 11. 2. ISO2631-4:2001/EN12299:2009 W_b 振動知覚補正フィルタ.....	58
2. 11. 3. ISO2631-1 C 補正フィルタ.....	59
2. 11. 4. ISO2631-1 D 補正フィルタ.....	60
2. 11. 5. ISO2631-1 E 補正フィルタ.....	61
2. 11. 6. ISO2631-1 K 補正フィルタ.....	62
2. 11. 7. ISO2631-1 F 補正フィルタ.....	63
2. 11. 8. ISO2631-1 J 補正フィルタ.....	64

2. 11. 9. ANSI 聴感補正 A カーブ・フィルタ	65
2. 11. 10. ANSI 聴感補正 C カーブ・フィルタ	65
2. 11. ロゼッタゲージ解析関数	66
2. 12. 1. 最小主ひずみ関数	66
2. 12. 2. 最大主ひずみ関数	66
2. 12. 3. 主ひずみ方向関数	66
2. 12. 4. 最大せん断ひずみ関数	67
2. 12. 7. 最小主応力関数	67
2. 12. 8. 最大主応力関数	67
2. 12. 9. 最大せん断応力関数	68
2. 12. 10. ゲージ率補正関数	68
2. 13. 数列操作関数	69
2. 13. 1. 波形数列生成関数	69
2. 13. 2. 数列生成関数	70
2. 13. 3. Index 範囲指定データ抽出関数	70
2. 13. 4. Index 指定データ抽出関数	72
2. 13. 5. Index 飛び越しデータ抽出関数	72
2. 13. 6. 論理値指定データ抽出関数	72
2. 13. 7. Index 範囲指定データ置換関数	73
2. 13. 8. Index 指定データ置換関数	73
2. 13. 9. Index 飛び越しゼロ内挿関数	74
2. 13. 10. 振幅値変換関数	75
2. 13. 11. 著大値除去関数	77
2. 13. 12. 数列連結関数	78
2. 13. 13. 数列縫合関数	78
2. 13. 14. 数列整列関数	79
2. 13. 15. 数列反転関数	79
2. 13. 16. 数列並べ替え関数	80
2. 13. 17. データ個数関数	80
2. 13. 18. 数列指定 Index 削除関数	80
2. 13. 19. 数列同値削除関数	81
2. 13. 20. テーブル補間関数	81
2. 13. 21. 戻り値選択関数	82
2. 13. 22. 収録時間経過関数	82
2. 13. 23. 山谷検索関数	83
2. 13. 24. 飛越連結関数	83
2. 14. Index 検索関数	84
2. 14. 1. 閾値通過 Index 検索関数 1	84
2. 14. 2. 閾値通過 Index 検索関数 2	85

2. 14. 3. 閾値通過持続 Index 検索関数1	86
2. 14. 4. 閾値通過持続 Index 検索関数 2	87
2. 14. 5. 閾値通過持続 Index 検索関数 3	88
2. 14. 6. 2 信号閾値通過 Index 検索関数	89
2. 14. 7. 直近山谷 Index 検索関数.....	90
2. 14. 8. 値一致 Index 検索関数	91
2. 14. 9. データ照合関数.....	91
2. 14. 10. 論理値遷移 Index 検索関数.....	92
2. 14. 11. 最小値 Index 検索関数	92
2. 14. 12. 最大値位置 Index 検索関数.....	93
2. 14. 13. 区間最大値/最小値 Index 検索関数.....	94
2. 14. 14. 一定値区間 Index 検索関数.....	95
2. 14. 15. ゼロ交点 Index 検索関数.....	95
2. 14. 16. 存在 Index 取得関数.....	96
2. 14. 17. PEAK 検出関数	97
2. 14. 18. 閾値区間最大値/最小値 Index 検索関数	98
2. 15. 文字列取扱関数	99
2. 15. 1. 文字列範囲指定削除関数	99
2. 15. 2. 文字列切り出し関数	99
2. 15. 3. 文字列検索関数	99
2. 15. 4. 文字列挿入関数	100
2. 15. 5. 文字列配列再構成関数	100
2. 15. 6. 文字列置換関数	100
2. 15. 7. 文字列配列文字数取得関数.....	101
2. 15. 8. 文字列要素数取得関数	101
2. 15. 9. 文字列数値変換関数.....	101
2. 15. 10. 文字列大文字小文字変換関数.....	101
2. 15. 11. 文字列要素置換関数	101
2. 15. 12. 文字列比較関数.....	102
2. 15. 13. 文字列指定文字削除関数	102
2. 15. 14. 文字列 Index 指定抽出関数.....	103
2. 15. 15. 文字列分離関数.....	103
2. 15. 16. 文字列結合関数.....	104
2. 15. 17. 現在年月日取得関数	104
2. 15. 18. 現在時刻取得関数	104
2. 15. 19. 数値ASCII文字列変換関数	104
2. 16. 20. 文字列繰り返し生成関数.....	105
2. 16. 21. 文字列接続関数.....	105
2. 16. 22. 文字列配列連結関数	105

2. 16. 23. 数値文字列変換関数	105
2. 16. 24. 文字列文字数整合関数	105
2. 16. 解析情報取得関数	107
2. 16. 1. 収録ファイル名取得関数	107
2. 16. 2. 収録チャンネル信号名取得関数	107
2. 16. 3. 収録チャンネル単位名取得関数	107
2. 16. 4. 収録チャンネル数取得関数	107
2. 16. 5. 収録チャンネル番号取得関数	107
2. 16. 6. マーク個数取得関数	107
2. 16. 7. マークメモ取得関数	108
2. 16. 8. マーク番号 Index 取得関数	108
2. 16. 9. 絶対マーク番号取得関数	108
2. 16. 10. サンプリング周期取得関数	108
2. 16. 11. 解析範囲開始 Index 取得関数	108
2. 16. 12. サンプリング単位取得関数	108
2. 16. 13. 収録開始年月日取得関数	109
2. 16. 14. 収録開始時刻取得関数	109
2. 16. 15. 収録開始 X 軸オフセット取得関数	109
2. 16. 16. 収録データ数取得関数	109
2. 16. 17. 解析対象ファイルコメント取得関数	109
2. 16. 18. カレントフォルダパス名取得関数	110
2. 16. 19. ファイルステータス取得関数	110
2. 16. 20. カレントチャンネル番号取得関数	110
関数索引	111

1. 演算関数一覧表

1. 三角関数	文法	機能
正弦関数	SIN(X)	Xの単位をradと見なし正弦値を求める
余弦関数	COS(X)	Xの単位をradと見なし余弦値を求める
正接関数	TAN(X)	Xの単位をradと見なし正接値を求める
逆正弦関数	ASN(X)	Xを正弦値と見なし角度radを求める
逆余弦関数	ACS(X)	Xを余弦値と見なし角度radを求める
逆正接関数	ATN(X)	Xを正接値と見なし角度radを求める
ディグリー変換関数	DEG(X)	X(rad)をディグリーに変換する
ラジアン変換関数	RAD(X)	X(deg)をラジアンに変換する
π 関数	PI()	π を戻す

2. 算術関数	文法	機能
絶対値関数	ABS(X)	Xの絶対値を求める
累積関数	ACC(X)	Xの累積値を求める Ans(0)=X(0)、Ans(n)=Ans(n-1)+X(n)
変則累積関数	CHT(k,X,Y)	k=初期値 Ans(0)=k×X(0)+Y(0)、Ans(n)=Ans(n-1)×X(n)+Y(n)
差分関数	DIF(X)	Xの1階差分値を求める
微分関数	DIV(X)	Xの微分値を求める
包絡線関数	ENV(k,X)	k=1:絶対値、2:正領域、3:負領域、Xの包絡線を求める
波形上限制限関数	GOV(k,X)	k=閾値、Xの閾値より大きい値を全て閾値に置き換える
波形下限制限関数	GUN(k,X)	k=閾値、Xの閾値より小さい値を全て閾値に置き換える
積分関数	ING(k,X)	k=積分初期値、Xの台形積分値を求める
移動積分関数	MIG(k,X)	k=時定数、Xの移動積分を求める
逆数関数	INV(X)	Xの逆数を求める
最小公倍数関数	LGM(k,X,Y)	k=小数点以下切り捨て桁数、XとYの最小公倍数を求める
常用対数関数	LGT(X)	Xの底10の対数値を求める
自然対数関数	LOG(X)	Xの底eの対数値を求める
指数関数	EXP(X)	Xの指数値を求める
負領域化関数	MNS(x)	Xの正領域値を0にする
正領域化関数	PLS(X)	Xの負領域値を0にする
整数化関数	INT(X)	Xの小数点以下を切り捨て整数化する
丸め関数	ROD(n,m,X)	n=小数点以下桁数、m=0:四捨五入、1:切り捨て、2:切り上げ
実効値関数	EFF(k,X)	Xの実効値を求める、kは演算範囲(省略可)
時間重み付け移動実効値関数	RRT(k,X,m)	k=時定数、Xの移動実効値をm(重み打ち切り値)まで遡った移動実効値を求める
移動実効値関数	RRV(k,X,f)	k=時定数、Xの移動実効値を求める。f=0:後方積分、f<0:前方積分
符号反転関数	SGN(X)	Xの符号を反転する
平方根関数	SQR(X)	Xの平方根を求める
リサンプリング関数	RSM(k,X)	k=リサンプリング周波数、Xのデータ列をkで指定したサンプリング周波数でリサンプリングする。
クレストファクタ演算関数	CFT(k,X,m)	k=時定数、Xのクレストファクタを求める m=0 過渡領域付加なし m>0 過渡領域付加あり

3. 複素数取扱関数	文法	機能
フーリエ変換関数	FFT(k,n,m,j,X)	Xのフーリエ変換を求める
極座標変換関数	PCX(X)	Xの直交座標系複素数を極座標系複素数に変換する
位相角アンラップ関数	UNW(X)	Xの位相角をスムージングする
FFT周波数列取得関数	FFQ(k,m)	m=FFT点数としてkに従ってスペクトラムに対応する周波数を戻します。

4. 統計関連関数	文法	機能
相関関数	COR(X,Y)	XとYの相互相関関数を求める
移動相関値関数	MCO(k,X,Y)	k=データ個数、XとYの移動相互相関値を求める
移動振幅頻度関数	DOP(k,m,X)	k=振幅計数時間、m=無効振幅値、Xの振幅計数時間内の無効振幅を超える振幅数を求める
最大値関数	MAX(k,X)	Xの最大値を求める、kは演算範囲(省略可)
最小値関数	MIN(k,X)	Xの最小値を求める、kは演算範囲(省略可)
平均値関数	MEA(k,X)	Xの平均値を求める、kは演算範囲(省略可)
中央値関数	MED(X)	Xの中央値を求める。
合計値関数	SUM(k,X)	Xの合計値を求める、kは演算範囲(省略可)
標準偏差関数	STD(k,X)	Xの標準偏差を求める、kは演算範囲(省略可)
積和演算関数	MAC(X,Y)	XとYの要素毎乗算結果の合計を求める
共分散関数	VAR(X,Y)	XとYの共分散値を求める
相関係数関数	CRR(X,Y)	XとYの相関係数を求める
最小二乗法係数演算関数	LSM(X,Y)	XとYの最小二乗法で直線方程式 $y=ax+b$ の傾きaとオフセットbを求める
ガンマ関数	GMM(X)	Xのガンマ関数値を求める
セル番号/セル代表値演算関数	CNV(k,m,n,f)	k=0セル番号数列、k=1セル代表値、m=セルサイズ、n=セル数、f=0正領域、f=1正負領域

1. 演算関数一覧表

5. 頻度解析関連関数	文法	機能
度数分布関数	TRC(k,m,X)	k=セルサイズ、m=セル個数、Xの度数分布を求める
度数分布統計値関数	TRX(k,m,X,Y,a)	k=セルサイズ、m=セル個数、X=度数対象、Y=統計対象、a=統計値選択フラグから統計値を求める
二次元時間率頻度解析関数	TR2(k,m,X,j,n,Y)	k=セルサイズ 1、m=セル個数 1、X=度数対象 1、j=セルサイズ 2、n=セル個数 2、Y=度数対象 2の二次元度数分布を求める
レインフロー法頻度解析関数	RFM(k,m,n,X)	k=セルサイズ、m=セル個数、n=無効振幅、X=度数対象のレインフロー法頻度解析を行う
極大/極小法頻度解析関数	PVM(k,m,n,X,f)	k=セルサイズ、m=セル個数、n=無効振幅、X=度数対象、f=解析返値フラグの極大極小法頻度解析を行う
最大/最小法頻度解析関数	MMM(k,m,n,X)	k=セルサイズ、m=セル個数、n=無効振幅、X=度数対象の最大最少法頻度解析を行う
レベルクロス法頻度解析関数	LCR(k,m,n,X)	k=セルサイズ、m=セル個数、n=無効振幅、X=度数対象のレベルクロス法頻度解析を行う
折り畳み関数	FLD(X)	X=対象数列の折り畳み加算を行う

6. 行列取扱関数	文法	機能
行列の列合計演算	MCS(k,X)	k=列数、X=対象数列の列ごとの加算を求める

7. 解析補助関数	文法	機能
非線形補正テーブル関数	TBL(AAAA)	AAAAで指定したテーブルを参照する
1,2,5刻み数列関数	ATS(k,n,m)	k:刻み上限数、n:最小値、m:最大値 最小値と最大値間を1,2,5系列で刻んだ数列を求める
周波数R10系列Index変換関数	FTI(X)	X:周波数数列をR10系列Index値に変換する
R10系列Index取得関数	RTI(k,m)	kで示す最小周波数からmで示す最大周波数を含むR10系列Indexを求めます。
R10公称周波数変換関数	RTF(X)	R10系列Indexを公称周波数に変換します。

8. 論理演算関数	文法	機能
論理積関数	AND(X,Y)	X,Yの論理積を求める
否定論理積関数	NAD(X,Y)	X,Yの否定論理積を求める
論理和関数	OR(X,Y)	X,Yの論理和を求める
排他的論理和関数	XOR(X,Y)	X,Yの排他的論理和を求める
否定論理関数	NOT(X)	Xの論理を反転する
部分否定論理関数	ECP(k,m,X)	k=論理値、m=連続個数 Xのk論理がm個連続しない所を反転する
X=Y比較演算関数	EQU(X,Y,k)	k=0の時X=Y成立"1"、不成立"0"を戻し、k<>0の時、成立したXのIndexを戻す。
X<>Y比較演算関数	NEQ(X,Y)	k=0の時X<>Y成立"1"、不成立"0"を戻し、k<>0の時、成立したXのIndexを戻す。
X>Y比較演算関数	GT(X,Y,k)	k=0の時X>Y成立"1"、不成立"0"を戻し、k<>0の時、成立したXのIndexを戻す。
X>=Y比較演算関数	GTE(X,Y,k)	k=0の時X>=Y成立"1"、不成立"0"を戻し、k<>0の時、成立したXのIndexを戻す。
X<Y比較演算関数	LT(X,Y,k)	k=0の時X<Y成立"1"、不成立"0"を戻し、k<>0の時、成立したXのIndexを戻す。
X<=Y比較演算関数	LTE(X,Y,k)	k=0の時X<=Y成立"1"、不成立"0"を戻し、k<>0の時、成立したXのIndexを戻す。
論理値化関数	RWC(k,m,X)	k=論理"1"の閾値、m=論理"0"の閾値 Xを論理波形に変換する
Bit論理関数	BIT(k,X)	k=bit位置0~15、Xを2byte整数に変換し、bit位置の論理を求める

9. パルス列取扱関数	文法	機能
パルス差分値合成関数	ASS(X,Y)	X,Yともパルス差分値で、Xの正数に対応したYの負数を合成する。
周期計数関数	CNT(X)	Xの平均値を上昇で過ぎり再び下降で過ぎた回数を求める
F-V変換関数	FVC(k,X,m)	k=1回転当たりパルス数、Xパルス列から回転数に変換する
パルス通倍関数	PDV(k,X)	k=通倍数、Xパルス列を通倍する
回転パルス生成関数	RTP(k,X)	k=1回転当たりパルス数、X回転数列からパルスに変換する
パルス間引き整形関数	TPC(k,X)	k=パルス間引き数、Xパルス列を間引き後、50%比に整形

10. フィルタ関数	文法	機能
4次バターワース・ハイパス関数	HPF(k,X)	k=遮断周波数、XのHPF処理をする
4次バターワース・ハイパス関数	HPR(k,X)	k=遮断周波数、(時間逆方向)HPF処理をする
4次バターワース・ローパス関数	LPF(k,X)	k=遮断周波数、XのLPF処理をする
4次バターワース・ローパス関数	LPR(k,X)	k=遮断周波数、Xの(時間逆方向)LPF処理をする
積分フィルタ	LBF(k,X)	k=遮断周波数 Xに-6dB/octフィルタ処理する
6次チェビシェフ・ローパス関数	LCF(k,X)	k=遮断周波数、Xに6次チェビシェフフィルタ処理する
移動平均関数	MAV(k,X)	k=平均化個数、Xの移動平均を求める
FIRフィルタ実行関数	FIR(k,X)	k=フィルタ係数、XにFIRフィルタ処理する
矢弦関数	VSN(n,m,X)	n:前方弦長、m:後方弦長、Xの矢弦値を求める

1. 演算関数一覧表

11. 補正フィルタ関数	文法	機能
ISO5349 補正フィルタ	HTV(X)	手腕振動暴露解析 手腕 X,Y,Z 軸補正フィルタ
ISO262. 14. /EN12299B 補正フィルタ関数	WBB(k,X)	全身振動暴露解析座位仰臥位 X,Y 軸補正フィルタ
ISO2631-1 C 補正フィルタ関数	WBC(X)	全身振動暴露解析背もたれ X 軸補正フィルタ
ISO2631-1 D 補正フィルタ関数	WBD(X)	全身振動暴露解析座位仰臥位 X,Y 軸補正フィルタ
ISO2631-1 E 補正フィルタ関数	WBE(X)	全身振動暴露解析座位回転モーメント補正フィルタ
ISO2631-1 K 補正フィルタ関数	WBK(X)	全身振動暴露解析座位仰臥位 Z 軸補正フィルタ
ISO2631-1 F 補正フィルタ関数	WBF(X)	動揺病(乗り物酔い)補正フィルタ
ISO2631-1 J 補正フィルタ関数	WBJ(X)	頭部暴露解析仰臥位補正フィルタ
聴感補正 Aカーブ・フィルタ関数	WAC(X)	Xに聴感補正 A フィルタ処理をする
聴感補正 Cカーブ・フィルタ関数	WCC(V)	Xに聴感補正 C フィルタ処理をする

12. ロゼッタゲージ解析関数	文法	機能
最小主ひずみ関数	EMN(a,b,c)	ひずみ量 a,b,c から最小主ひずみを求める
最大主ひずみ関数	EMX(a,b,c)	ひずみ量 a,b,c から最大主ひずみを求める
主ひずみ方向関数	ETH(a,b,c)	ひずみ量 a,b,c から主ひずみ方向を求める
最大せん断ひずみ関数	GMX(a,b,c)	ひずみ量 a,b,c から最大せん断ひずみを求める
最小主応力関数	SMN(E,v,a,b,c)	E=縦弾性係数、v=ポアソン比、a,b,c から最小応力を求める
最大主応力関数	SMX(E,v,a,b,c)	E=縦弾性係数、v=ポアソン比、a,b,c から最大応力を求める
最大せん断応力関数	TMX(E,v,a,b,c)	E=縦弾性係数、v=ポアソン比、a,b,c から最大せん断応力を求める
ゲージ率補正関数	GRT(k,X)	k=ゲージ率、Xを補正する

13. 数列操作関数	文法	機能
波形数列生成関数	DAQ(k,a,m,f,p)	k=生成データ個数、a=波高値、m=波形種別 0:正弦波 1:三角波 2:矩形波 3:鋸波、f=周波数、p=開始位相角(deg)
数列生成関数	VRP(k,n,m,X)	k=演算フラグ、k=0 補間なし、k=1 補間あり、n:=生成データ個数、m=データ番号列、Xのデータ値を m のデータ番号位置に埋め込んだ配列を生成する
Index 範囲指定データ抽出関数	ERC(k,m,X)	k=切り出し開始 Index、m=切り出し終了 Index X から切り出す
Index 指定データ抽出関数	PTV(k,X)	k=Index 値、X の k で指定された位置のデータを求める
Index 飛び越しデータ抽出関数	SEP(k,m,X)	k=抽出開始 Index、m =飛び越し数、X から該当データを抽出する
論理値指定データ抽出関数	ZSP(X,Y)	Y のデータ列を X のデータ列値=1 の Index で再構成する
Index 範囲指定データ置換関数	REP(k,X,Y)	k=置き換え開始 Index、Y の k で指定した開始 Index から X のデータに置き換える
Index 指定データ置換関数	SBV(k,m,X)	k=Index 値、m=データ値、X のkで指定された位置に m を戻す
Index 飛び越しゼロ挿入関数	INS(k,X)	k=挿入するゼロ個数、X データ間にk個ゼロ値挿入
数列振幅値変換関数	AMP(k,X,m)	X を振幅値(m=0)または最大値最小値の絶対値(m<>0)に変換して並び変える。k=0: 抽出順、1:昇順、-1:降順
数列著大値除去関数	LMT(k,X,m)	k=振幅閾値、X から閾値を超える振幅部を除去して詰める
数列連結関数	LNK(X,Y)	X の Index 順に Y を連結します
数列縫合関数	MGR(X,Y,k)	X と Y を k に従い縫合する。k=0 Index 順、K=1 大きい数字優先、K=-1 小さい数字優先
数列整列関数	QUE(X,Y)	X=Index 並び、Y を X の値順に並び変える
数列反転関数	REV(X)	X の数列を反転する
数列並べ替え関数	SRT(k,X)	k=1:昇順、2:昇順 Index 戻り -1:降順、-2:降順 Index 戻り
データ個数関数	LEN(k,X)	X データ列のデータ個数を求める、k は演算範囲(省略可)
数列指定 Index 削除関数	DEL(k,X)	数列 X から k で指定された Index(要素)を削除する
数列同値除去関数	UNQ(X)	昇順並び X 数列から同値を 1 個だけ残す
テーブル補間関数	ITP(n,m,X)	n= 補間テーブル X 軸列データ、m=補間テーブル対応にした Y 軸データ、n、m から補間して X の値に相当した Y 値を求める
戻り値選択関数	RVS(k,X,Y)	k=フラグ、フラグ値により X または Y を戻す
収録時間経過関数	SPB(k)	k=初期値(sec)、累積サンプリング周期データ列の生成
山谷検索関数	PVF(k,m,X)	k=演算モードフラグ、m=無効振幅、X の Peak/Valley 値または Peak/Valley 位置データ番号を求める
飛越連結関数	INL(X1,X2,X3,,,X10)	数列同士を Index 順に連結した数列を生成する。

1. 演算関数一覧表

14. Index 検索関数	文法	機能
閾値通過 Index 検索関数 1 (検索条件毎単発検索)	DTD(k,m,j,n,X)	k=閾値通過方向 0:DownCross 1:UpCross、m=閾値、j=検索方向 0:Index 逆順 1:Index 順、n=検索開始 Index、X の最初の閾値通過 Index を求める
閾値通過 Index 検索関数 2 (検索条件固定繰り返し検索)	DTM(k,m,j,n,X,o)	k=閾値通過方向 0:DownCross 1:UpCross、m=閾値、j=検索方向 0:Index 逆順 1:Index 順、n=検索開始 Index、X の閾値通過位置 Index を求める
閾値通過持続 Index 検索関数 1 (検索条件毎単発検索)	CTD(k,m,j,s,n,d,X)	k=閾値通過方向 0:DownCross 1:UpCross、m=閾値、n=検索方向 0:Index 逆順 1:Index 順、n=検索開始 Index、s=持続時間、d=持続時間内許容変動値、X の s から j 方向に m 閾値を k 方向に過ぎてから d 時間内の値が m±d 以内の時、閾値通過位置 Index を求める
閾値通過持続 Index 検索関数 2 (持続回数固定繰り返し検索)	DTE(k,m,n,X)	k=閾値通過方向 0:DownCross 1:UpCross、m=閾値、n=持続データ回数、X の閾値通過後、n データ数通過が保持されている場合の閾値通過位置 Index を求める
閾値通過持続 Index 検索関数 3 (持続時間固定繰り返し検索)	DTF(j,k,m,n,X)	j=検索開始 Index、k=閾値通過方向 0:DownCross 1:UpCross、m=閾値、n=持続時間、X の検索開始 Index 以降で閾値を通過し、n 時間通過が保持されている場合の閾値通過位置 Index を求める
2 信号閾値通過 Index 検索関数	DTC(j,k,m,X,n,o,p,Y)	j=検索開始 Index、k=開始点変化量閾値、m=次回開始点変化量閾値、n=遅延データ回数、o=終了点閾値変化量、p=終了点検索方向 0:Index 順 1:Index 逆順、X の j 地点から開始点変化量閾値通過位置 Index を、Y から開始点閾値通過後遅延データ回数後の終了点閾値通過位置 Index を求める
直近山谷 Index 検索関数	DTP(k,m,X)	k=検索方向&山谷種別、0:Index 逆順谷 1:Index 逆順山 2:Index 順谷 3:Index 順山、m=検索開始 Index、X の m 地点直近山谷位置 Index を求める
値一致データ番号関数	EQP(k,m,X)	k=演算モードフラグ、k=0 一致したデータ番号、k=1 一致したデータ回数、m=演算データ列、X のデータに m で指定したデータを含んでいるかを求める
データ照合関数	FND(k,X,Y)	k=0 一致論理、k=1 一致 Index
論理遷移 Index 検索関数	LST(k,X)	k=0 の時、1⇒0 に遷移したデータ番号、k=1 の時、0⇒1 に遷移したデータ番号
最小値 Index 検索関数	MNP(X)	X の最小値位置 Index を求める
最大値 Index 検索関数	MPX(X)	X の最大値位置 Index を求める
区間最大値/最小値 Index 関数	PMP(j,k,m,X)	j=最大値/最小値 0:最小値、1:最大値、k=検索閾値、m=有効閾値 閾値 k を Up Cross して再び Down Cross する区間の最大値または最小値で有効閾値を越えた位置 Index を求める
一定値区間 Index 検索関数	SWP(m,n,k,X)	m= 窓時間幅、n=窓 Y 軸幅、k=窓 Y 軸 Offset 値、X から設定した窓範囲にデータが入った時の Index、窓範囲からデータが出た時の Index を求める
ゼロ交点 Index 検索関数	ZIS(k,j,X)	k=解析中心 Index、j=解析時間幅、X の k 位置の j で指定した土幅の最大傾きとゼロ軸との交点 Index を求める
存在 Index 取得関数	CMP(k,X,Y)	k=検索条件フラグに従い検索数列 X が検索対象 Y に含まれる Index を検索する。
閾値区間最大値/最小値 Index 検索関数	CMM(k,m,n,X)	k+m,k-n の不感幅を超えるピークの Index を検索対象数列 X から検索する。

15. 文字列取扱関数	文法	機能
文字列指定範囲削除関数	CDEL(k,n,&m)	k で指定した文字位置から n 文字数を &m 文字列から削除する← char delete と等価
文字列指定範囲切り出し関数	CEXT(k,n,&m)	k 文字位置から n 文字数を &m 文字列から切り出す← char extract と等価
文字列指定範囲検索関数	CFND(k,&n,&m)	k 文字位置から &n 文字列が &m 文字列に含まれているか検索する← char find と等価
文字列挿入関数	CINS(k,&n,&m)	k で指定した文字位置から &n 文字列を &m 文字列に挿入する← char insert と等価
文字列配列再構成関数	CREC(k,n,&m)	k はフラグ、k=0 の時、n は論理数列、論理数列 "1" に対応する &m 文字列 Index で再構成 k=1 の時、n は Index 数列、n に従って &m 文字列並び替え← char recombination と等価
文字列置換関数	CREP(k,&s,&n,&m)	k で指定した文字位置から &m 文字列に &s 文字列に一致した箇所を &n 文字列に置換する← char replace と等価
文字列配列文字数取得関数	CLN(&m)	&m の文字数を取得する← char length と等価
文字列配列要素数取得関数	ELM(&m)	&m 文字列の要素数を取得する← char num_element と等価
文字列数値変換関数	CTN(&m)	&m を数値に変換します。← assign \$n = &m と等価
文字列大文字小文字変換関数	CBSC(&m,k)	k=0 の時、&m に含まれる大文字を小文字に、k=1 の時、&m に含まれる小文字を大文字に変換
文字列要素置換関数	CERP(k,&n,&m)	k=置換する Index、&n 置換文字列、&m 置換対象文字列
文字列一致検査関数	CEQ(&m,&n,k)	&m と &n を比較して k=0 の時論理値を戻し、k=1 の時一致した &n の Index を戻す。ン
文字列指定文字削除関数	CEC(k,n,&n,&m)	k:検索開始文字位置、&n: 検索文字列、&m: 対象文字列、n=0 の時、異なる孤児列まで、n=1 の時、一致した全ての文字列を削除
文字列 Index 指定抽出関数	CPTV(k,&m)	k で指定した &m の Index の文字列を戻す。
文字列分離関数	CSEP(k,&m)	k:セパレータコード k=0改行、1 半角カンマ、2tab、3 半角スペースで文字列を要素に分離する
文字列結合関数	CNCT(k,&m)	k:セパレータコード k=0 改行、1 半角カンマ、2tab、3 半角スペースで文字列要素を結合する
現在年月日取得関数	CNDT()	現在年月日を取得します。
現在時刻取得関数	CNTM()	現在時刻を取得します。
数値 ASCII 文字列変換関数	CVTA(k,X,f)	k=0 の時 ascii 文字列へ変換します。
文字列繰り返し生成関数	CSTR(k,&m)	&m を k で示す繰り返し回数分接続した文字列を生成します。
文字列接続関数	CCON(&m,&n,&k,,)	文字列を接続します。
文字列配列連結関数	CLNK(&m,&n,&k,,)	文字列配列を連結します。
数値文字列変換関数	CNSC(k,m,n,X)	k 型式、m 全桁、n 小数点以下桁で X を文字列に変換します。
文字列文字数整合関数	CADJ(k,&m)	&m を k で示す文字数に整合します。

16. 解析情報取得関数	文法	機能
解析対象ファイル名取得関数	CFNM()	解析対象ファイルのファイル名を取得します。← read file_name と等価
収録チャンネル信号名取得関数	CHNM(k)	解析対象ファイルの信号名を取得します。← read ch_name と等価
収録チャンネル単位名取得関数	CUNT(k)	解析対象ファイルの単位名を取得します。
収録チャンネル数取得関数	NCH()	解析対象ファイルの収録チャンネル数を取得します。← read num_ch と等価
収録チャンネル番号取得関数	CHS()	解析対象ファイルの収録チャンネル番号を取得します。← read ch_series と等価
マーク個数取得関数	NMK()	解析範囲の付けられているマーク個数を取得します。
マークメモ取得関数	CMRM(k)	解析範囲の k で示したマーク番号メモを取得します。← read mark_memo と等価
マーク番号 Index 取得関数	MRK(k)	k=マーク番号、マーク番号位置の Index を取得します。
絶対マーク番号取得関数	MKS()	解析範囲に存在する先頭マーク絶対番号を取得します、
サンプリング周期関数	PRD()	解析対象ファイルのサンプリング周期を取得
解析範囲開始 Index 取得関数	STAC()	解析対象ファイルから解析対象開始絶対 Index 値の取得
サンプリング単位取得関数	CXUT()	収録サンプリング周期の単位を取得します。
収録開始年月日取得関数	CSDT()	収録開始年月日を取得します。
収録開始時刻取得関数	CSTM()	収録開始時刻を取得します。
収録開始 X 軸オフセット取得関数	XOF()	収録開始 X 軸オフセット値を取得します。
収録データ個数取得関数	NDT()	収録データ個数を取得します。
解析対象ファイルコメント取得関数	CCMT(k)	ヘッダーファイルコメント行からコメント文を取得します。
カレントフォルダパス名取得関数	CFPN()	カレントに設定されているフォルダパス名を取得します。
ファイルステータス取得関数	RFC(&m)	設定したファイルのステータスを取得します。
カレントチャンネル取得関数	CCH()	解析範囲に指定されたカレントチャンネル番号を習得します

2. 演算関数仕様

2. 1. 三角関数

2. 1. 1. 正弦関数

機能	X をラジアン単位と見なし、その正弦値を戻します。戻り値は±1 の範囲となります。
文法	SIN(X)
引数	【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	SIN(#1) SIN(RAD(#1-\$1))
備考	$Ans_n = \sin X_n$

2. 1. 2. 余弦関数

機能	X をラジアン単位と見なし、その余弦値を戻します。戻り値は±1 の範囲となります。
文法	COS(X)
引数	【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	COS(#2)
備考	$Ans_n = \cos X_n$

2. 1. 3. 正接関数

機能	X をラジアン単位と見なし、その正接値を戻します。
文法	TAN(X)
引数	【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X は $-\pi/2 < X < \pi/2$ の範囲でなくてはなりません。
記述例	TAN(\$2)
備考	$Ans_n = \tan X_n$ なお、X の値が $-\pi/2$ 、および $\pi/2$ の時は倍精度で取り得る最大値 22273405434 を返します。 ただし、演算結果を格納する場合、データ形式を INTEGER(INT16) に設定すると、最大値を 25,000 とした SLOPE と OFFSET 値を求め、データ範囲を縮退しますので、当該チャンネルの最小値が正しく求まりませんので注意が必要です。

2. 1. 4. 逆正弦関数

機能	X の逆正弦値を戻します。戻り値は $-\pi/2 \sim \pi/2$ の範囲となります。
文法	ASN(X)
引数	【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X は $-1 < X < 1$ の範囲でなくてはなりません。
記述例	ASN(#3)
備考	$Ans_n = \sin^{-1} X_n$

2. 1. 5. 逆余弦関数

機能	X の逆余弦値を戻します。戻り値は $0 \sim \pi$ の範囲となります。
文法	ACS(X)
引数	【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X は $-1 < X < 1$ の範囲でなくてはなりません。
記述例	ACS(\$3)
備考	$Ans_n = \cos^{-1} X_n$

2.1.6. 逆正接関数

機能	Xの逆正接値を戻します。戻り値は $-\pi/2 \sim \pi/2$ の範囲となります。
文法	ATN(X)
引数	【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	ATN(\$1/\$3)
備考	$Ans_n = \tan^{-1} X_n$

2.1.7. デグリー変換関数

機能	Xをラジアン単位と見なしXに $180/\pi$ を掛け、デグリー単位に変換した値を戻します。
文法	DEG(X)
引数	【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	DEG(\$1)
備考	$Ans_n = X_n \times \frac{180}{\pi}$

2.1.8. ラジアン変換関数

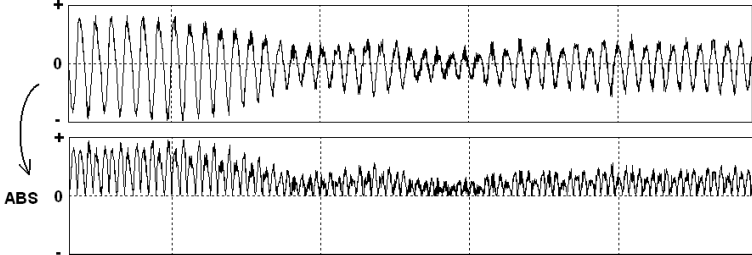
機能	Xをデグリー単位と見なしXに $\pi/180$ を掛け、ラジアン単位に変換した結果を戻します。演算で π の値を使用したい場合は本関数使用し、引数に180を記述します。
文法	RAD(X)
引数	【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	RAD(#2) RAD(180) ← 演算式で π を使用したい時
備考	$Ans_n = X_n \times \frac{\pi}{180}$

2.1.9. π 関数

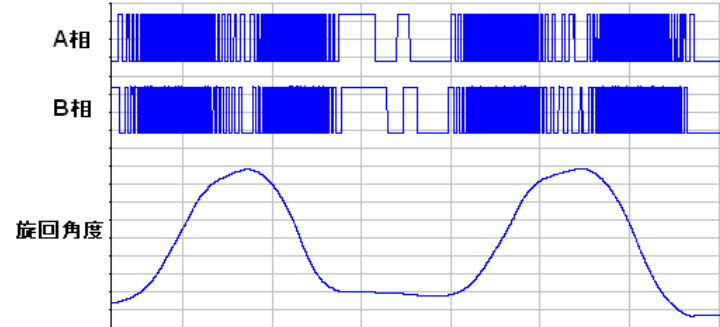
機能	定数 π (pi:円周率)を戻します。
文法	PI()
引数	なし
記述例	PI()
備考	ラジアン変換関数の引数に180rとして記述した事と等価です。RAD(180) = PI()

2. 2. 算術関数

2. 2. 1. 絶対値関数

機能	X の絶対値を戻します。
文法	ABS(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	ABS(\$2)
備考	<p>\$1 ← 1.2,-2,-4,2.2 \$1 = ABS(\$1) \$1 → 1.2,2,4,2.2</p>  <p>演算対象データの要素数が1個の場合は戻り値も要素数 1 個となり、演算対象データが複数要素の場合は戻り値も複数要素となります。</p>

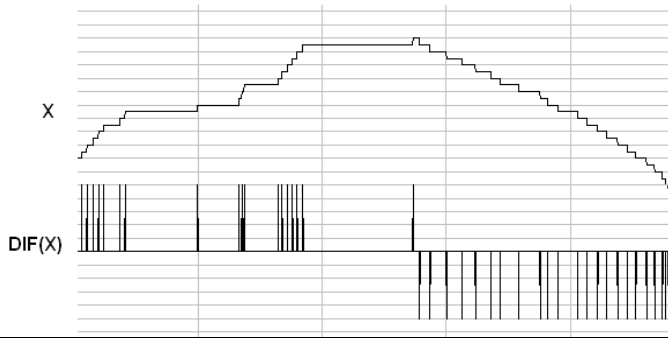
2. 2. 2. 累積関数

機能	X の累積値を戻します。
文法	ACC(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	ACC(\$1)
備考	$Ans_n = \sum_{i=1}^{n-1} X_i + X_n$ <p>\$1 ← 1,1,1,1,1 \$1 = ACC(\$1)-1 \$1 → 0,1,2,3,4</p> <p>ACC 関数の適用例: 2 相パルス列から旋回角度を求める。 A 相:=\$12、B 相:=\$13、1 回転当たりパルス数:=\$15。パルス閾値:=\$14 演算式: ACC(DIF(GTE(\$12,\$14))*(GTE(\$13,\$14)-0.5)*2+DIF(GTE(#13,\$14))*(LTE(\$12,\$14)-0.5)*2)*360/\$15</p> 

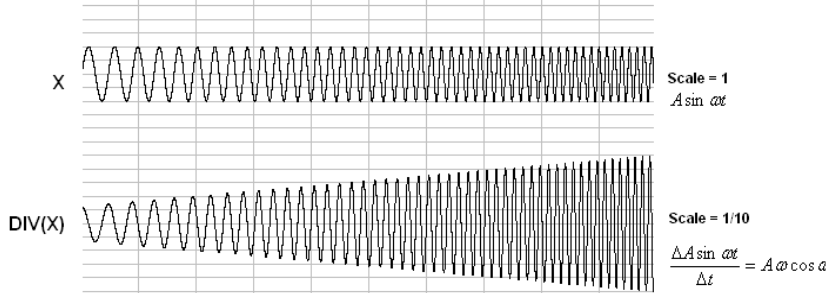
2. 2. 3. 変則累積関数

機能	回帰型累積値を戻します。
文法	CHT(k,X,Y)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> 係数初期値を記述します。</p> <p>【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p> <p>【Y】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 加算データ数列をを記述します。</p> <p>演算結果のデータ個数は、Xの個数とYの個数に依存します。 Xのデータ個数>1 & Yのデータ個数>1の場合 演算結果のデータ個数は、X,Yの何れか少ない方のデータ個数となります。 {Xのデータ個数=1 & Yのデータ個数>1} OR {Xのデータ個数>1 & Yのデータ個数=1}の場合 演算結果のデータ個数は、X,Y何れかデータ個数が>1でない方のデータ個数となります。 なお、データ個数1個の配列は全て同じ値として参照されます</p>
記述例	CHT(80.EXP(SGN(\$1)),\$2)
備考	$A_0 = k \times X_0 + Y_0$ $A_1 = A_0 \times X_1 + Y_1$ $A_2 = A_1 \times X_2 + Y_2$ $A_n = A_{n-1} \times X_n + Y_n$

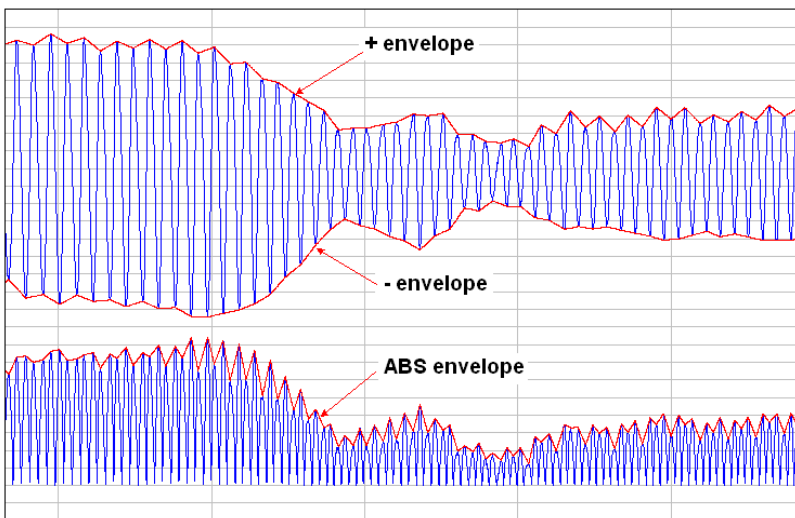
2. 2. 4. 差分関数

機能	Xの一階差分値を戻します。
文法	DIF(X)
引数	<p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、Xのデータ個数は>=2でなければなりません。</p>
記述例	DIF(#1)
備考	<p>演算結果データ個数はXのデータ個数-1となります。</p> $Ans_n = X_{n+1} - X_n$ 

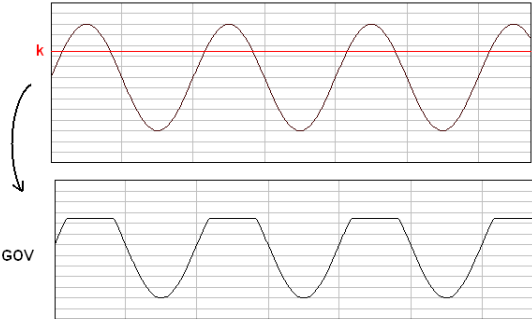
2. 2. 5. 微分関数

機能	X の微分値を戻します。
文法	DIV(X)
引数	[X]<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X のデータ個数は>=2 でなければなりません。
記述例	DIV(#1)
備考	演算結果データ個数は X のデータ個数-1となります。 $Ans_n = \frac{X_{n+1} - X_n}{\Delta t}$ 

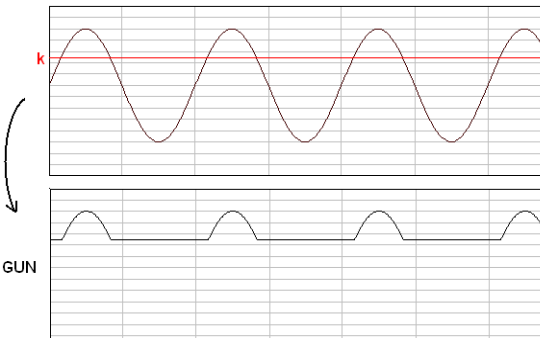
2. 2. 6. 包絡線関数

機能	ゼロを過る最大値、または最小値を接続した包絡線を戻します。																					
文法	ENV(k,X)																					
引数	[k]<必須><即値/数値属性参照チャンネル> 演算モードフラグを記述します。記述する演算モードフラグを下表に示します。																					
	<table border="1"> <thead> <tr> <th>K</th> <th>抽出方法</th> <th>補間方法</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>絶対値 peak</td> <td>直線</td> </tr> <tr> <td>2</td> <td>+側 peak</td> <td>直線</td> </tr> <tr> <td>3</td> <td>-側 peak</td> <td>直線</td> </tr> <tr> <td>-1</td> <td>絶対値 peak</td> <td>スプライン</td> </tr> <tr> <td>-2</td> <td>+側 peak</td> <td>スプライン</td> </tr> <tr> <td>-3</td> <td>-側 peak</td> <td>スプライン</td> </tr> </tbody> </table>	K	抽出方法	補間方法	1	絶対値 peak	直線	2	+側 peak	直線	3	-側 peak	直線	-1	絶対値 peak	スプライン	-2	+側 peak	スプライン	-3	-側 peak	スプライン
K	抽出方法	補間方法																				
1	絶対値 peak	直線																				
2	+側 peak	直線																				
3	-側 peak	直線																				
-1	絶対値 peak	スプライン																				
-2	+側 peak	スプライン																				
-3	-側 peak	スプライン																				
	[X]<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X は peak が検出可能なデータ(ゼロを過ぎている)で、最低 2 個の peak が存在する必要があります。																					
記述例	ENV(1,#1)																					
備考	演算対象データ X から設定した検出方法により peak を検出して、指定した補間方法で peak 間を補間した包絡線を戻します。 解析例																					
																						

2.2.7. 波形上制限関数

機能	Xの波形に設定した上限値を上回ったデータを上限値に置き換えます。
文法	GOV(k,X)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> 上限値を記述します。 【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。
記述例	GOV(10,\$1)
備考	演算対象データを上限値kと比較し、上限値を越えているデータを閾値に置き換えます。 

2.2.8. 波形下制限関数

機能	Xの波形に設定した下限値を下回ったデータを下限値に置き換えます。
文法	GUN(k,X)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> 下限値を記述します。 【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。
記述例	GUN(2,\$1)
備考	演算対象データを下限値kと比較し、下限値に満たないデータを下限値に置き換えます。 

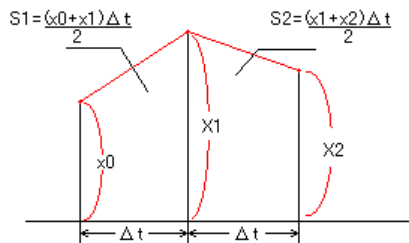
2.2.9. 積分関数

機能	X を初期値 k として台形公式で積分した結果を戻します。
文法	ING(k,X)
引数	【k】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 積分初期値を記述します。
	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X のデータ個数は>=2 でなければなりません。
記述例	ING(0,#1) → 積分初期値を 0 とした ch1 の積分結果を戻します。 ING(0,#1-MEA(#1)) → 積分初期値を 0 とした Ch1 の平均値を除去した結果の積分結果を戻します。
備考	演算結果データ個数は、X のデータ個数-1 となります。

$$S_0 = k$$

$$S_1 = S_0 + \left(\frac{(x_0 + x_1)\Delta t}{2} \right)$$

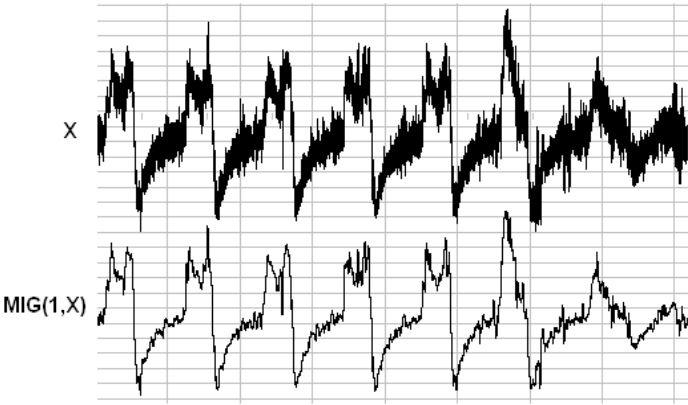
$$S_n = \sum_{i=0}^{n-1} S_i + \left(\frac{(x_{n-1} + x_n)\Delta t}{2} \right)$$



積分対象データ列に直流分が含まれていると、右肩上がりの結果(直流分が正の時)となります。
 演算例-上段は車両速度データ、下段は演算で求めた走行距離を示します。



2. 2. 10. 移動積分関数

機能	X を指定した積分周期での移動積分値を戻します。
文法	MIG(k,X,m)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> 積分周期(単位 sec)を記述します。記述する k が複数の要素で構成されていても参照される値は Index0 の値のみとなります。 なお、K は、0 より大きくなくてはなりません。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X のデータ個数は、積分周期 k より長くなくてはなりません。</p> <p>【m】<省略可><即値/数値属性参照チャンネル/演算式> 過渡領域付加有無フラグを記述します。 過渡領域付加フラグ:=0 の時:(記述省略した場合は、フラグ 0 と見なします) 過渡領域付加しないを意味し、後方積分となりますので全体のデータ長は積分周期分短くなります。 フラグ:<>0 の時: 過渡領域付加を意味し、前方積分となりますので、現在のデータ長は演算対象データ X と同じ長さですが、先頭から設定した積分 周期 τ に達するまでが過渡領域となります。過渡領域の計算結果は積分周期に満たないデータ数で計算された値となります。</p>
記述例	MIG(\$1,#1)
備考	<p>演算結果のデータ個数は設定した積分周期相当分短くなります。又、積分は台形則を使用します。</p> $n = \frac{k}{\Delta t}$ <p>k は設定した積分周期、Δt はサンプリング周期を意味します。</p> $Ans_i = \int_{m=i}^{i+\tau} X_m dt = \sum_{m=i}^{i+n} \frac{X_m + X_{m+1}}{2} \times \Delta t$  <p>移動積分処理結果を周波数軸上で見ると、移動平均関数と同様 LowPassFilter の特性を持ちます。</p>

2. 2. 11. 逆数関数

機能	X の逆数を戻します。
文法	INV(X)
引数	<p>【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X は $X \neq 0$ でなくてはなりません</p>
記述例	INV(#1) → ch1 の逆数を戻します。INV(#1)と 1/#1 は同じ結果が得られます。
備考	$Ans_n = \frac{1}{X_n}$

2. 2. 12. 最小公倍数関数

機能	X,Y の最小公倍数を戻します。
文法	LCM(k,X,Y)
引数	<p>【k】<必須><即値/数値属性参照チャネル> 小数点以下有効桁数を記述します。 k=0 は X,Y で記述された値の小数点以下切り捨て整数化して使用されます。 k=n は X,Y で記述された小数点以下 n 桁目以降が切り捨てられ使用されます。</p> <p>【X】<必須><即値/数値属性参照チャネル/演算式> 最小公倍数を演算する値を記述します。なお、複数の要素を持つ配列型であっても Index0 のみ参照されます。</p> <p>【Y】<必須><即値/数値属性参照チャネル/演算式> 最小公倍数を演算する値を記述します。なお、複数の要素を持つ配列型であっても Index0 のみ参照されます。</p>
記述例	LCM(0,\$1,1024)
備考	設定される X および Y は演算過程で整数化されて使用します。これは、小数点以下の値が切り捨てられるという意味ではなく、例えば、小数点以下 3 桁と指定された場合、×1000 され整数化するという意味です。ただし、整数化された値の最大値は、32bit 整数で扱える最大値 4293967296 までの範囲にないと最小公倍数は正しく求まりません。

2. 2. 13. 常用対数関数

機能	X を底 10 とした常用対数値を戻します。ただし、X<=0 の場合は実行時エラーとせず 0 を戻します。
文法	LGT(X)
引数	<p>【X】<必須><即値/収録チャネル/数値属性参照チャネル/演算式> 演算対象数値を記述します。なお、X>0 の必要があります。</p>
記述例	<p>LGT(\$3*#2) 20*LGT(RRV(1,WAC(#1))/20e-6) →#1 音圧と見なし聴感補正後の音圧レベル(SPL)に変換します。</p>
備考	$Ans_n = \log_{10} X_n = \frac{\log_e X_n}{\log_e 10}$

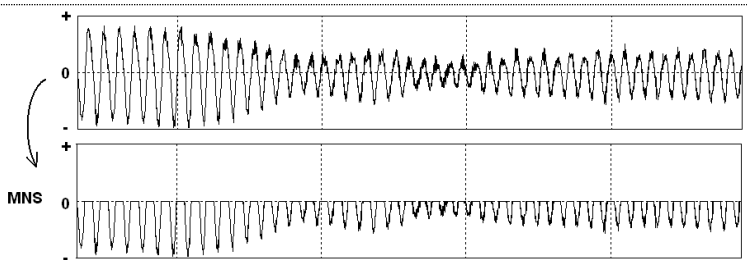
2. 2. 14. 自然対数関数

機能	X を底 e とした自然対数値を戻します。ただし、X<=0 の場合は実行時エラーとせず 0 を戻します。
文法	LOG(X)
引数	<p>【X】<必須><即値/収録チャネル/数値属性参照チャネル/演算式> 演算対象数値を記述します。なお、X>0 の必要があります。</p>
記述例	LOG(#1)
備考	$Ans_n = \log_e X_n$

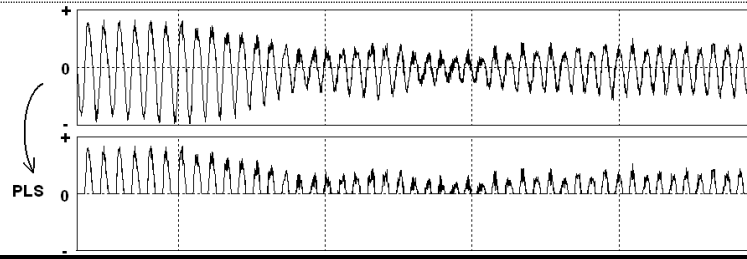
2. 2. 15. 指数関数

機能	X の指数値を戻します。
文法	EXP(X)
引数	<p>【X】<必須><即値/収録チャネル/数値属性参照チャネル/演算式> 演算対象数値を記述します。</p>
記述例	EXP(\$1)
備考	$Ans_n = e^{X_n}$

2. 2. 16. 負領域化関数

機能	X の正領域のデータをすべて 0 に置き換えます。
文法	MNS(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	MNS(#30)
備考	 <p>演算対象データの要素数が1個の場合は戻り値も要素数 1 個となり、演算対象データが複数要素の場合は戻り値も複数要素となります。</p>

2. 2. 17. 正領域化関数

機能	X の負領域のデータをすべて 0 に置き換えます。
文法	PLS(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	PLS(#2)
備考	

2. 2. 18. 整数化関数

機能	X の小数点以下を切り捨て、整数化した値を戻します。
文法	INT(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	INT(\$2)
備考	検算結果を波形データとして格納する場合、格納形式に INTEGER を選択すると格納時のデータ縮退による演算誤差により崩れる場合があります。

2. 2. 19. 丸め関数

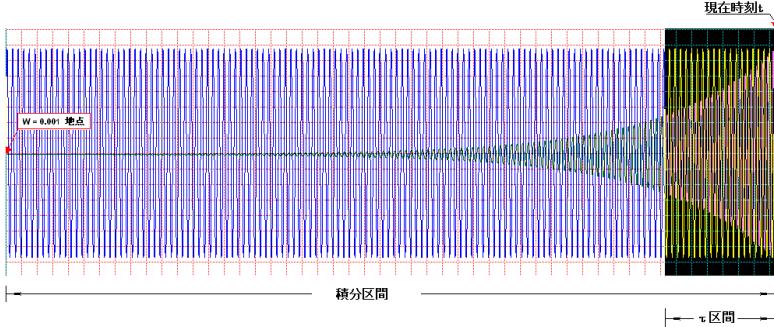
機能	X をで指定した桁数位置に丸めます。																									
文法	ROD(n,m,X)																									
引数	<p>【n】<必須><即値/数値属性参照チャネル> 有効桁位置を記述します。有効桁位置とは、小数点位置を中心に正の整数が小数点以下桁方向、負の整数が小数点以上桁方向を示します。ただし、0 は 1 と同じ意味を持ちます。指定された桁値が整数で無い場合、整数化されて使用されます。丸める桁は、ここで指定された有効桁位置の一桁下となります。</p> <p style="text-align: center;">n <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>-3</td><td>-2</td><td>-1</td><td>.</td><td>1</td><td>2</td><td>3</td></tr></table></p> <p>なお、n が複数の要素を持つ配列で記述されても Index0 以外は参照されません。</p> <p>【m】<必須><即値/参照チャネル> 丸め方式フラグを記述します。記述する丸め方式フラグを可表に示します。</p> <table border="1" style="margin-left: 20px;"> <tr><th>m 値</th><th>丸め方式</th></tr> <tr><td>0</td><td>四捨五入</td></tr> <tr><td>1</td><td>切り捨て</td></tr> <tr><td>2</td><td>切り上げ</td></tr> </table> <p>なお、m が複数の要素を持つ配列で記述されても Index0 以外は参照されません。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>	-3	-2	-1	.	1	2	3	m 値	丸め方式	0	四捨五入	1	切り捨て	2	切り上げ										
-3	-2	-1	.	1	2	3																				
m 値	丸め方式																									
0	四捨五入																									
1	切り捨て																									
2	切り上げ																									
記述例	ROD(3,0,#2)																									
備考	<p>設定桁数 m と丸め方式 n の例を示します。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>X 値</th> <th>n</th> <th>m=0</th> <th>m=1</th> <th>m=2</th> </tr> </thead> <tbody> <tr><td>123.456</td><td>2</td><td>123.460</td><td>123.450</td><td>123.460</td></tr> <tr><td>123.456</td><td>1</td><td>123.500</td><td>123.400</td><td>123.500</td></tr> <tr><td>123.456</td><td>-1or0</td><td>123.000</td><td>123.000</td><td>124.000</td></tr> <tr><td>123.456</td><td>-2</td><td>120.000</td><td>120.000</td><td>130.000</td></tr> </tbody> </table> <p>検算結果を波形データとして格納する場合、格納形式に INTEGER を選択すると格納時のデータ縮退による演算誤差により崩れる場合があります。</p>	X 値	n	m=0	m=1	m=2	123.456	2	123.460	123.450	123.460	123.456	1	123.500	123.400	123.500	123.456	-1or0	123.000	123.000	124.000	123.456	-2	120.000	120.000	130.000
X 値	n	m=0	m=1	m=2																						
123.456	2	123.460	123.450	123.460																						
123.456	1	123.500	123.400	123.500																						
123.456	-1or0	123.000	123.000	124.000																						
123.456	-2	120.000	120.000	130.000																						

2. 2. 20. 実効値関数

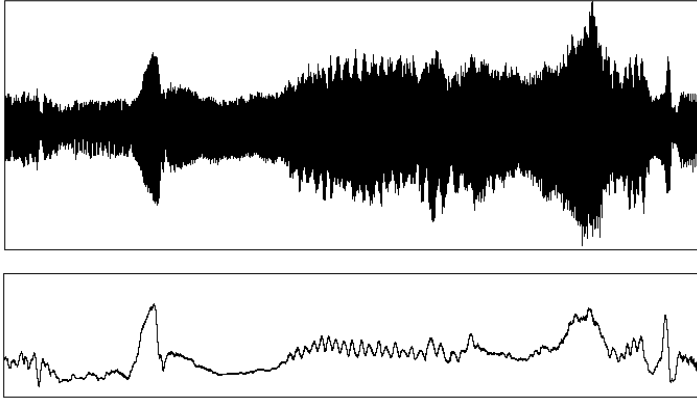
機能	X の実効値を戻します。k が記述された場合は、k で記述した区間毎の実効値を戻します。
文法	EFF(k,X)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式> 演算範囲を Index で記述します。 即値または要素 1 個の参照チャネルで記述した場合 Index0 から k 個毎を演算範囲とします。最終区間が設定したデータ個数に満たない場合は、該当区間の演算を行いません。 設定したデータ個数に演算対象数列の個数が満たない場合は実行時エラーとなります。 複数要素の参照チャネルで記述した場合 k(n)を X の Index 値として k(n)から k(n+1)-1 までの Index を演算範囲とします。複数要素の場合 k は昇順並びの必要があり、k で指定した Index が存在しない場合は実行時エラーとなります。 複数要素の参照チャネルで記述し要素数が演算対象数列と同じ場合 記述された k を"1"と"0"で構成された論理数列と見なし、論理"1"の区間を演算範囲とします。論理数列が全て"0"の場合は実行時エラーとなります。なお、全て"1"の場合は記述省略された時と同じ結果となります。また、記述された k が論理数列でない場合、結果は保証されません。 記述省略した場合は、演算対象数列 X の全体を演算対象とします。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>
記述例	EFF(#1) EFF(\$1,#1)
備考	<p>$Ans = \sqrt{\frac{1}{n} \int_1^n X^2} = \sqrt{\frac{1}{n} \sum_1^n X^2}$</p> <p>戻り個数は下記の様になります。</p> <p>引数 k が省略された場合: データ戻り個数 1 個 引数 k が即値または要素数 1 の場合: 戻りデータ個数=INT(LEN(X)/k) 例えば X のデータ数が 1000、k=300 の場合は戻り個数は 3 個となります。 引数 k が複数要素の場合: データ個数=LEN(k)-1 例えば k=100,200,300 の場合、Index100 から 199 までの演算値、Index200 から 299 までの演算値の 2 個が戻ります。 引数 k が 0/1 で記述されていた場合 例えば k=1,1,1,0,0,1,1,1,1,0,1,1、X の要素数が 12 の場合、k が 1 で記述されている 0~2、5~8、10~11 の演算結果の 3</p>

個が戻ります。

2. 2. 21. 時間重み付け移動実効値関数

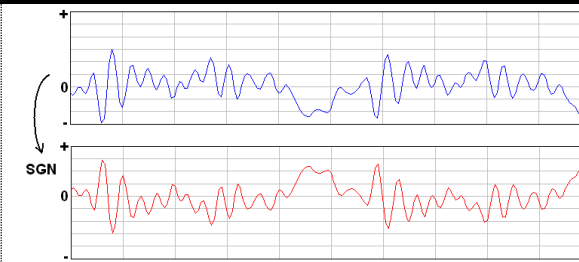
機能	X を k で指定した実効値時定数で、現時刻から時間重み付け打ち切り値まで遡った積分区間の移動実効値を戻します。
文法	RRT(k,X,m)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 積分周期 τ (単位 sec) を記述します。なお、記述した k が複数の要素で構成されていても参照される値は Index0 の値のみとなります。また、k は、0 < k でなくてはなりません。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。なお、X のデータ個数 × サンプル周期は、積分周期 k より長くなくてはなりません。</p> <p>【m】<省略可><即値/数値属性参照チャネル/演算式> 演算打ち切り重み値を記述します。 時間重み付け移動実効値演算は現時刻より前方を指数関数的に減少させ実効値演算を行います。演算式での $-\infty$ は理論的概念を示すもので関数内では打ち切りを行います。現在地点より遡ったある地点の重み値を W、与えられた実効値時定数を τ、現時刻を t、遡る時間を ξ とすると、$W = \exp^{-(t-\xi)/\tau}$ で表せます。ここで設定する m は W を意味し重み値の打ち切りとなります。積分範囲はここで設定した重み打ち切り値に達するまで遡って積分範囲として演算します。記述する値は、正数かつ 1 以下である必要があります。負数或いは 1 以上の場合は実行時 Error (パラメタ不正) となります。なお、記述省略した場合は 0.001 をとります。 現時刻 t を 0 とすると、$\xi = \tau \times \text{Ln}(W)$ で表せ、0.001 で打ち切る場合は、 $\xi = \tau \times \text{Ln}(0.001) = \tau \times (-6.907755279)$ となります。例えば、実効値時定数を 1s とした場合は、-6.907755279 秒遡ることを意味します。実際には、データ個数に変換する為、遡り戻りデータ個数を求める必要があり、サンプル周期 Δt で割り整数化した値を遡り個数とします。</p>
記述例	RRT(0.125,#1) → Ch1 を実効値時定数 125ms で打ち切り重み値 0.001(863.5ms を積分範囲)とした移動実効値を戻します。
備考	$Ans_t = \left[\frac{1}{\tau} \int_{\xi=-\infty}^t x_{\xi}^2 \cdot e^{-(t-\xi)/\tau} d\xi \right]^{\frac{1}{2}}$ <p>X: 演算対象データ列、τ: 実効値時定数、$-\infty$: 重み付け打ち切り値、t: 現時刻、ξ: 積分変数 演算対象数列の先頭から積分範囲に達するまでは過渡領域となります。打ち切り重み値まで遡れない場合は遡れるまでを積分区間と見なし、時定数まで遡れない領域では更に遡れる範囲を時定数と見なします。 下図に一定振幅正弦波を打ち切り重み 0.001 で、時間重み付け移動実効値で演算した場合の積分周期を示します。 図中、現時刻より指数的に減少している波形が実際に積分される範囲を模式的に示しています。</p> 

2. 2. 22. 移動実効値関数

機能	X を k で指定した積分周期での実効値を戻します。
文法	RRV(k,X,m)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> 積分周期 τ (単位 sec)を記述します。なお、記述した k が複数の要素で構成されていても参照される値は Index0 の値のみとなります。また、k は、$0 < k$ でなくてはなりません。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X のデータ個数 \times サンプル周期は、積分周期 k より長くなくてはなりません。</p> <p>【m】<省略可><即値/数値属性参照チャンネル/演算式> 過渡領域付加有無フラグを記述します。 過渡領域付加フラグ:=0 の時:(記述省略した場合は、フラグ 0 と見なします) 過渡領域付加しないを意味し、後方積分となりますので全体のデータ長は積分周期分短くなります。 フラグ:<>0 の時: 過渡領域付加を意味し、前方積分となりますので、現在のデータ長は演算対象データ X と同じ長さですが、先頭から設定した積分周期 τ に達するまでが過渡領域となります。過渡領域とは、設定した積分周期 τ に満たない範囲を指します。</p>
記述例	RRV(0.125,#1) \rightarrow Ch1 を積分周期 125ms での移動実効値を戻します。
備考	<p>m=0 の時 演算結果のデータ個は τ 分データが短くなります。</p> $Ans_i = \left[\frac{1}{\tau} \int_{i-t}^{i+\tau} x_i^2 dt \right]^{\frac{1}{2}}$ <p>m<>0 の時 τ に満たない領域は過渡領域となり、正しい実効値を示しません。</p> $Ans_i = \left[\frac{1}{\tau} \int_{i-t-\tau}^t x_i^2 dt \right]^{\frac{1}{2}}$ <p>演算例: 上段\Rightarrow加速度、下段\Rightarrow移動加速度実効値(積分周期 0.125sec)</p> 

2. 2. 23. 符号反転関数

機能	X の符号を反転します。
文法	SGN(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列記述します。
記述例	SGN(#5)
備考	データ列に(-1)を掛けたことと等価です。



2. 2. 24. 平方根関数

機能	X の平方根を戻します。
文法	SQR(X)
引数	【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数値を記述します。なお、X は X>=0 でなくてはなりません。
記述例	SQR(ABS(#1)) → Ch1 の絶対値の平方根を戻します。
備考	$Ans_n = \sqrt{X_n}$ 演算対象データの要素数が1個の場合は戻り値も要素数 1 個となり、演算対象データが複数要素の場合は戻り値も複数要素となります。

2. 2. 25. リサンプリング関数

機能	元々のサンプリング周波数と近接したサンプリング周波数でリサンプリングする場合に使用するリサンプリング関数です。
文法	RSM(k,X)
引数	【k】<必須><即値/数値属性参照チャンネル/演算式> リサンプリング周波数を記述します。 【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数値を記述します。
記述例	RSM(1024,#1)
備考	結果のデータ個数は、リサンプルする周波数により増減します。例えば、データ個数 10,000 点のデータで元のサンプリング周波数が 1kHz で、リサンプルする周波数が 1.024kHz の場合、戻りデータ個数は、10,240 点となります。なお、設定したリサンプリング周波数でサンプルされた結果が 2 個以上存在しない場合は、実行時 Error となります。 $\Delta t_0 = PRD()$ $\Delta t_1 = \frac{1}{k}$ $X_m = \sum_{n=0}^{N-1} \left\{ X_n \frac{\sin\left(\frac{\pi}{\Delta t_0}(m \cdot \Delta t_1 - n \Delta t_0)\right)}{\frac{\pi}{\Delta t_0}(m \cdot \Delta t_1 - n \cdot \Delta t_0)} \right\}$ なお、本関数は、サンプリング周期自体は変更されませんので、チャンネル間演算機能で使用する場合、留意して下さい。また、Script 中で使用する場合は、本関数行以降で def sampl_period 文にて、明示的にサンプリング周波数を定義する必要があります。また、元のサンプリング周波数と設定したサンプリング周波数の比等により、生成波形の先端部および終端部に波形ひずみを生じる場合があります。

2. 2. 26. クレストファクタ演算関数

機能	X を k で指定した積分周期内の最大値をその実効値で割り算したクレストファクタを戻します。
文法	CFT(k,X,m)
引数	【k】<必須><即値/数値属性参照チャンネル/演算式> 積分周期 τ (単位 sec) を記述します。なお、記述した k が複数の要素で構成されていても参照される値は Index0 の値のみとなります。また、k は、0<k でなくてはなりません。 【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数値を記述します。なお、X のデータ個数×サンプリング周期は、積分周期 k より長くなくてはなりません。 【m】<省略可><即値/数値属性参照チャンネル/演算式> 過渡領域付加有無フラグを記述します。 過渡領域付加フラグ=0 の時:(記述省略した場合は、フラグ 0 と見なします) 過渡領域付加しないを意味し、後方積分となりますので全体のデータ長は積分周期分短くなります。 フラグ<>0 の時: 過渡領域付加を意味し、前方積分となりますので、現在のデータ長は演算対象データ X と同じ長さですが、先頭から設定した積

	分 周期 τ に達するまでが過渡領域となります。過渡領域とは、設定した積分周期 τ に満たない範囲を指します。
記述例	CFT(0.125,#1) → Ch1 を積分周期 125ms でのクレストファクタを戻します。
備考	クレストファクタ(Crest Factor)とは、積分区間内の ABS(最大値)を積分区間内の実効値で割り算した結果を意味し、戻り値は必ず正となります。 m=0 の時 演算結果のデータ個は τ 文データが短くなります。 m<>0 の時 τ に満たない領域は過渡領域となり、正しいクレストファクタを示しません。

2. 3. 複素数取扱関数

2. 3. 1. フーリエ変換関数

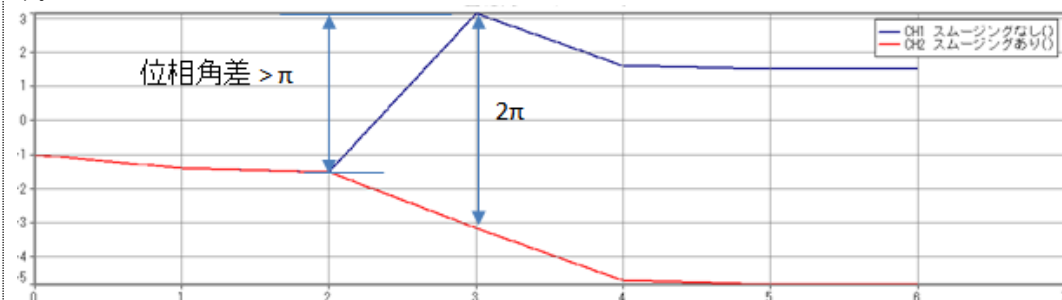
機能	記述された引数のフーリエ変換結果を戻します																						
文法	FFT(k,n,m,i,X)																						
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> FFT 点数を記述します。記述範囲は 256~32768 となります。使用される FFT 点数は $2^{\text{INT}(\log_2(k))}$ として 2 のべき乗で割り切れる値に整合化されます。</p> <p>【n】<必須><即値/数値属性参照チャネル/演算式> 使用する窓関数コードを記述します。下表に窓関数コードを示します。</p> <table border="1"> <thead> <tr> <th>窓関数コード</th> <th>窓関数</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>なし</td> </tr> <tr> <td>1</td> <td>ハミング</td> </tr> <tr> <td>2</td> <td>ハニング</td> </tr> <tr> <td>3</td> <td>ブラックマン</td> </tr> <tr> <td>4</td> <td>カイザー・ベッセル</td> </tr> <tr> <td>5</td> <td>フラット・トップ</td> </tr> </tbody> </table> <p>【m】<必須><即値/数値属性参照チャネル/演算式> 検算コードを記述します。下表に演算コード表を示します。</p> <table border="1"> <thead> <tr> <th>演算コード</th> <th>演算結果</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>スペクトラム</td> </tr> <tr> <td>1</td> <td>パワー・スペクトラム</td> </tr> <tr> <td>2</td> <td>パワー・スペクトラム密度</td> </tr> </tbody> </table> <p>【i】<必須><即値/数値属性参照チャネル/演算式> オーバーラップ率を単位%で記述します。記述範囲は $0 \leq \text{オーバーラップ率} < 100$ となります。小数点以下を含む場合は切り捨てられて参照されます。尚、100 と記述された場合は平均化演算を行わず演算対象数列の先頭から k で記述したデータ個数まで 1 回のフーリエ変換となります。</p> <p>【X】<必須><即値/数値属性参照チャネル/演算式> 解析対象チャネルを記述します。解析対象チャネルの要素数は k で設定した FFT 点数以上で必要です。</p>	窓関数コード	窓関数	0	なし	1	ハミング	2	ハニング	3	ブラックマン	4	カイザー・ベッセル	5	フラット・トップ	演算コード	演算結果	0	スペクトラム	1	パワー・スペクトラム	2	パワー・スペクトラム密度
窓関数コード	窓関数																						
0	なし																						
1	ハミング																						
2	ハニング																						
3	ブラックマン																						
4	カイザー・ベッセル																						
5	フラット・トップ																						
演算コード	演算結果																						
0	スペクトラム																						
1	パワー・スペクトラム																						
2	パワー・スペクトラム密度																						
記述例	FFT(2048,1,0,90,\$1)																						
備考	<p>関数の戻り数列は index 順に実数,虚数,実数虚数と交互に並ぶ直交座標系複素数数列となり、要素数は与えた解析チャネルの要素数に同じ個数となります。</p> <p>なお、パワー・スペクトラム及びパワー・スペクトラム密度を指定した場合及びスペクトラムで平均化処理を行った場合は、虚数部はゼロとなります。各スペクトラに対応する周波数を求める関数は特に用意していません。スペクトラに対応した周波数は次の式で計算します。</p> <ul style="list-style-type: none"> ◆ スペクトラ間隔(Δf)の計算 $\Delta f = 1 / (\text{FFT 点数} \times \text{サンプリング周期}) = \text{サンプリング周波数} / \text{FFT 点数}$ 計算例: サンプリング 2kHz で FFT 点数 2048 の場合 $\Delta f = 2000 / 2048 = 0.9765625\text{Hz}$ ◆ スペクトラ対応周波数(fn)の計算 $fn = \Delta f \times \text{スペクトラ番号} = (\text{サンプリング周波数} / \text{FFT 点数}) \times \text{スペクトラ番号}$ 計算例: サンプリング周波数 2kHz、FFT 点数 2048 でスペクトラ番号 3 番目の周波数を求める場合 $f3 = 2000 / 2048 \times 3 = 2.9296875$ ※ スペクトラ番号は 0~FFT 点数/2-1 となり、例えば FFT 点数 2048 の時は、0~1023 まで存在します。Index0 位置は直流成分です。 ◆ スペクトラ対応周波数数列を一度に生成する場合 累算関数(ACC 関数)と波形生成関数(DAG 関数)を使用し次式で求めます。 $(\text{ACC}(\text{DAG}(\text{FFT 点数}, 1)) - 1) \times \Delta f$ 例えばサンプリング周波数 2kHz、FFT 点数 2048 の時は下記の用に記述します。 $\\$2 = (\text{ACC}(\text{DAG}(2048, 1)) - 1) * 2000 / 2048$ ◆ 解析対象音圧チャネル ch1 とし FFT 点数 4096、窓関数なし、オーバーラップ率 80%でスペクトラム解析を行いグラフ表示 $\\$1 = \text{FFT}(4096, 0, 0, 80, \#1) \quad /* \text{FFT 解析スペクトラム結果を\\$1 に格納}*/$ $\\$2 = (\text{ACC}(\text{DAG}(4096/2, 1)) - 1) * 1 / (4096 * \text{PRD}()) \quad /* \text{スペクトラ対応周波数数列を生成}*/$ $\\$1 = \text{SEP}(0, 1, \\$1) \quad /* \text{実数部のみ抽出}*/$ <pre>def graph_id @1 "FFT" def graph_x_axis @1 0.0001, 10 F4 log def graph_y_axis @1 0, 0 F3 plot @1 \$2 \$1</pre> 																						

2. 3. 2. 極座標変換関数

機能	記述された引数を直交座標系複素数(実数、虚数)と見なし極座標複素数(絶対値、角度)に変換します。 なお、戻り角度単位はラジアンとなります。
文法	PCX(X)
引数	【X】<必須><即値/数値属性参照チャンネル/演算式> 直交座標系複素数数列を記述します。属性が異なっている場合は Error となります。 複素数数列は、index 順に実数,虚数,実数,虚数と交互に並んでいるものと見なします。
記述例	PCX(\$1)
備考	演算内容は $X=(a+bi)$ と見なし 絶対値 r は $\sqrt{a^2+b^2}$ 角度 θ は $\tan^{-1}(b/a)$ となり、 $\pm\pi$ 範囲となります。 但し、 a が 0 の場合、 $b>0$ の時 $\pi/2$ 、 $b<0$ の時 $-\pi/2$ 、 $b=0$ 時、0 が戻ります。 戻り数列は、index 順に絶対値,角度,絶対値,角度と交互に極座標系複素数数列となり、属性は極座標系複素数となります。 ◆ 絶対値と位相角を分離する場合は SEP 関数を使用します。 \$3 に極座標系複素数数列が格納されている場合絶対値の抽出 \$4 = SEP(0,1,\$3) 絶対値を実効値に変換する場合は絶対値抽出操作時に $\sqrt{2}$ で割り算して変換します。 \$4 = SEP(0,1,\$3)/SQR(2) 位相角の抽出は \$5 = SEP(1,1,\$3) 位相角単位をラジアン(rad)から度(deg)への変換は位相角抽出操作時に DEG 関数を使用して変換します。 \$5 = DEG(SEP(1,1,\$3))

2. 3. 3. 位相角アンラップ関数

機能	連続する位相角の差が π ラジアン以上だった場合に、 $\pm 2\pi$ ラジアンを加算し、位相角の変化量が π ラジアン以下になるようにスムージングします。
文法	UNW(X)
引数	【k】<必須><即値/数値属性参照チャンネル/演算式> 位相角をラジアン単位で記述します。
記述例	UNW(\$1)
備考	位相角が \$1 = -1,-1.4,-1.5,3.14,1.6,1.5,1.5 だった場合、本関数を使用した場合としなかった場合の位相角は下図のようになります。



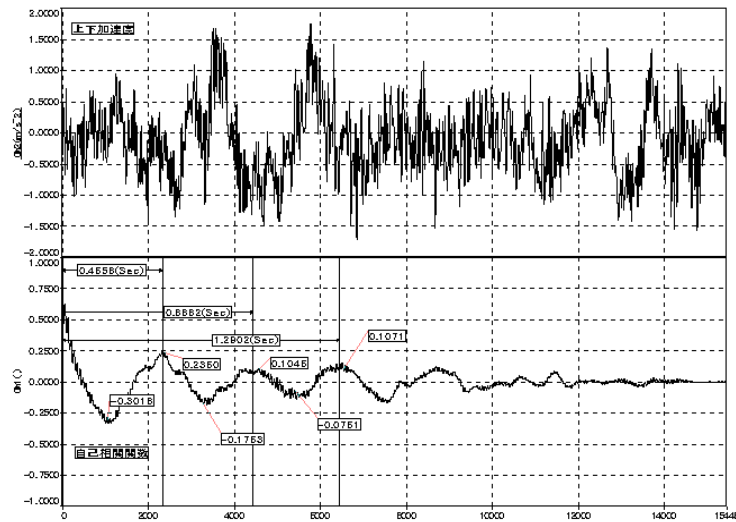
2. 3. 4. FFT 周波数列取得関数

機能	k に従い、スペクトラムに対応した周波数列を戻します。
文法	FFQ(k,m)
引数	【k】<必須><即値/数値属性参照チャンネル/演算式> k<0: 直流成分を除いた周波数列を戻します。 k=0: 直流成分を含んだ周波数列を戻します。 k>1~m-1: スペクトラム番号 k とみなしスペクトラムに対応した $k\Delta f$ を戻します。 【m】<必須><即値/数値属性参照チャンネル/演算式> FFT 点数を記述します。
記述例	FFQ(-1,2048) FFQ(16384)
備考	関数内部でサンプリング周期を参照するため、サンプリング周期が不定の場合は def_sampl_period で本関数を使用する前にサンプリング周期を定義する必要があります。 k<0: ACC(DAG(m/2-1,1))/m*PRD() と等価です。 k=0: (ACC(DAG(m/2,1))-1)/m*PRD() と等価です。 k>1~m-1: 1/PRD()/m*k と等価です。

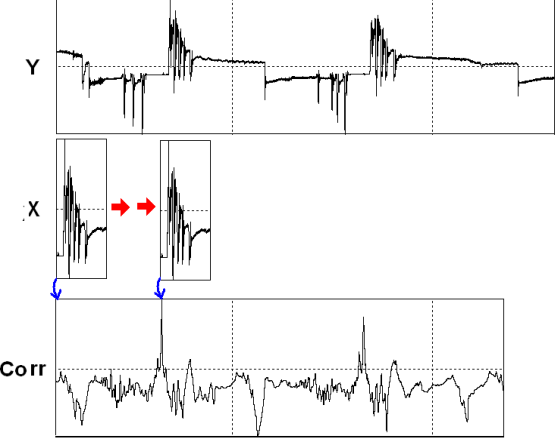
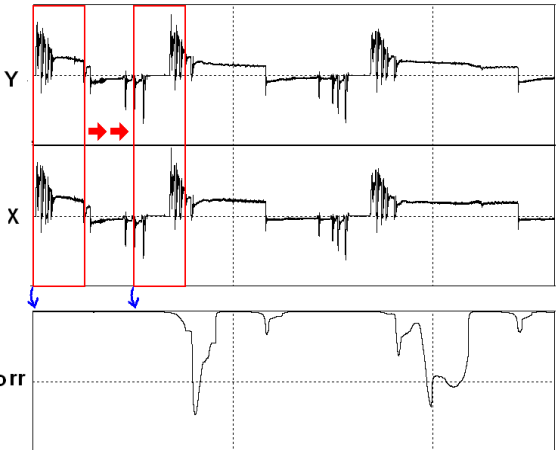
2. 4. 統計関連関数

2. 4. 1. 相関関数

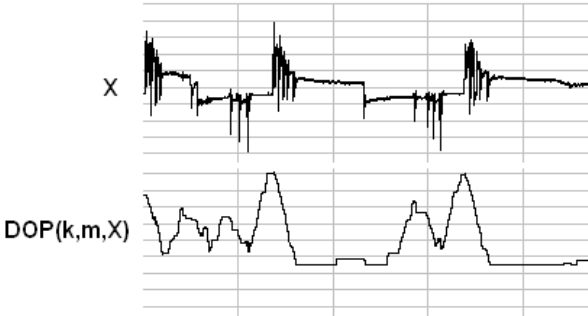
機能	演算対象データ X,Y の相関関数を戻します。
文法	COR(X,Y)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。 【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、演算対象数列 X,Y のデータ個数は等しい必要があります。
記述例	COR(\$1,\$2) COR(#4,#4)
備考	X と Y に同じチャンネルを指定した場合は自己相関関数が、異なるチャンネルを指定した場合は相互相関関数が求まります。 解析例: 自己相関関数を求めています。



2.4.2. 移動相関値関数

機能	XとYの相関係数値を戻します。
文法	MCO(k,X,Y)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> 相関値を求めるデータ個数を記述します。なお、k=0 は特別な意味を持ち、Y で記述されたデータ列上を X で記述されたデータ列がスライディングして相関係数値を戻します。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 比較数列を記述します。なお、k=0 以外の場合、X のデータ個数>k でなければなりません。</p> <p>【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、k=0 の時は X のデータ個数 < Y のデータ個数でなければなりません。</p>
記述例	<p>MCO(5000,#6,#5)</p> <p>MCO(0,\$1,#3)</p>
備考	<p>k=0 の時は、Y 上を X がスライディングしながら相関値を戻します。 演算結果のデータ個数は Y のデータ個数より X のデータ個数分短くなります。</p>  <p>k>0 の時は、X、Y の同じ Index から k 個分のデータ個数を切り出しスライディングして相関値を戻します。 演算結果のデータ個数は k で指定したデータ個数分短くなります。</p> 

2. 4. 3. 移動振幅頻度関数

機能	X の振幅計数移動幅 k の範囲から極大値/極小値を抽出し、無効振幅値 m を越えた極大値極小値から振幅に変換し、振幅数を戻します。
文法	DOP(k,m,X)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> 振幅計数移動幅を単位 sec で記述します。</p> <p>【m】<必須><即値/数値属性参照チャンネル/演算式> 無効振幅値を記述します。なお、単位は演算対象数列 X と同じ単位です。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	DOP(15,20,#4)
備考	<p>戻りデータ番号 0 は演算対象数列の先頭から計数幅 k 内の振幅数、同様に戻りデータ番号 1 は演算対象数列の先頭+1 から計数幅 k 内の振幅数なり、結果のデータ個数は、演算対象数列のデータ個数から計数幅 k 個分少なくなります。</p> 

2. 4. 4. 最大値関数

機能	X の最も大きい値を戻します。k が記述された場合は、k で記述した区間毎の最も大きい値を戻します。
文法	MAX(k,X)
引数	<p>【k】<省略><即値/数値属性参照チャンネル/演算式> 演算範囲を Index で記述します。 即値または要素 1 個の参照チャンネルで記述した場合 Index0 から k 個毎を演算範囲とします。最終区間が設定したデータ個数に満たない場合は、該当区間の演算を行いません。設定したデータ個数に演算対象数列の個数が満たない場合は実行時エラーとなります。</p> <p>複数要素の参照チャンネルで記述した場合 k(n)を X の Index 値として k(n)から k(n+1)-1 までの Index を演算範囲とします。複数要素の場合 k は昇順並びの必要があり、k で指定した Index が存在しない場合は実行時エラーとなります。</p> <p>複数要素の参照チャンネルで記述し要素数が演算対象数列と同じ場合 記述された k を"1"と"0"で構成された論理数列と見なし、論理"1"の区間を演算範囲とします。論理数列が全て"0"の場合は実行時エラーとなります。なお、全て"1"の場合は記述省略された時と同じ結果となります。また、記述された k が論理数列でない場合、結果は保証されません。 記述省略した場合は、演算対象数列 X の全体を演算対象とします。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	<p>MAX(#2)</p> <p>MAX(\$1,#2)</p>
備考	<p>戻り個数は下記のようになります。</p> <p>引数 k が省略された場合: データ戻り個数 1 個 引数 k が即値または要素数 1 の場合: 戻りデータ個数=INT(LEN(X)/k) 例えば X のデータ数が 1000、k=300 の場合は戻り個数は 3 個となります。</p> <p>引数 k が複数要素の場合: データ個数=LEN(k)-1 例えば k=100,200,300 の場合、Index100 から 199 までの演算値、Index200 から 299 までの演算値の 2 個が戻ります。</p> <p>引数 k が 0/1 で記述されていた場合 例えば k=1,1,1,0,0,1,1,1,1,0,1,1、X の要素数が 12 の場合、k が 1 で記述されている 0~2、5~8、10~11 の演算結果の 3 個が戻ります。</p>

2. 4. 5. 最小値関数

機能	X の最も小さい値を戻します。k が記述された場合は、k で記述した区間毎の最も小さい値を戻します。
文法	MIN(k,X)
引数	<p>【k】<省略可><即値/数値属性参照チャンネル/演算式> 演算範囲を Index で記述します。 即値または要素 1 個の参照チャンネルで記述した場合 Index0 から k 個毎を演算範囲とします。最終区間が設定したデータ個数に満たない場合は、該当区間の演算を行いません。 設定したデータ個数に演算対象数列の個数が満たない場合は実行時エラーとなります。</p> <p>複数要素の参照チャンネルで記述した場合 k(n)を X の Index 値として k(n)から k(n+1)-1 までの Index を演算範囲とします。複数要素の場合 k は昇順並びの必要があり、k で指定した Index が存在しない場合は実行時エラーとなります。</p> <p>複数要素の参照チャンネルで記述し要素数が演算対象数列と同じ場合 記述された k を"1"と"0"で構成された論理数列と見なし、論理"1"の区間を演算範囲とします。論理数列が全て"0"の場合は実行時エラーとなります。なお、全て"1"の場合は記述省略された時と同じ結果となります。また、記述された k が論理数列でない場合、結果は保証されません。</p> <p>記述省略した場合は、演算対象数列 X の全体を演算対象とします。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	MIN(\$2) MIN(\$1,\$2)
備考	<p>戻り個数は下記の様になります。</p> <p>引数 k が省略された場合: データ戻り個数 1 個 引数 k が即値または要素数 1 の場合: 戻りデータ個数=INT(LEN(X)/k) 例えば X のデータ数が 1000、k=300 の場合は戻り個数は 3 個となります。</p> <p>引数 k が複数要素の場合: データ個数=LEN(k)-1 例えば k=100,200,300 の場合、Index100 から 199 までの演算値、Index200 から 299 までの演算値の 2 個が戻ります。</p> <p>引数 k が 0/1 で記述されていた場合 例えば k=1,1,1,0,0,1,1,1,1,0,1,1、X の要素数が 12 の場合、k が 1 で記述されている 0~2、5~8、10~11 の演算結果の 3 個が戻ります。</p>

2. 4. 6. 平均値関数

機能	X の算術平均値を戻します。k が記述された場合は、k で記述した区間毎の算術平均値を戻します。
文法	MEA(k,X)
引数	<p>【k】<省略可><即値/数値属性参照チャンネル/演算式> 演算範囲を Index で記述します。 即値または要素 1 個の参照チャンネルで記述した場合 Index0 から k 個毎を演算範囲とします。最終区間が設定したデータ個数に満たない場合は、該当区間の演算を行いません。 設定したデータ個数に演算対象数列の個数が満たない場合は実行時エラーとなります。</p> <p>複数要素の参照チャンネルで記述した場合 k(n)を X の Index 値として k(n)から k(n+1)-1 までの Index を演算範囲とします。複数要素の場合 k は昇順並びの必要があり、k で指定した Index が存在しない場合は実行時エラーとなります。</p> <p>複数要素の参照チャンネルで記述し要素数が演算対象数列と同じ場合 記述された k を"1"と"0"で構成された論理数列と見なし、論理"1"の区間を演算範囲とします。論理数列が全て"0"の場合は実行時エラーとなります。なお、全て"1"の場合は記述省略された時と同じ結果となります。また、記述された k が論理数列でない場合、結果は保証されません。</p> <p>記述省略した場合は、演算対象数列 X の全体を演算対象とします。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	MEA(\$2+\$3) MEA(\$1,#2)
備考	<p>$Ans = \frac{1}{m} \sum_{i=1}^m X_i$</p> <p>戻り個数は下記の様になります。</p> <p>引数 k が省略された場合: データ戻り個数 1 個 引数 k が即値または要素数 1 の場合: 戻りデータ個数=INT(LEN(X)/k) 例えば X のデータ数が 1000、k=300 の場合は戻り個数は 3 個となります。</p> <p>引数 k が複数要素の場合: データ個数=LEN(k)-1 例えば k=100,200,300 の場合、Index100 から 199 までの演算値、Index200 から 299 までの演算値の 2 個が戻ります。</p> <p>引数 k が 0/1 で記述されていた場合 例えば k=1,1,1,0,0,1,1,1,1,0,1,1、X の要素数が 12 の場合、k が 1 で記述されている 0~2、5~8、10~11 の演算結果の 3 個が戻ります。</p>

2. 4. 7. 中央値関数

機能	X の中央値を戻します。
文法	MED(k,X)
引数	<p>【k】<省略可><即値/数値属性参照チャンネル/演算式> 演算範囲を Index で記述します。 記述省略した場合 演算対象数 X の全体を演算対象とします。 即値または要素 1 個の参照チャンネルで記述した場合 Index0 から k 個毎を演算範囲とします。最終区間が設定したデータ個数に満たない場合は、該当区間の演算を行いません。設定したデータ個数に演算対象数列の個数が満たない場合は実行時エラーとなります。 複数要素の参照チャンネルで記述し要素数が演算対象数列より少ない場合 k(n)を X の Index 値として k(n)から k(n+1)-1 までの Index を演算範囲とします。複数要素の場合 k は昇順並びの必要があり、k で指定した Index が存在しない場合は実行時エラーとなります。 複数要素の参照チャンネルで記述し要素数が演算対象数列と同じ場合 記述された k を"1"と"0"で構成された論理数列と見なし、論理"1"の区間を演算範囲とします。論理数列が全て"0"の場合は実行時エラーとなります。なお、全て"1"の場合は記述省略された時と同じ結果となります。また、記述された k が論理数列でない場合、結果は保証されません。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	MED(\$1)
備考	<p>演算対象数列を昇順に並べ、Index の中央の値を戻します。なお、要素数が偶数の場合は、中央の 2 個の平均値を戻します。戻り個数は下記の様になります。</p> <p>引数 k が省略された場合: データ戻り個数 1 個 引数 k が即値または要素数 1 の場合: 戻りデータ個数=INT(LEN(X)/k) 例えば X のデータ数が 1000、k=300 の場合は戻り個数は 3 個となります。 引数 k が複数要素の場合: データ個数=LEN(k)-1 例えば k=100,200,300 の場合、Index100 から 199 までの演算値、Index200 から 299 までの演算値の 2 個が戻ります。 引数 k が 0/1 で記述されていた場合 例えば k=1,1,1,0,0,1,1,1,1,0,1,1、X の要素数が 12 の場合、k が 1 で記述されている 0~2、5~8、10~11 の演算結果の 3 個が戻ります。</p>

2. 4. 8. 合計値関数

機能	X の合計値を戻します。k が記述された場合は、k で記述した区間毎の合計値を戻します。
文法	SUM(k,X)
引数	<p>【k】<省略可><即値/数値属性参照チャンネル/演算式> 演算範囲を Index で記述します。 即値または要素 1 個の参照チャンネルで記述した場合 Index0 から k 個毎を演算範囲とします。最終区間が設定したデータ個数に満たない場合は、該当区間の演算を行いません。設定したデータ個数に演算対象数列の個数が満たない場合は実行時エラーとなります。 複数要素の参照チャンネルで記述した場合 k(n)を X の Index 値として k(n)から k(n+1)-1 までの Index を演算範囲とします。複数要素の場合 k は昇順並びの必要があり、k で指定した Index が存在しない場合は実行時エラーとなります。 複数要素の参照チャンネルで記述し要素数が演算対象数列と同じ場合 記述された k を"1"と"0"で構成された論理数列と見なし、論理"1"の区間を演算範囲とします。論理数列が全て"0"の場合は実行時エラーとなります。なお、全て"1"の場合は記述省略された時と同じ結果となります。また、記述された k が論理数列でない場合、結果は保証されません。 記述省略された場合は、演算対象数列 X の全体を演算対象とします。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	<p>SUM(#1) SUM(\$1,#1)</p>
備考	<p>$Ans = \sum_{i=1}^m X_i$</p> <p>戻り個数は下記の様になります。</p> <p>引数 k が省略された場合: データ戻り個数 1 個 引数 k が即値または要素数 1 の場合: 戻りデータ個数=INT(LEN(X)/k) 例えば X のデータ数が 1000、k=300 の場合は戻り個数は 3 個となります。 引数 k が複数要素の場合: データ個数=LEN(k)-1 例えば k=100,200,300 の場合、Index100 から 199 までの演算値、Index200 から 299 までの演算値の 2 個が戻ります。 引数 k が 0/1 で記述されていた場合 例えば k=1,1,1,0,0,1,1,1,1,0,1,1、X の要素数が 12 の場合、k が 1 で記述されている 0~2、5~8、10~11 の演算結果の 3 個が戻ります。</p>

2. 4. 9. 標準偏差関数

機能	X の標準偏差値を戻します。k が記述された場合は、k で記述した区間毎の標準偏差値を戻します。
文法	STD(k,X)
引数	<p>【k】<省略可><即値/数値属性参照チャンネル/演算式> 演算範囲を Index で記述します。 即値または要素 1 個の参照チャンネルで記述した場合 Index0 から k 個毎を演算範囲とします。最終区間が設定したデータ個数に満たない場合は、該当区間の演算を行いません。 設定したデータ個数に演算対象数列の個数が満たない場合は実行時エラーとなります。</p> <p>複数要素の参照チャンネルで記述した場合 k(n)を X の Index 値として k(n)から k(n+1)-1 までの Index を演算範囲とします。複数要素の場合 k は昇順並びの必要があり、k で指定した Index が存在しない場合は実行時エラーとなります。</p> <p>複数要素の参照チャンネルで記述し要素数が演算対象数列と同じ場合 記述された k を"1"と"0"で構成された論理数列と見なし、論理"1"の区間を演算範囲とします。論理数列が全て"0"の場合は実行時エラーとなります。なお、全て"1"の場合は記述省略された時と同じ結果となります。また、記述された k が論理数列でない場合、結果は保証されません。 記述省略した場合は、演算対象数列 X の全体を演算対象とします。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	STD(\$1) STD(\$2,#4)
備考	$Ans = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$ <p>戻り個数は下記の様になります。 引数 k が省略された場合: データ戻り個数 1 個 引数 k が即値または要素数 1 の場合: 戻りデータ個数=INT(LEN(X)/k) 例えば X のデータ数が 1000、k=300 の場合は戻り個数は 3 個となります。 引数 k が複数要素の場合: データ個数=LEN(k)-1 例えば k=100,200,300 の場合、Index100 から 199 までの演算値、Index200 から 299 までの演算値の 2 個が戻ります。 引数 k が 0/1 で記述されていた場合 例えば k=1,1,1,0,0,1,1,1,1,0,1,1、X の要素数が 12 の場合、k が 1 で記述されている 0~2、5~8、10~11 の演算結果の 3 個が戻ります。</p>

2. 4. 10. 積和演算関数

機能	数列 X と数列 Y の積和演算値を戻します。
文法	MAC(X,Y)
引数	<p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p> <p>【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。 X と Y 数列の要素数は等しくなければなりません。</p>
記述例	MAC(\$1,\$2)
備考	$T_{xy} = \sum_{i=1}^n (X_i \cdot Y_i)$ <p>既存関数合計関数 SUM(X)を使用して MAC(\$1,\$2) = SUM(\$1*\$2) で求めた結果と一致します。</p>

2. 4. 11. 共分散関数

機能	数列 X と数列 Y の共分散値を戻します。
文法	VAR(X,Y)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。 【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。 X と Y 数列の要素数は等しくなければなりません。
記述例	VAR(\$1,\$2)
備考	$S_{xy} = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$ 数列 X と Y に同じチャンネルを記述すると、分散値となります。 なお、分散値は既存関数標準偏差関数 STD(X)を 2 乗した値と一致します。 $\text{VAR}(\$1,\$1) = \text{STD}(\$1)^2$ 又、共分散は既存関数積和演算関数 MAC(X,Y)と平均値関数 MEA(X)を使用して $\text{VAR}(\$1,\$2) = \text{MAC}(\$1,\$2)/\text{LEN}(\$1) - \text{MEA}(\$1)*\text{MEA}(\$2)$ 求めた結果と一致します。

2. 4. 12. 相関係数関数

機能	数列 X と数列 Y の相関係数を戻します。
文法	CRR(X,Y)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。 【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。 X と Y 数列の要素数は等しくなければなりません。
記述例	CRR(\$1,\$2)
備考	$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$ 既存関数 共分散関数 VAR(X,Y)と標準偏差関数 STD(X)を使用して $\text{CRR}(\$1,\$2) = \text{VAR}(\$1,\$2)/(\text{STD}(\$1)*\text{STD}(\$2))$ または COR(X,Y)相互相関関数で求めた Index0 の値と一致します。

2. 4. 13. 最小二乗法係数関数

機能	数列 X と数列 Y の最小二乗法による $y=ax+b$ の傾き a とオフセット b を戻します。
文法	LSM(X,Y)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。 【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。 ※ X と Y 数列の要素数は等しくなければなりません。
記述例	LSM(\$1,\$2)
備考	戻り数列の Index0 に傾き a が、Index1 にオフセット b が戻ります。 $a = \frac{n \sum_{i=1}^n X_i Y_i - \sum_{i=1}^n X_i \sum_{i=1}^n Y_i}{n \sum_{i=1}^n X_i^2 - (\sum_{i=1}^n X_i)^2}$ $b = \frac{n \sum_{i=1}^n X_i^2 \sum_{i=1}^n Y_i^2 - \sum_{i=1}^n X_i Y_i \sum_{i=1}^n X_i}{n \sum_{i=1}^n X_i^2 - (\sum_{i=1}^n X_i)^2}$ 既存関数 積和演算関数 MAC(X,Y)と合計値関数 SUM(X)を使用して $\text{LSM}(\$1,\$2)_0 = (\text{LEN}(\$1)*\text{MAC}(\$1,\$2) - \text{SUM}(\$1)*\text{SUM}(\$2))/(\text{LEN}(\$1)*\text{SUM}(\$1^2) - \text{SUM}(\$1)^2)$ で求めた結果と一致します。

2. 4. 14. ガンマ関数

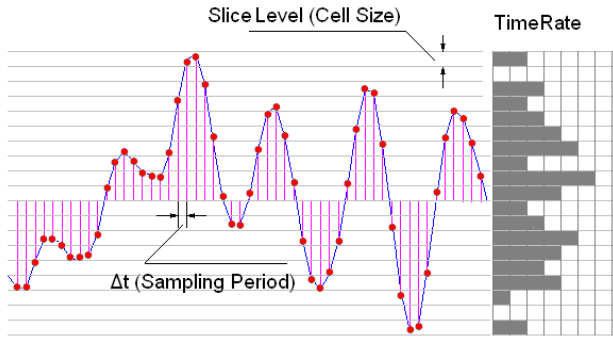
機能	引数のガンマ関数値を戻します。
文法	GMM(X)
引数	【X】<必須><数値属性参照チャネル/演算式> 演算対象数列を記述します。
記述例	GMM(\$1)
備考	戻り値個数は演算対象数列の個数に一致します。 Xに負数或いは $X < 1e-20$ 以下が含まれていた場合、戻り値は-1となります。 $\Gamma(x) = \int_0^{\infty} u^{x-1} \cdot e^{-u} du$

2. 4. 15. セル番号/セル代表値演算関数

機能	セルサイズ、セル個数からセル番号またはセル中央値数列を戻します。
文法	CNV(k,m,n,f)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> 戻り属性を示すフラグで、0または1を記述します。 戻り属性⇒0の時はセル番号数列を戻し、戻り属性⇒1の時はセル中央値数列を戻します。 なお、複数要素を持つ数列で記述してもIndex0のみ参照されます。 【m】<必須><即値/数値属性参照チャネル/演算式> セルサイズを正数で記述します。負数で記述された場合は絶対値化して参照されます。又、複数要素を持つ数列で記述してもIndex0のみ参照されます。 【n】<必須><即値/数値属性参照チャネル/演算式> セルの合計個数を正整数で記述します。負数で記述された場合は絶対値化して参照されます。小数点以下を含む場合は切り捨てられます。又、後述するフラグが1の場合、記述されたセル合計個数が奇数の場合は+1して偶数化して参照されます。なお、複数要素を持つ数列で記述してもIndex0のみ参照されます。 【f】<省略可><即値/数値属性参照チャネル/演算式> セルが±領域を持つか正領域だけかを示す流域フラグで、0または1を記述します。 領域フラグ⇒0の時は正領域、領域フラグ⇒1の時は正負領域を意味します。なお、記述省略した場合は0と見なします。又、複数要素を持つ数列で記述してもIndex0のみ参照されます。
記述例	CNV(1,25,256,0)
備考	戻り数列は昇順並びとなります。戻り数列の個数は、合計セル数に一致します。 セル番号の演算方法 n:=セル合計個数として 領域フラグ=0の時: セル番号⇒ 1,2,3,4,5,6,...n 領域フラグ=1の時: セル番号⇒ -n/2,...-3,-2,-1,1,2,3,...n/2 セル代表値の演算方法(セル内は一様分布していると仮定) C:=セル番号、m:=セルサイズとして 領域フラグ=0の時: 代表値⇒ C·m-m/2 領域フラグ=1の時: 代表値(C<0の時)⇒C·m+m/2 代表値(C>0の時)⇒C·m-m/2

2. 5. 頻度解析関連関数

2. 5. 1. 度数分布関数

機能	演算対象データの度数分布(時間率頻度)を戻します。
文法	TRC(k,m,X)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> セルのサイズ(区画の大きさ)を記述します。</p> <p>【m】<必須><即値/数値属性参照チャンネル/演算式> セルの合計個数を記述します。ここで記述された合計個数/2 がそれぞれ正領域個数および負領域個数となります。 奇数で記述された場合、自動的に関数内で+1 され、偶数化して演算されます。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	TRC(0.1,200,#1)
備考	<p>データ値毎に当該セルに計数します。 総頻度数は解析対象データ個数に等しくなります。</p>  <p>演算結果の格納値と Index の関係は、セル中央値の昇順に並びます。 例えば、セルサイズが 10 として、セルの個数が 100 と設定した場合、セル番号(仮想)は-50,-49,-48,・・・-3,-2,-1,1,2,3,4,・・・48,49,50 と並んで合計 100 個のセル(区画)があると考えます。セル番号-50 つまり Index0 は、セルサイズが 10 ですので、$-500 < X_i < -490$ の範囲にあるデータの個数を計数し、同様にセル番号-1 は $-10 < X_i < 0$ の範囲にあるデータの個数を計数、セル番号 1 は、$0 < X_i < 10$ の範囲のデータ個数を計数します。 なお、範囲外のデータ個数は、負方向に範囲外のデータ個数はセル番号-50 に加算し、正方向に範囲外のデータ個数は、セル番号 50 に加算します。</p>

2.5.2. 度数分布統計値関数

機能	度数分布演算対象チャンネルの度数分布解析したセル区画毎の統計値演算対象チャンネルの統計値を戻します。																
文法	TRX(k,m,X,Y,a)																
引数	<p>[k]<必須><即値/数値属性参照チャンネル/演算式> セルのサイズ(区画の大きさ)を記述します。</p> <p>[m]<必須><即値/数値属性参照チャンネル/演算式> セルの合計個数を記述します。ここで記述された合計個数/2 がそれぞれ正領域個数および負領域個数となります。奇数で記述された場合、自動的に関数内で+1し、偶数化して演算されます。</p> <p>[X]<必須><収録チャンネル/数値属性参照チャンネル/演算式> 時間率頻度解析演算対象数値を記述します。</p> <p>[Y]<必須><収録チャンネル/数値属性参照チャンネル/演算式> 統計値演算対象数値を記述します。なお、X と Y の要素数は一致している必要があります。又、X と Y は同じチャンネルでも問題ありません。</p> <p>[a]<省略可><即値/数値属性参照チャンネル> 求める統計値種別フラグを記述します。記述する統計値種別フラグを下表に示します。</p> <table border="1" data-bbox="347 725 651 958"> <thead> <tr> <th>k</th> <th>演算内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>平均値</td> </tr> <tr> <td>1</td> <td>最大値</td> </tr> <tr> <td>2</td> <td>最小値</td> </tr> <tr> <td>3</td> <td>標準偏差</td> </tr> <tr> <td>4</td> <td>実効値</td> </tr> <tr> <td>5</td> <td>合計値</td> </tr> <tr> <td>6</td> <td>データ個数</td> </tr> </tbody> </table> <p>なお、記述省略した場合は、a=0 と等価で平均値となります。</p>	k	演算内容	0	平均値	1	最大値	2	最小値	3	標準偏差	4	実効値	5	合計値	6	データ個数
k	演算内容																
0	平均値																
1	最大値																
2	最小値																
3	標準偏差																
4	実効値																
5	合計値																
6	データ個数																
記述例	TRX(10,200,#1,#3,0)																
備考	戻りデータ個数は m で記述したセルの個数となります。又、戻り値の並び順は負領域の最も小さいセルから正領域の最も大きいセルの昇順となります。 本関数は、例えば、回転数チャンネルを時間率頻度対象チャンネル、圧力を統計値演算対象チャンネルとした場合、回転数チャンネルを時間率頻度解析した結果の当該セルに計数されるデータの Index と同じ Index の圧力チャンネルの統計値をセル毎に求めます。従って TRX(10,200,#1,#1,6)は TRC(10,200,#1)と同じ結果となります。																

2.5.3. 二次元時間率頻度解析関数

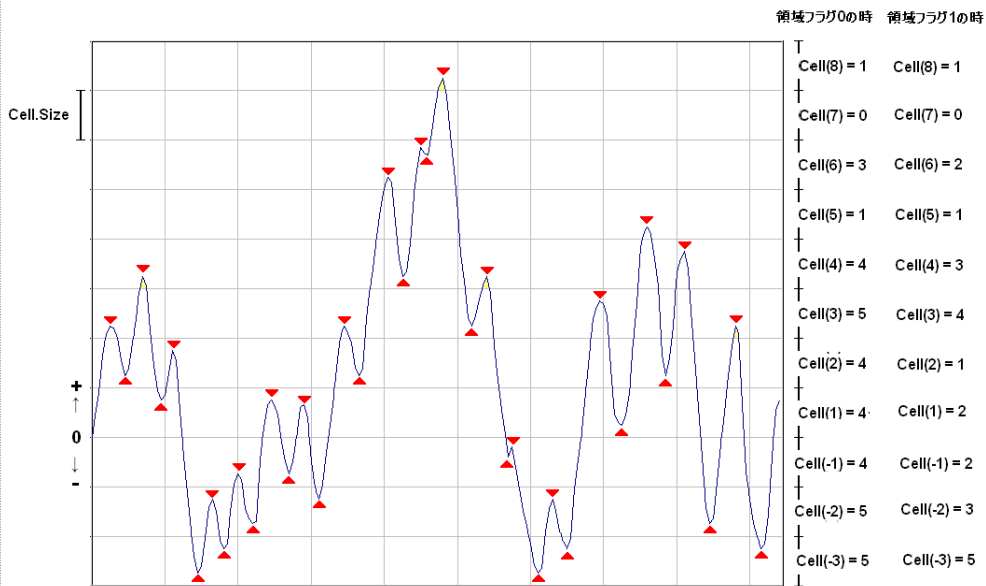
機能	2つの解析対象数値の 2 次元時間率頻度解析を行います。
文法	TR2(k,m,X,i,n,Y)
引数	<p>[k]<必須><即値/数値属性参照チャンネル/演算式> 対象数値 1 のセルの大きさを記述します。セル間隔と同じ意味です。</p> <p>[m]<必須><即値/数値属性参照チャンネル/演算式> 対象数値 1 のセルの個数を記述します。正負領域を持ちますので領域毎に、設定したセル個数/2 となります。なお、奇数で記述された場合は内部で+1して参照します。</p> <p>[X]<必須><数値属性参照チャンネル/演算式> 頻度解析対象数値 1 を記述します。</p> <p>[i]<必須><即値/数値属性参照チャンネル/演算式> 対象数値 2 のセルの大きさを記述します。セル間隔と同じ意味です。</p> <p>[n]<必須><即値/数値属性参照チャンネル/演算式> 対象数値 2 のセルの個数を記述します。正負領域を持ちますので領域毎に、設定したセル個数/2 となります。なお、奇数で記述された場合は内部で+1して参照します。</p> <p>[Y]<必須><数値属性参照チャンネル/演算式> 頻度解析対象数値 2 を記述します。</p>
記述例	TR2(1,64,#1,2,128,#2)
備考	戻り数値(セル)が行列構造を持つ 2 次元となり、同じ index で対象数値 1 の要素値により列番号が対象数値 2 の要素値により行番号が決定されたセルに計数します。

2.5.4. レインフロー法頻度解析関数

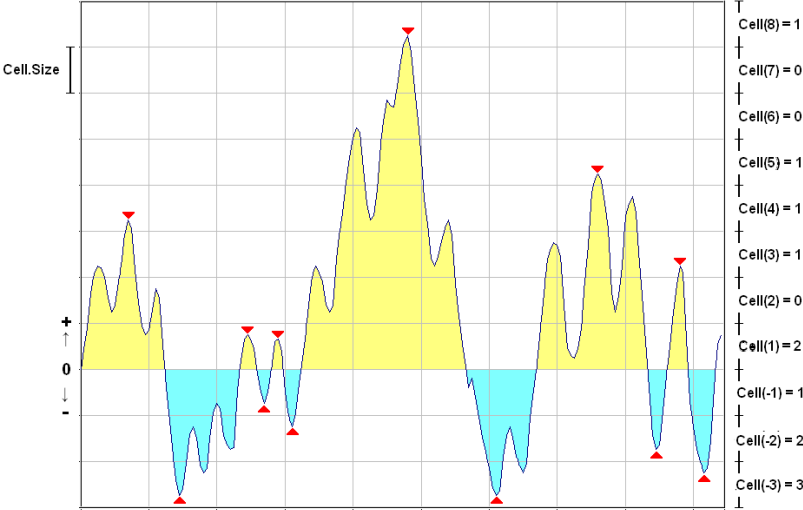
機能	レインフロー法(両振り)による頻度解析を行います。
文法	RFM(k,m,n,X)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> セルの大きさを記述します。セル間隔と同じ意味です。</p> <p>【m】<必須><即値/数値属性参照チャンネル/演算式> セルの個数を記述します。</p> <p>【n】<必須><即値/数値属性参照チャンネル/演算式> 無効振幅値をセルの何パーセントとするかを%単位で記述します。</p> <p>【X】<必須><数値属性参照チャンネル/演算式> 解析対象数列を記述します。</p>
記述例	RFM(0.2,128,63,#1)
備考	解析波形の正傾斜から負傾斜に移行した点を極大値(Peak)、負傾斜から正傾斜に移行した点を極小値(Valley)として抽出し、無効振幅除去した後、開始点から逆傾斜で開始点と等しいか過る地点間の開始点から正傾斜の場合は最大値、負傾斜の場合は最小値までを1解析区間として開始点から最大値又は最小値までの振幅と、間に存在する小ループを抽出し、ループの振幅×2を計数します。

2.5.5. 極大/極小法頻度解析関数

機能	極大/極小法による頻度解析を行います。
文法	PVM(k,m,n,X,f)
引数	<p>[k]<必須><即値/数値属性参照チャンネル/演算式> セルの大きさを記述します。セル間隔と同じ意味です。</p> <p>[m]<必須><即値/数値属性参照チャンネル/演算式> セルの個数を記述します。正負領域を持ちますので領域毎に、設定したセル个数/2 となります。なお、奇数で記述された場合は内部で+1して参照します。</p> <p>[n]<必須><即値/数値属性参照チャンネル/演算式> 無効振幅値をセルの何パーセントとするかを%単位で記述します。</p> <p>[X]<必須><数値属性参照チャンネル/演算式> 解析対象数列を記述します。</p> <p>[f]<省略可><即値/数値属性参照チャンネル/演算式> 検出した極大値極小値をそのまま計数するか、或いは、正領域は極大値、負領域は極小値を計数するかのフラグを意味します。 0又は記述省略した場合:領域に拘らず全ての極大値極小値を計数します。 1と記述した場合:正領域の極大値、負領域の極小値を計数します。</p>
記述例	PVM(1,256,63,#1)
備考	解析波形の正傾斜から負傾斜に移行した点を極大値(Peak)、負傾斜から正傾斜に移行した点を極小値(Valley)として抽出し、無効振幅除去した極大値極小値を該当するセルに計数します。



2.5.6. 最大/最小法頻度解析関数

機能	最大/最小法による頻度解析を行います。
文法	MMM(k,m,n,X)
引数	<p>文法: 格納先 = MMM(セルサイズ,セル个数,無効振幅,解析対象数列) 引数: 【k】<必須><即値/数値属性参照チャンネル/演算式> セルの大きさを記述します。セル間隔と同じ意味です。 【m】<必須><即値/数値属性参照チャンネル/演算式> セルの個数を記述します。正負領域を持ちますので領域毎に、設定したセル个数/2 となります。なお、奇数で記述された場合は内部で+1して参照します。 【n】<必須><即値/数値属性参照チャンネル/演算式> 無効振幅値をセルの何パーセントとするかを%単位で記述します。 【X】<必須><数値属性参照チャンネル/演算式> 解析対象数列を記述します。</p>
記述例	MMM(2,128,80,#1)
備考	<p>解析波形の正傾斜から負傾斜に移行した点を極大値(Peak)、負傾斜から正傾斜に移行した点を極小値(Valley)として抽出し、無効振幅除去した極大値極小値からゼロを正傾斜で過ぎてから再び負傾斜でゼロを過ぎるまでの最大値(Max)、ゼロを負傾斜で過ぎてから再び正傾斜で過ぎるまでの最小値(Min)を抽出し該当セルに計数します。解析対象波形がゼロを過ぎないと求まりません。</p> 

2. 5. 7. レベルクロス法頻度解析関数

機能	レベルクロス法による頻度解析を行います。
文法	LOR(k,m,n,X)
引数	<p>【k】<必須> セルの大きさを記述します。セル間隔と同じ意味です。</p> <p>【m】<必須> セルの個数を記述します。</p> <p>【n】<必須> 無効振幅値をセルの何パーセントとするかを%単位で記述します。</p> <p>【X】<必須> 解析対象数列を記述します。</p>
記述例	
備考	<p>解析波形の正傾斜から負傾斜に移行した点を極大値(Peak)、負傾斜から正傾斜に移行した点を極小値(Valley)として抽出し、無効振幅除去した極大値極小値から正領域は極小値から極大値が設定した Slice Level(セルの大きさ)を超えた時、負領域は極大値から極小値が Slice Level を下方に超えた時に計数します。又、Slice Level を飛び越して超えた時は、飛び越された全ての Cell に計数します。</p>

2. 5. 8. 折り畳み関数

機能	正負領域をもつ頻度解析結果を絶対値化します。
文法	FLD(X)
引数	<p>【X】<必須><数値属性参照チャンネル/演算式> 対象数列を記述します。対象数列の要素数は偶数である必要があり、奇数個の場合、実行時エラーとなります。</p>
記述例	FLD(\$1)
備考	<p>戻り数列の要素数は対象数列の 1/2 となります。</p> <p>例えば、\$1 が 6 個の要素を持つ場合、戻り数列は 3 個となり、各要素は以下の内容が格納されます。</p> <p>Ans(0) = \$1(3)+\$1(2)</p> <p>Ans(1) = \$1(4)+\$1(1)</p> <p>Ans(2) = \$1(5)+\$1(0)</p>

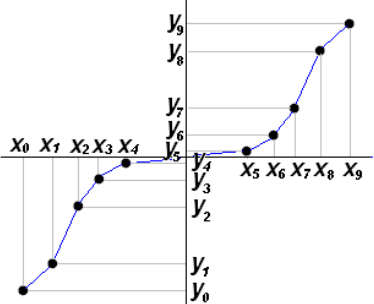
2. 6. 解析補助関数

2. 6. 1. 行列の列合計演算

機能	与えられた行列の列毎の合計を求め演算結果の列ベクトルを戻します。
文法	MCS(k,X)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> 与える行列 X の列数を記述します。 【X】<必須><数値属性参照チャネル/演算式> 演算対象行列を記述します。
記述例	MCS(20,\$1) /* \$1 は 20 列で構成された行列*/
備考	X の行数 m は $m = \text{LEN}(X)/k$ 行列の行毎の合計を求める場合は既存関数 SUM を使用して、SUM(k,X)として求めます。 なお、列合計を求める場合、転置行列変換関数を使用して対象行列を転置行列に変換して SUM 関数で求める方法と一致します。

2. 7. 解析補助関数

2. 7. 1. 非線形補正テーブル関数

機能	X のデータを、指定した部分線形テーブルファイルを使用して非線形処理した結果を戻します。
文法	TBL(テーブル名)
引数	【テーブル名】<必須><文字列即値/文字属性参照チャンネル/文字属性戻り演算式> 参照する非線形テーブルファイル名を意味します。なお、文字列即値で記述する場合は、ファイル名をダブルコーテーション""で囲んで記述します。なお、拡張子は付けません。
記述例	#1-(\$1*TBL("Nonlinear"))/#2
備考	<p>テーブル参照する場合は事前手続きが必要です。チャンネル間演算で使用する場合は、チャンネル演算 Window 上の「TABLE」をクリックして予め使用するテーブルを登録します。</p> <p>また、Script で参照する場合は、参照する前に def file_id %n “テーブル名” tbl とファイル id を定義します。なお、カレントフォルダ以外から参照する場合は、カレントフォルダをテーブルファイルが格納されているフォルダに切り替えてから、ファイル ID 定義することが必要です。</p> <p>本関数自体は演算自体を行いません。演算式中でテーブル変換したいチャンネルに掛け算します。ただし、演算式中の記述則に制限があり、本関数の前に、先行して非線形補正するデータ列、または演算式を記述し、演算子"*"で接続されている必要があります。</p> <p>\$1*TBL("Nonlinear")</p> <p>と記述された場合、数値属性参照チャンネル\$1 は、テーブルファイル名 NonLinear.tbl ファイルに記載されている部分線形テーブルにより非線形補正されます。なお、テーブル補正は補正対象データを X と見なし、テーブル内に X が存在する範囲であれば直線補間し、テーブル範囲外であれば直線補外され Y 値を戻します。</p>  <p>テーブルファイルは、拡張子 ".tbl" のテキスト形式で、カレントフォルダに格納されている必要があります。</p> <p>テーブルファイルフォーマット:</p> <p>1 行目 PWF_TABLE ← keyword 固定です。</p> <p>2 行目 PAIR=10 ← 記述するテーブルペア数を記述</p> <p>3 行目以降 X 軸値,Y 軸値 ← 半角カンマで","区切り、X 軸と Y 軸のペアを記述します。</p> <p>なお、最大ペア数は 30 迄となり、また、X 軸値昇順に記述します。</p>

2. 7. 2. 1,2,5 刻み数列生成関数

機能	最大値と最小値及び分割数を与え、分割数の最も内輪で近い 1,2,5 ステップでの数列を戻します。
文法	ATS(k,n,m)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> 分割数の上限値を記述します。記述する値は正整数の必要があります。小数点以下を含む場合は小数点以下を切り捨てて参照します。但し、負数の場合は実行時 Error となります。</p> <p>分割数とは後述する引数最大値-最小値を分割する分割数-1 を意味します。なお、複数要素を持つ数列で記述しても Index0 のみ参照されます。</p> <p>【n】<必須><即値/数値属性参照チャンネル/演算式> 最小値を記述します。なお、複数要素を持つ数列で記述しても Index0 のみ参照されます。</p> <p>【m】<必須><即値/数値属性参照チャンネル/演算式> 最大値を記述します。なお、複数要素を持つ数列で記述しても Index0 のみ参照されます。 記述する最大値と最小値の関係は最小値<最大値でなければなりません。最小値>最大値の場合は実行時 Error となります。</p>
記述例	ATS(6,\$1,\$2)
備考	<p>グラフ描画などのグリッド線位置を求める場合などに使用します。</p> <p>戻り値は最小値と最大値内の 1,2,5 ステップで刻まれた数列が戻ります。</p> <p>例1: \$1 = ATS(5,-13.5,120)とした場合、\$1→0.50,100 と戻ります。</p> <p>刻み幅を求める場合は戻り値を、差分関数を使用して DIF(ATS(k,n,m,i))で求めます。</p>

2.7.3. R10 系列 Index 変換関数

機能	与えた周波数を R10 系列 Index に変換します。
文法	FTI(X)
引数	【X】<必須><即値/数値属性参照チャネル/演算式> 変換対象周波数数列を記述します。負数で記述された場合は実行時 Error となります。 戻り Index = $3 * \text{LOG}(\text{Freq}/1000) / \text{LOG}(2)$
記述例	FTI(\$1)
備考	戻り値は小数点以下を含みます。 例えば 1024Hz を R10Index に変換すると、0.102647146 と戻ります。

2.7.4. R10 系列 Index 取得関数

機能	最低周波数と最高周波数を与え、その範囲を含む R10 系列 Index 番号を戻します。
文法	RTI(k,m)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> 最低周波数を単位"Hz"で記述します。数列で記述しても参照される Index は 0 のみとなります。 【m】<必須><即値/数値属性参照チャネル/演算式> 最高周波数を単位"Hz"で記述します。数列で記述しても参照される Index は 0 のみとなります。
記述例	RTI(70,1500)
備考	R10 周波数系列は ISO266:1997[5]に規定された R10 系列を意味します。 戻り数列は、最低周波数から最高周波数を含む完全ベース 2 での Index 列となります。 演算は、与えられた周波数から Index を $\text{Index} = 3 * \text{LOG}(\text{Freq}/1000) / \text{LOG}(2)$ で求め、最低周波数の場合は小数点以下切り捨て、最大周波数の場合は小数点以下切り上げ処理を行い、最低周波数 Index から最大周波数 Index 迄をインクリメントした数列を戻します。 \$1 = RTI(70,1500) \$1 → -11,-10,-9,-8,-7,-6,-4,-3,-2,-1,0,1,2 ※ 最低周波数 70Hz の Index は -11.5095 から -11、最高周波数 1500Hz の Index は 1.75488 から 2 となり、戻り数列は -12,-11,-10,-9,-8,-7,-6,-4,-3,-2,-1,0,1,2 となります。 なお、R10 系列 Index から周波数への変換を既存の関数で示すと \$2 "Freq(R10):Hz" = $2^{(\text{RTI}(79,1500)/3)*1000}$ となります。但し、ここで Index から求めた周波数は完全ベース 2 周波数と呼ばれ、表示上の公称周波数と異なります。Index から公称周波数を求める場合、演算関数:R10 公称周波数変換関数 RTF(k)を使用して行います。

R10_INDEX	R10周波数(Hz)
-20	9.9431
-19	12.4016
-18	15.6250
-17	19.6863
-16	24.8031
-15	31.2500
-14	39.3725
-13	49.6063
-12	62.5000
-11	78.7451
-10	99.2126
-9	125.0000
-8	157.4901
-7	198.4251
-6	250.0000
-5	314.9803
-4	396.0503
-3	500.0000
-2	629.9605
-1	793.7005
0	1000.0000
1	1259.9210
2	1587.4011
3	2000.0000
4	2519.8421
5	3174.8021
6	4000.0000
7	5039.6842
8	6349.6042
9	8000.0000

2.7.5. R10 公称周波数変換関数

機能	R10 系列 Index 数列から公称周波数数列に変換します。
文法	RTF(k)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> R10 系列 Index を記述します。複数要素を持つ数列で記述した場合、記述される R10 系列 Index の並び順は拘りません。 なお、記述した Index に小数点以下を含む場合、四捨五入され整数化して参照されます。
記述例	RTF(\$1)
備考	\$1 ← -3,-2,-1,0,1,2,3 \$2 = RTF(\$1) \$2 → 500,630,800,1000,1250,1600,2000

2. 8. 論理演算関数

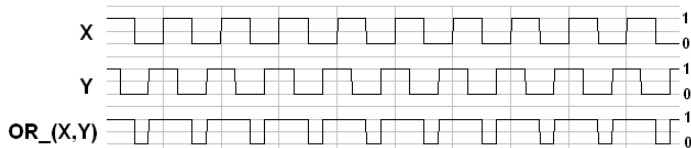
2. 8. 1. 論理積関数

機能	X,Y の論理積演算結果を戻します。															
文法	AND(X,Y)															
引数	<p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。なお、X のデータは 1 または 0 でなければなりません。</p> <p>【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。なお、Y のデータは"1"または"0"でなければなりません。</p>															
記述例	AND(\$1,\$2)															
備考	<p>演算結果データ個数は、X または Y の何れか少ない方のデータ個数となります。</p> <p>論理積：</p> <table border="1"> <thead> <tr> <th>X</th> <th>Y</th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> <p> </p>	X	Y	結果	0	0	0	1	0	0	0	1	0	1	1	1
X	Y	結果														
0	0	0														
1	0	0														
0	1	0														
1	1	1														

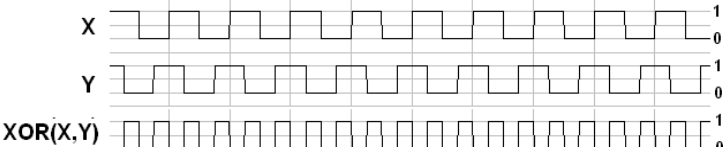
2. 8. 2. 否定論理積関数

機能	X,Y の否定論理積演算結果を戻します。															
文法	NAD(X,Y)															
引数	<p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。なお、X のデータは 1 または 0 でなければなりません。</p> <p>【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。なお、Y のデータは"1"または"0"でなければなりません。</p>															
記述例	NAD(\$1,\$2)															
備考	<p>演算結果データ個数は、X または Y の何れか少ない方のデータ個数となります。</p> <p>否定論理積：</p> <table border="1"> <thead> <tr> <th>X</th> <th>Y</th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table> <p> </p>	X	Y	結果	0	0	1	1	0	1	0	1	1	1	1	0
X	Y	結果														
0	0	1														
1	0	1														
0	1	1														
1	1	0														

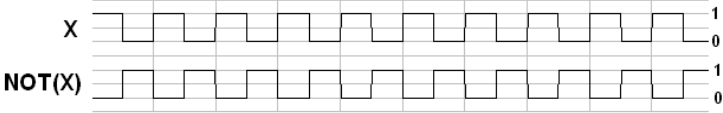
2. 8. 3. 論理和関数

機能	X,Y の論理和演算結果を戻します。															
文法	OR(X,Y)															
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。なお、X のデータは 1 または 0 でなければなりません。 【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。なお、Y のデータは"1"または"0"でなければなりません。															
記述例	OR(\$1,\$2)															
備考	演算結果データ個数は、X または Y の何れか少ない方のデータ個数となります。 論理和: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>X</th> <th>Y</th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> 	X	Y	結果	0	0	0	1	0	1	0	1	1	1	1	1
X	Y	結果														
0	0	0														
1	0	1														
0	1	1														
1	1	1														

2. 8. 4. 排他的論理和関数

機能	X,Y の排他的論理和演算結果を戻します。															
文法	XOR(X,Y)															
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。なお、X のデータは 1 または 0 でなければなりません。 【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。なお、Y のデータは"1"または"0"でなければなりません。															
記述例	XOR(\$1,\$2)															
備考	演算結果データ個数は、X または Y の何れか少ない方のデータ個数となります。 排他的論理和: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>X</th> <th>Y</th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table> 	X	Y	結果	0	0	0	1	0	1	0	1	1	1	1	0
X	Y	結果														
0	0	0														
1	0	1														
0	1	1														
1	1	0														

2. 8. 5. 否定論理関数

機能	X の論理を反転結果した結果を戻します。						
文法	NOT(X)						
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。なお、X のデータは"1"または"0"でなければなりません。						
記述例	NOT(\$1)						
備考	<table border="1" style="margin-left: 20px;"> <thead> <tr> <th>X</th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table> 	X	結果	0	1	1	0
X	結果						
0	1						
1	0						

2. 8. 6. 部分否定論理関数

機能	X の論理データから、指定したデータ個数分 k で指定した論理が連続しない部分を反転した結果を戻します。
文法	ECP(k,m,X)
引数	<p>【k】<必須><即値/数値属性参照チャンネル> 判定する論理値を記述します。</p> <p>【m】<必須><即値/数値属性参照チャンネル> k で指定した論理が連続するデータ個数を記述します。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象論理値数列を記述します。X は論理"1"または"0"で構成される論理値数列で、X のデータ個数は、X>=m の必要がありません。</p>
記述例	ECP(1,20,\$1)
備考	<p>演算対象数列から k で指定した論理値が m で指定した個数連続しない部分を論理反転した結果を戻します。</p>

2. 8. 7. X=Y 比較演算関数

機能	比較演算 X=Y を行い結果の論理値または成立した Index を戻します。												
文法	EQU(X,Y,k)												
引数	<p>【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較対象数列を記述します。なお、X のデータ数は、1 個または Y のデータ数と等しくなくてはなりません。</p> <p>【Y】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較数列を記述します。なお、Y のデータ数は、1 個または X のデータ数と等しくなくてはなりません。 X,Y 何れかの要素が 1 個の場合は、比較演算対象の全ての要素と比較され、要素数が等しい場合は要素毎に比較されます。</p> <p>【k】<省略可><即値/収録チャンネル/数値属性参照チャンネル/演算式> 論理値を戻すか、成立した Index(データ番号)戻すかのフラグを記述します。 k=0 の時: 記述省略された場合は、k=0 と等価となります 論理値を戻します。 k<>0 の時: 成立した Index を戻します。条件成立が存在しない場合は、-1 が戻ります。</p>												
記述例	EQU(\$1,\$2)												
備考	<table border="1"> <thead> <tr> <th></th> <th>比較</th> <th></th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>=</td> <td>Y</td> <td>1</td> </tr> <tr> <td>X</td> <td><></td> <td>Y</td> <td>0</td> </tr> </tbody> </table>		比較		結果	X	=	Y	1	X	<>	Y	0
	比較		結果										
X	=	Y	1										
X	<>	Y	0										

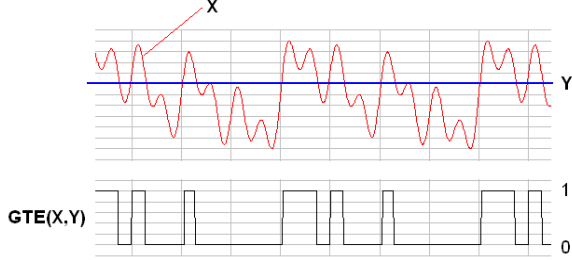
2. 8. 8. X<>Y 比較演算関数

機能	比較演算 X<>Y を行い結果の論理値または成立した Index を戻します。														
文法	NEQ(X,Y,k)														
引数	<p>【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較対象数列を記述します。なお、X のデータ数は、1 個または Y のデータ数と等しくなくてはなりません。</p> <p>【Y】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較数列を記述します。なお、Y のデータ数は、1 個または X のデータ数と等しくなくてはなりません。 X,Y 何れかの要素が 1 個の場合は、比較演算対象の全ての要素と比較され、要素数が等しい場合は要素毎に比較されます。</p> <p>【k】<省略可><即値/収録チャンネル/数値属性参照チャンネル/演算式> 論理値を戻すか、成立した Index(データ番号)戻すかのフラグを記述します。 k=0 の時:記述省略された場合は、k=0 と等価となります 論理値を戻します。 k<>0 の時: 成立した Index を戻します。条件成立が存在しない場合は、-1 が戻ります。</p>														
記述例	NEQ(#1,#2)														
備考	<table border="1"> <thead> <tr> <th></th> <th>比較</th> <th></th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>=</td> <td>Y</td> <td>0</td> </tr> <tr> <td>X</td> <td><></td> <td>Y</td> <td>1</td> </tr> </tbody> </table>				比較		結果	X	=	Y	0	X	<>	Y	1
	比較		結果												
X	=	Y	0												
X	<>	Y	1												

2. 8. 9. X>Y 比較演算関数

機能	比較演算 X>Y を行い結果の論理値または成立した Index を戻します。														
文法	GT_(X,Y,k)														
引数	<p>【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較対象数列を記述します。なお、X のデータ数は、1 個または Y のデータ数と等しくなくてはなりません。</p> <p>【Y】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較数列を記述します。なお、Y のデータ数は、1 個または X のデータ数と等しくなくてはなりません。 X,Y 何れかの要素が 1 個の場合は、比較演算対象の全ての要素と比較され、要素数が等しい場合は要素毎に比較されます。</p> <p>【k】<省略可><即値/収録チャンネル/数値属性参照チャンネル/演算式> 論理値を戻すか、成立した Index(データ番号)戻すかのフラグを記述します。 k=0 の時:記述省略された場合は、k=0 と等価となります 論理値を戻します。 k<>0 の時: 成立した Index を戻します。条件成立が存在しない場合は、-1 が戻ります。</p>														
記述例	GT_(#1,#2)														
備考	<table border="1"> <thead> <tr> <th></th> <th>比較</th> <th></th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>></td> <td>Y</td> <td>0</td> </tr> <tr> <td>X</td> <td><=</td> <td>Y</td> <td>1</td> </tr> </tbody> </table>				比較		結果	X	>	Y	0	X	<=	Y	1
	比較		結果												
X	>	Y	0												
X	<=	Y	1												

2. 8. 10. X>=Y 比較演算関数

機能	比較演算 $X \geq Y$ を行い結果の論理値または成立した Index を戻します。																
文法	GTE(X,Y,k)																
引数	<p>【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較対象数列を記述します。なお、X のデータ数は、1 個または Y のデータ数と等しくなくてはなりません。</p> <p>【Y】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較数列を記述します。なお、Y のデータ数は、1 個または X のデータ数と等しくなくてはなりません。 X,Y 何れかの要素が 1 個の場合は、比較演算対象の全ての要素と比較され、要素数が等しい場合は要素毎に比較されます。</p> <p>【k】<省略可><即値/収録チャンネル/数値属性参照チャンネル/演算式> 論理値を戻すか、成立した Index(データ番号)戻すかのフラグを記述します。 k=0 の時:記述省略された場合は、k=0 と等価となります 論理値を戻します。 k>0 の時: 成立した Index を戻します。条件成立が存在しない場合は、-1 が戻ります。</p>																
記述例	GTE(\$1,.2)																
備考	<table border="1"> <thead> <tr> <th></th> <th>比較</th> <th></th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>></td> <td>Y</td> <td>1</td> </tr> <tr> <td>X</td> <td>=</td> <td>Y</td> <td>1</td> </tr> <tr> <td>X</td> <td><</td> <td>Y</td> <td>0</td> </tr> </tbody> </table>  <p>関数の引数 X と Y を入れ替えると、次の意味となります。 $GTE(X,Y) \Rightarrow X \geq Y$ $GTE(Y,X) \Rightarrow X \leq Y$</p>		比較		結果	X	>	Y	1	X	=	Y	1	X	<	Y	0
	比較		結果														
X	>	Y	1														
X	=	Y	1														
X	<	Y	0														

2. 8. 11. X<Y 比較演算関数

機能	比較演算 $X < Y$ を行い結果の論理値または成立した Index を戻します。												
文法	LT_(X,Y,k)												
引数	<p>【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較対象数列を記述します。なお、X のデータ数は、1 個または Y のデータ数と等しくなくてはなりません。</p> <p>【Y】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較数列を記述します。なお、Y のデータ数は、1 個または X のデータ数と等しくなくてはなりません。 X,Y 何れかの要素が 1 個の場合は、比較演算対象の全ての要素と比較され、要素数が等しい場合は要素毎に比較されます。</p> <p>【k】<省略可><即値/収録チャンネル/数値属性参照チャンネル/演算式> 論理値を戻すか、成立した Index(データ番号)戻すかのフラグを記述します。 k=0 の時:記述省略された場合は、k=0 と等価となります 論理値を戻します。 k>0 の時: 成立した Index を戻します。条件成立が存在しない場合は、-1 が戻ります。</p>												
記述例	LT_(#1,#2)												
備考	<table border="1"> <thead> <tr> <th></th> <th>比較</th> <th></th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>X</td> <td><</td> <td>Y</td> <td>1</td> </tr> <tr> <td>X</td> <td>>=</td> <td>Y</td> <td>0</td> </tr> </tbody> </table>		比較		結果	X	<	Y	1	X	>=	Y	0
	比較		結果										
X	<	Y	1										
X	>=	Y	0										

2. 8. 12. X<=Y 比較演算関数

機能	比較演算 $X \leq Y$ を行い結果の論理値または成立した Index を戻します。																
文法	LTE(X,Y)																
引数	<p>【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較対象数列を記述します。X のデータ数は 1 個、または Y のデータ数と等しくなくてはなりません。</p> <p>【Y】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 比較数列を記述します。Y のデータ数は 1 個、または X のデータ数と等しくなくてはなりません。 X,Y 何れかの要素が 1 個の場合は、対象数列の全ての要素と比較され、要素数が等しい場合は要素毎に比較されます。</p> <p>【k】<省略可><即値/収録チャンネル/数値属性参照チャンネル/演算式> 論理値を戻すか、成立した Index(データ番号)戻すかのフラグを記述します。 k=0 の時:記述省略された場合は、k=0 と等価となります 論理値を戻します。 k>0 の時: 成立した Index を戻します。条件成立が存在しない場合は、-1 が戻ります。</p>																
記述例	LTE(#2,\$1)																
備考	<table border="1"> <thead> <tr> <th></th> <th>比較</th> <th></th> <th>結果</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>></td> <td>Y</td> <td>0</td> </tr> <tr> <td>X</td> <td>=</td> <td>Y</td> <td>1</td> </tr> <tr> <td>X</td> <td><</td> <td>Y</td> <td>1</td> </tr> </tbody> </table> <p>関数の引数 X と Y を入れ替えると、次の意味となります。 $LTE(X,Y) \Rightarrow X \leq Y$ $LTE(Y,X) \Rightarrow X \geq Y$</p>		比較		結果	X	>	Y	0	X	=	Y	1	X	<	Y	1
	比較		結果														
X	>	Y	0														
X	=	Y	1														
X	<	Y	1														

2. 8. 13. 論理値化関数

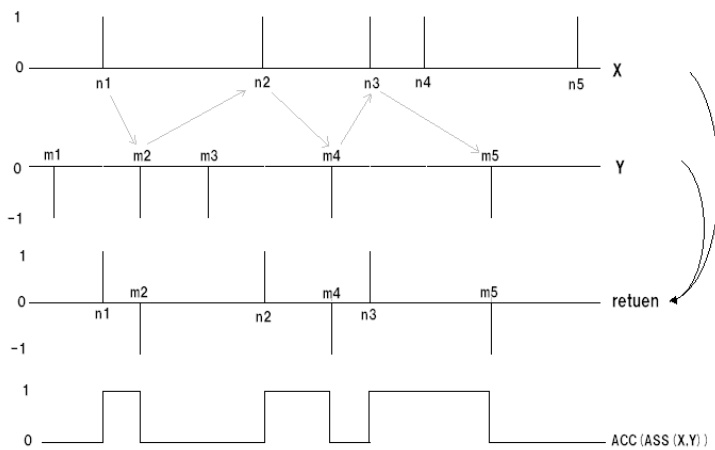
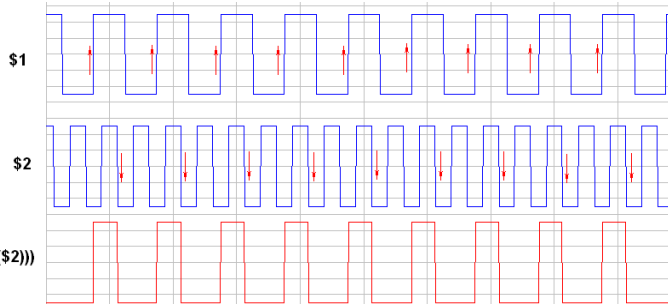
機能	X を "1" と "0" から成る論理値数列を戻します。
文法	RWC(k,m,X)
引数	<p>【k】<必須><即値/数値属性参照チャンネル> 論理"1"の閾値を意味し、即値、または先頭文字"\$"に続きチャンネル番号で表す数値属性参照チャンネルで記述します。</p> <p>【m】<必須><即値/数値属性参照チャンネル> 論理"0"の閾値を意味し、即値、または先頭文字"\$"に続きチャンネル番号で表す数値属性参照チャンネルで記述します。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象データで、先頭文字"#"に続きチャンネル番号で表す収録チャンネル、先頭文字"\$"に続きチャンネル番号で表す数値属性参照チャンネル、または演算式で記述します。</p>
記述例	RWC(\$1,\$2,#1)
備考	<p>演算対象データ列のデータが、k を上昇でよぎってから、m を下降でよぎるまでを、論理値"1"とし、同様に m を下降でよぎってから k を上昇でよぎるまでを、論理値"0"とした論理波形を生成します。 ただし、解析範囲開始点から k を上昇で過るか、m を下降で過るまでの間、 $X(n) > m$ の時は論理値"1"とし、$X(n) <= m$ の時は論理"0"と、します。</p>

2. 8. 14. Bit 論理関数

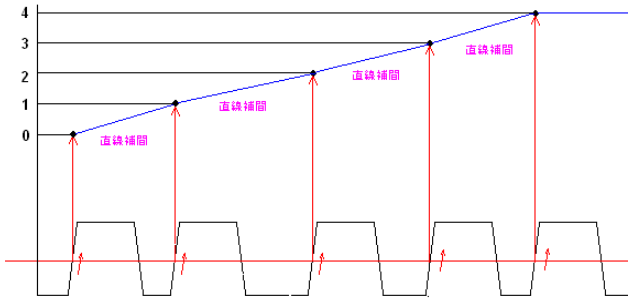
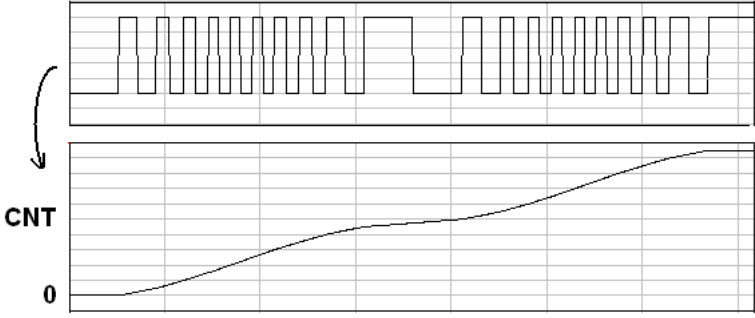
機能	X を 2Byte 整数化し、Bit 位置の論理を戻します。
文法	BIT(k,X)
引数	<p>【k】<必須><即値/数値属性参照チャネル> 抽出する Bit 位置を記述します。なお、記述する Bit 位置は $15 \geq k \geq 0$ の範囲でなければなりません。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。なお、X は -32768 ~ 32767 の 2Byte 整数範囲でなければなりません。</p>
記述例	BIT(3,#2)
備考	演算対象データを 2byte 整数データに縮退し、k で指定された bit 位置の論理を戻します。

2. 9. パルス列取扱関数

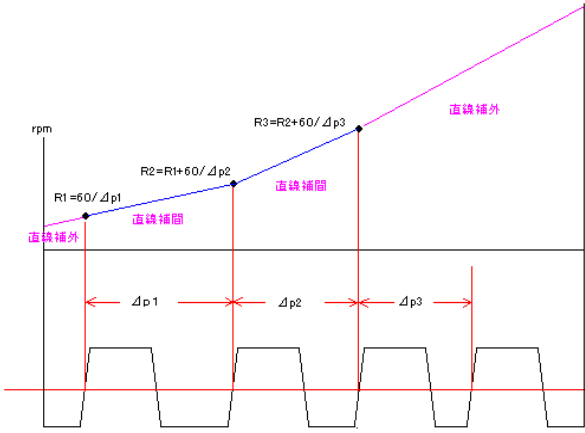
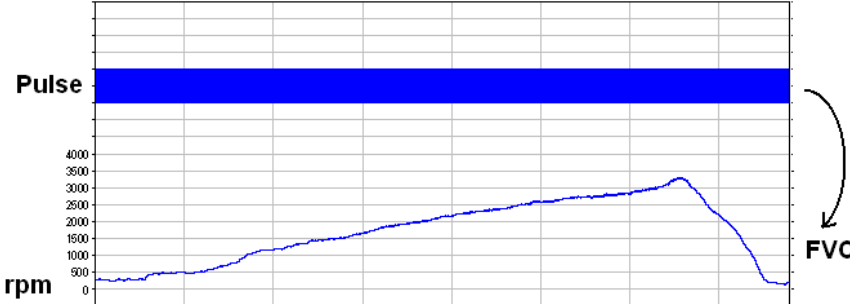
2. 9. 1. パルス波形差分値合成関数

機能	X に解析開始地点を立ち上がりパルス波形の差分値を、Y に解析終了地点を立下りとしたパルス波形の差分値を与えることで、解析開始地点と解析終了地点で構成されたパルス波形差分値を戻します。
文法	ASS(X,Y)
引数	<p>【X】<必須><即値/数値属性参照チャネル/演算式> 0 と 1 からなる数列を記述します。なお、0 と 1 以外で構成されている場合は、$X_n > 0$ を 1 とし $X_n < 0$ を 0 として扱います。</p> <p>【Y】<必須><数値属性参照チャネル/演算式> 0 と -1 からなる数列を記述します。なお、0 と -1 以外で構成されている場合は、$Y_m >= 0$ を 0 とし $Y_m < 0$ を -1 として扱います。</p>
記述例	ASS(\$1,\$2)
備考	<p>戻り数列(初期値は全て 0)に、X で与えられた数列を Index 昇順方向に"1"を検索して"1"が存在した Index 位置に"1"を代入し、次に Y で与えられた数列を現在の Index 位置+1 より Index 昇順方向に"-1"を検索し、戻り数列に"-1"が存在した Index 位置に"-1"を代入します。再び X で与えられた数列を現在の Index 位置+1 より Index 昇順方向に"1"を検索し、戻り数列に"1"が存在した Index 位置に"1"を代入する操作を繰り返します。ただし、X で与えられた数列で"1"が検索されても、対応する Y で与えられた数列に"-1"が検出されない場合は、戻り数列の代入した"1"位置を 0 に戻します。つまり、"1"の個数と"-1"の個数は必ず等しくなります。</p>  <p>戻り数列を累積演算すると、パルス波形となります。X、Y ともパルス列を与え、結果をパルス列で求める例を示します。</p> 

2. 9. 2. パルス計数関数

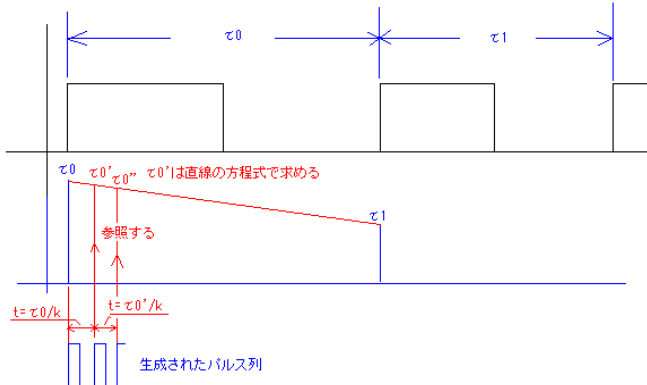
機能	パルス列の平均値を正傾斜で過ってから再び正傾斜で過った回数を戻します。
文法	CNT(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。演算対象数列はパルス列でなくても良く、閾値は平均値をとります。
記述例	CNT(\$1)
備考	<p>演算対象データの平均値を求め、平均値を上昇で過ぎてから再び上昇で過る回数を戻します。結果データ個数と演算対象データ個数を一致させる為に、上昇で過ぎてから再び上昇で過るまでの間は直線補間して戻します。</p>  <p>F-V 関数と類似していますが、異なる点は、F-V 関数が周期を求めているのに対し、平均値を上昇で過ぎた回数を求めている点です。また、計数地点は、周期を検出した時点で行い、演算対象データの開始から最初の周期検出までは 0、終端は最終値を維持します。</p> 

2. 9. 3. パルス周波数関数

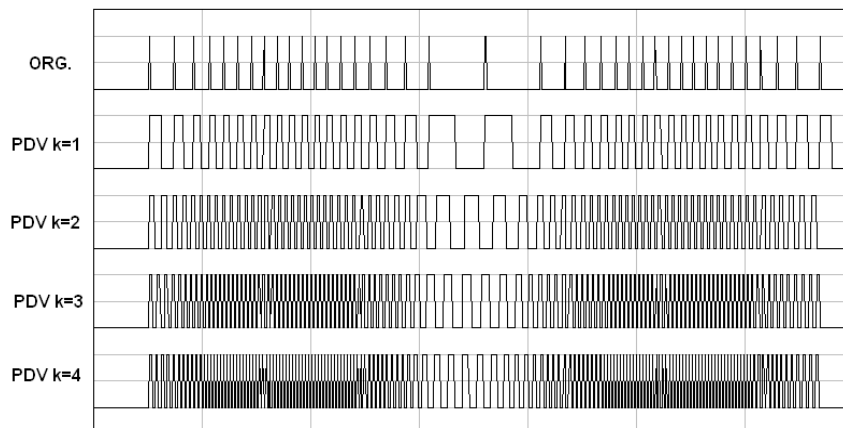
機能	Xの周期を求め、回転数に変換した結果を戻します。周波数を求める場合は結果を60で割ること求めます。
文法	FVC(k,X,m)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 1回転当たりのパルス数を記述します。なお、kは、$1 \leq k$ で、かつ整数でなければなりません。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象パルス列を記述します。</p> <p>【m】<省略可><即値/数値属性参照チャネル/演算式> 周期終止時間を単位 sec で記述します。 周期終止時間とは、検出したパルス周期がここで記述した周期終止時間(sec)より長い場合、回転数ゼロとして扱うことを意味します。記述省略された場合は、次のパルスの立ち上がりまでを直線補間します。</p>
記述例	FVC(2,#6)
備考	<p>パルス列の周期を演算し、直前の値に加算します。同時に指定された1回転当たりパルス数kを使用して回転数データrpmに変換した値を戻します。なお、補外処理区間は、直前または直後の傾斜で0rpmまで伸ばし、0rpmに達した後は0としています。</p>  <p>例えば、4サイクルエンジン、6気筒の1次側パルスは、120度で1パルス、従って1回転で3パルス存在します。この場合、k=3と設定します。パルス間の周期を計算して回転数に変換されます。 与えるパルス列の閾値はそのパルス列の平均値が採られ、閾値を正傾斜で過ぎてから再び正傾斜で過ぎるまでを周期と見なし直線補間処理にて回転数に変換されます。また、最初のパルス間での回転数および、最後のパルスが存在しないときのいずれも、直近の直線の延長線にあるものとして直線補外演算により求めます。 1回転当たり1パルス列から回転数を求める FVC(1,#6) #6:=パルスチャネル</p>  <p>パルス列の周期を求める場合、1回転当たりパルス数を1とし、60で割ることにより周波数が得られ、その逆数とすることで周期が求まります。ただし、回転数が0rpm、つまり0Hzを含む場合は、逆数演算時に0Hzを除外する処理が必要です。(下制限関数GUN(k,X))により制限させる等)</p>

2. 9. 4. パルス逡倍関数

機能	X のパルス列を逡倍したパルス列を戻します。
文法	PDV(k,X)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> パルスの逡倍数を記述します。なお、k は、$k > 0$ でなければなりません。 最小パルス周期が演算対象データのサンプリング周期の 2 倍以上でなければなりません。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象論理数列を記述します。</p>
記述例	PDV(5,#2)
備考	演算対象パルス列のパルス間を k で設定した逡倍数で分割した新たなパルス列を生成します。なお、内挿するパルス周期変化は元パルス列周期の直線補間処理によります。



演算処理対象データのパルス周期が、逡倍可能であるか否かを知る為には、FVC 関数を用いて $K < (PRD() / (MAX(FVC(1, 演算対象データ) / 60))) / 2$ が成立しているかを確認します。



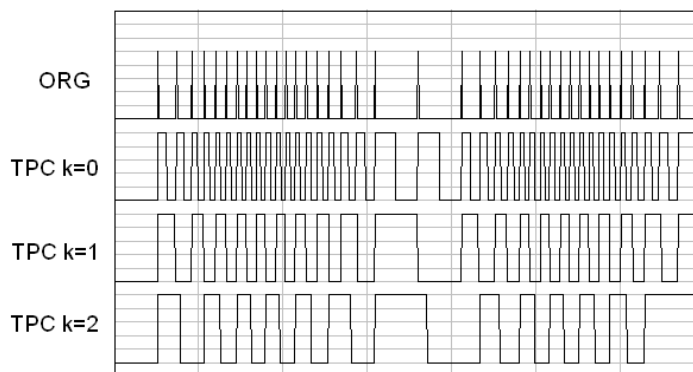
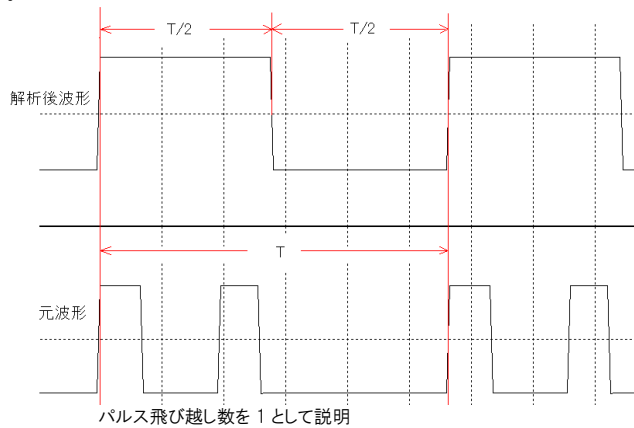
TCP 関数(パルス間引き整形関数)で飛び越し数 $k=0$ と指定した場合と、PDV 関数で逡倍数 $k=1$ と指定した場合、同じ波形が得られますが、演算対象数列の終端で周期が検出されない場合、TCP 関数は High または Low を維持しますが、PDV 関数では最終パルスの幅を直前のパルス幅と同じとして Low に戻します。

2.9.5. 回転パルス生成関数

機能	X から回転パルス列を戻します。
文法	RTP(k,X)
引数	【k】<必須><即値/数値属性参照チャンネル/演算式> 1 回転当たりのパルス数を記述します。なお、k は、 $K > 0$ でなければなりません。 【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。なお、X の単位は rpm と見なします。
記述例	RTP(120,#3)
備考	演算対象数列から設定された1回転当たりのパルス数kを使用して、回転パルスを生成しますが、生成される最小パルス周期がサンプリング周期の2倍以上である必要があります。

2.9.6. パルス間引き整形関数

機能	X のパルス列を間引きし、デューティ比を 50%に整合します。
文法	TPC(k,X)
引数	【k】<必須><即値/数値属性参照チャンネル/演算式> パルスの飛び越し数を記述します。 【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	TPC(1,#6)
備考	演算対象パルス列を k で指定された飛び越し数に従って有効パルスを抽出した後、デューティ比を 50%に整合し、パルス High 側の値を 1、Low 側の値を 0 とした論理値パルス列に置き換えます。なお、演算対象パルス列の閾値は解析チャンネルの平均値をとります。

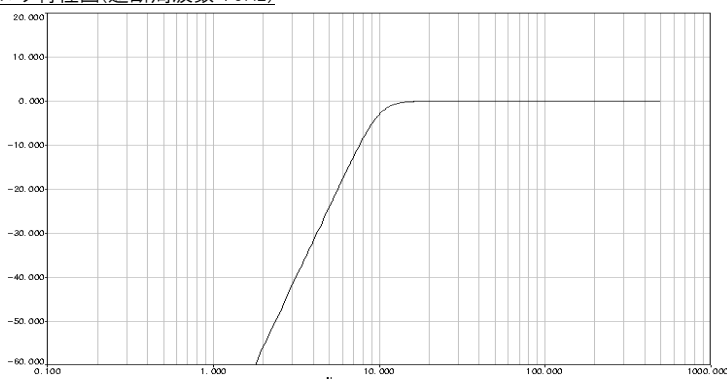


2. 10. フィルタ関数

2. 10. 1. 4次バターワース・ハイパスフィルタ関数

機能	X の 4 次バターワース IIR 形式ハイパスフィルタ処理結果を戻します
文法	HPF(k,X) → ハイパスフィルタ関数 HPR(k,X) → ハイパスフィルタ関数(時間軸逆)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> 遮断周波数を記述します。遮断周波数での減衰量は-3dB です。ただし、k は、 $0 < k < \text{サンプリング周波数}/4$ の範囲です。 【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。
記述例	HPF(20,#1) HPR(20,#5) HPF(20,HPR(20,#1)) → HPF 関数と HPR 関数の組み合わせにより、位相遅れなし-48dB/oct フィルタを構成します。
備考	遮断特性は-24dB/oct に固定されています。 HPF は時刻経過方向にフィルタ処理し、HPR は解析範囲の終了点から時間を遡る方向にフィルタ処理します。遮断周波数(k)と最低振幅平坦周波数の関係は、最低振幅平坦周波数= $k \times 2.13199$ となります。 ハイパスフィルタ特性図(遮断周波数 10Hz)

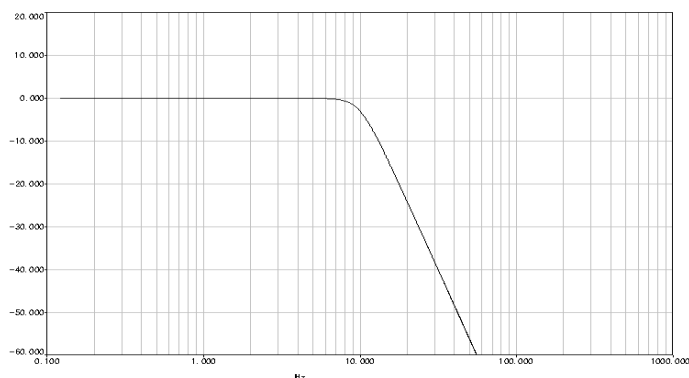
ハイパスフィルタ特性図(遮断周波数 10Hz)



2. 10. 2. 4次バターワース・ローパスフィルタ関数

機能	X の 4 次バターワース IIR 形式ローパスフィルタ処理結果を戻します。
文法	LPF(k,X) → ローパスフィルタ関数 LPR(k,X) → ローパスフィルタ関数(時間軸逆)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> 遮断周波数を記述します。遮断周波数での減衰量は-3dB です。ただし、k は、 $0 < k < \text{サンプリング周波数}/4$ の範囲となります。 【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。
記述例	LPF(20,#1) LPR(20,#5)
備考	遮断特性は-24dB/oct に固定されています。 LPF は時刻経過方向にフィルタ処理し、LPR は解析範囲の終了点から時間を遡る方向にフィルタ処理します。遮断周波数(k)と最大振幅平坦周波数の関係は、最大振幅平坦周波数= $k \times 0.46807$ となります。 ローパスフィルタ特性図(遮断周波数 10Hz)

ローパスフィルタ特性図(遮断周波数 10Hz)



2. 10. 3. 積分フィルタ

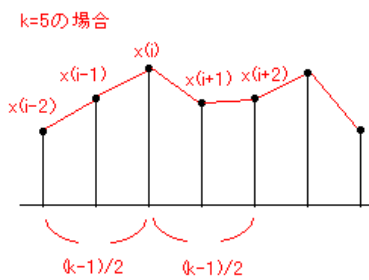
機能	アナログ積分アンプでの特性をエミュレートして X の積分結果(IIR 形式-6dB/oct フィルタ処理)を戻します。
文法	LBF(k,X)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> 遮断周波数を意味し、即値、先頭文字"\$"に続きチャンネル番号で表す数値属性参照チャンネル、または演算式で記述します。遮断周波数での減衰量は-3dB です。ただし、k は、$0 < k < \text{サンプリング周波数}/4$ の範囲です。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	LBF(2,#2)
備考	<p>振動波形等を-6dB/oct フィルタの減衰特性を使用して積分する時に使用します。正確に積分される周波数領域は-6dB/oct で傾斜している周波数範囲であり、直流領域まで積分しませんので積分関数とは結果が異なります。</p> <p>フィルタ特性図(遮断周波数 1Hz)</p>

2. 10. 4. 6 次チェビシェフ・ローパスフィルタ関数

機能	X の 6 次チェビシェフ IIR 形式ローパスフィルタ処理結果を戻します
文法	LCF(k,X)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> 遮断周波数を記述します。遮断周波数での減衰量は-3dB です。ただし、k は、$0 < k < \text{サンプリング周波数}/4$ の範囲です。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	LCF(2,#2)
備考	<p>サンプリング周波数の変更処理等、急峻な遮断特性を必要としている場合に使用します。</p> <p>フィルタ特性図(遮断周波数 10Hz)</p>

2. 10. 5. 移動平均関数

機能	X を指定した平均化個数で移動平均値を戻します。
文法	MAV(k,X)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 平均化するデータ個数を記述します。記述した k が複数の要素で構成されていても参照される値は Index0 の値のみとなります。 なお、k は k>0 でなくてはなりません。また小数点以下を含む場合は小数点以下が切り捨てられて参照されます。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。なお、X のデータ個数は、>k でなくてはなりません。</p>
記述例	MAV(50,#1)
備考	平均化個数は両翼を等しい数で平均化するため、偶数が設定された場合は+1 され奇数化されて使用されます。なお、移動平均結果のデータ数を平均前のデータ数と変らない様にするため、平均開始から k で設定された平均個数の 1/2 までは順次両翼を延ばし、1/2 個数に達してからは k 個の平均化を行い、同様にデータ列の終端(k-1)/2 より順次平均化個数を減じて行く演算方法を採用しています。



$$a_0 = x_0$$

$$a_1 = \frac{x_0 + x_1 + x_2}{3}$$

$$a_2 = \frac{x_0 + x_1 + x_2 + x_3 + x_4}{5}$$

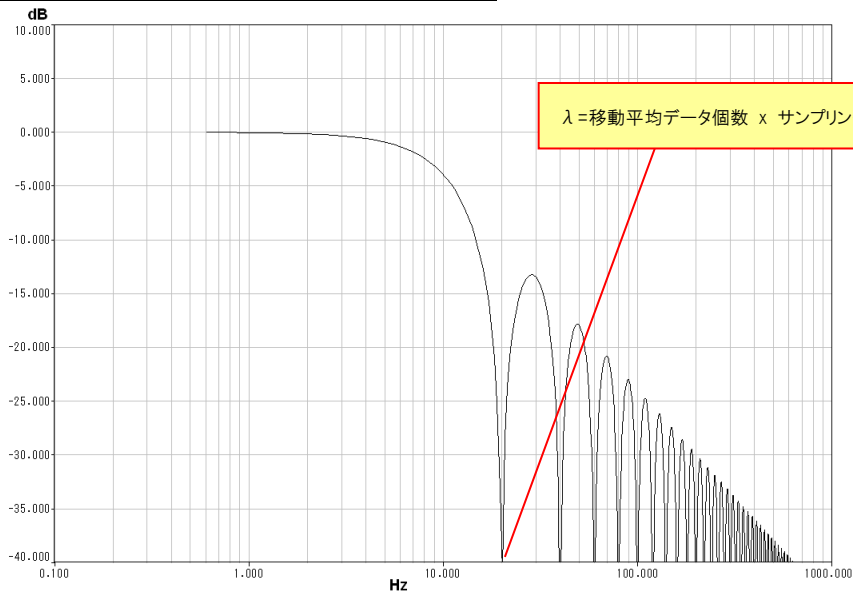
$$a_j = \frac{x_{j-2} + x_{j-1} + x_j + x_{j+1} + x_{j+2}}{5}$$

$$a_{n-1} = \frac{x_{n-2} + x_{n-1} + x_n}{3}$$

$$a_n = x_n$$

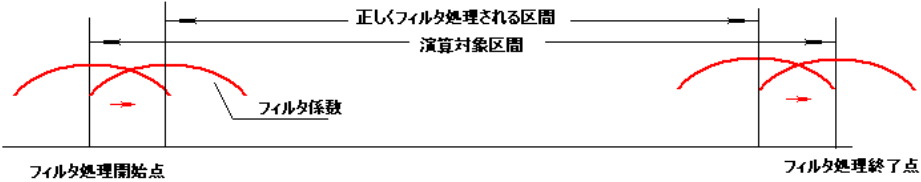
従って、過渡領域が先端部/終端部に発生します。過渡領域のデータ個数は $\text{INT}(k/2)$ 個となります。移動平均処理結果を周波数軸上で見ると、LowPassFilter の特性を持ちます。

周波数軸上特性図(平均化個数を 20Hz 相当に設定時)

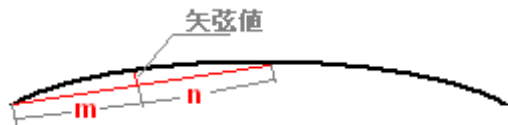


移動平均個数を周期とした λ の $n(1,2,3,\dots)$ 倍で最大減衰となります。

2. 10. 6. FIR フィルタ実行関数

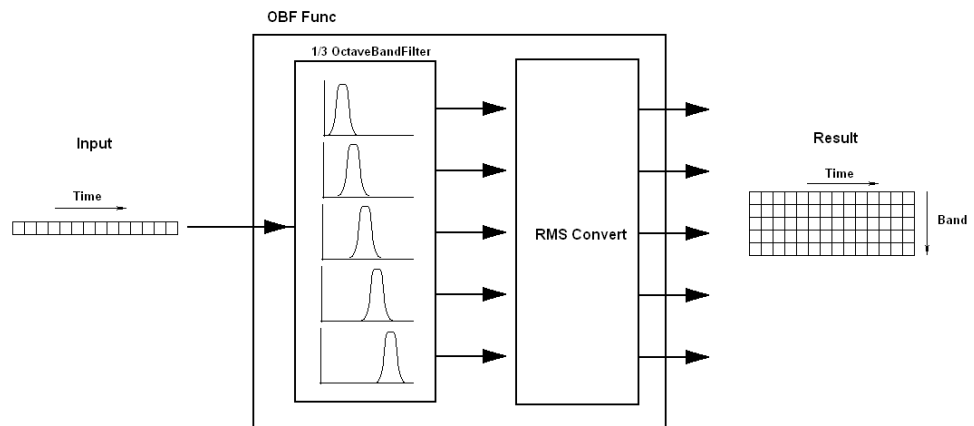
機能	係数 k で与えられた FIR フィルタ係数を X に掛けた結果を戻します。
文法	$\text{FIR}(k,X)$
引数	【 k 】<必須><即値/数値属性参照チャンネル/演算式> FIR フィルタ係数を記述します。 【 X 】<必須><数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	$\text{FIR}(\$1, \$2)$
備考	FIR フィルタ処理では、先頭からフィルタ係数個数の $1/2$ 個、終端から同じフィルタ係数個数の $1/2$ 個の過渡領域が発生します。過渡領域とは正しくフィルタが掛っていない領域を意味します。 なお、フィルタ個数が奇数の場合は、係数個数を N とすると $(N-1)/2$ となります。 過渡領域の処理に関しては次の様に処理します。  フィルタ係数がはみ出ている領域はデータ 0 と見なして処理します。 従って、フィルタ係数個数より演算対象数列のデータ個数が少ない場合は、正しくフィルタ処理されません。 又、従って、フィルタ係数個数より演算対象数列のデータ個数が十分に大きい必要があります。

2. 10. 7. 矢弦関数

機能	与えられた弦長 n, m から弦値を求めます。
文法	$\text{VSN}(n,m,X)$
引数	【 n 】<必須><即値/数値属性参照チャンネル/演算式> 前方弦長を記述します。なお、記述する値は演算対象数列の周期の倍数でなければなりません。 また、周期の単位は m と見なして処理されます。 【 m 】<省略可><即値/数値属性参照チャンネル/演算式> 後方弦長を記述します。記述省略された場合は n と同じと見なします。(正矢弦) 【 X 】<必須><数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	$\text{VSN}(5, \$1)$ /*10m 正矢弦を求めます*/
備考	演算結果の先頭から m で指定した個数範囲、及び終端から n で指定したデータ個数分は演算できませんので 0 が戻ります。 

2. 10. 8. 1/3 オクターブバンドフィルタ関数

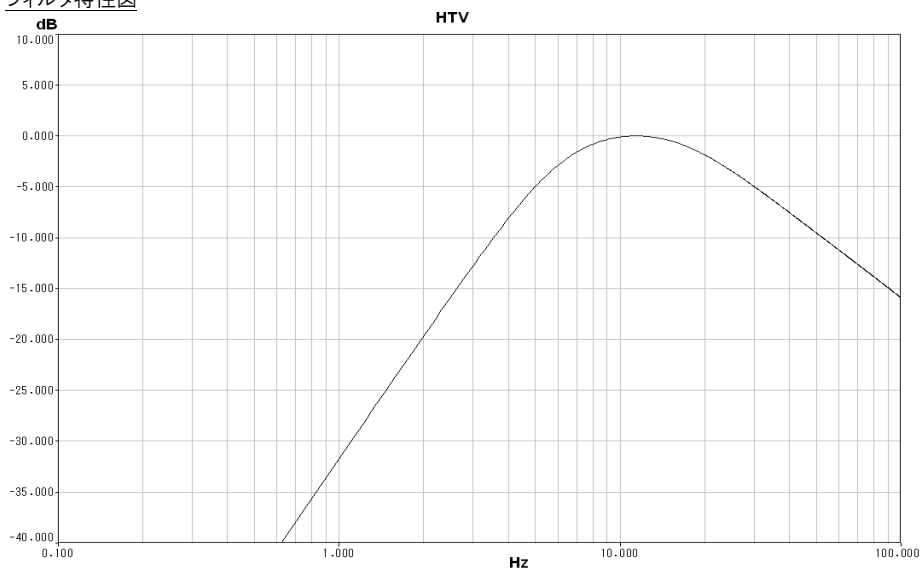
機能	指定された R10 系列 Index に相当する 1/3 オクターブフィルタを通過させた後、実効値に変換します。なお、実効値変換を行うか否か、及び積分時定数は任意に設定可能です。
文法	OBF(k,m,X)
引数	<p>[k]<必須><数値属性参照チャンネル/演算式> 解析対象バンド R10 系列 Index を記述します。要素並びは昇順の必要があります。</p> <p>[m]<省略可><即値/数値属性参照チャンネル/演算式> 移動実効値変換時定数を記述します。なお、0 及び-1 は特別な意味を持ち、0 と記述された場合は、X で与えたデータ個数分を時定数と見なします。又、-1 は実効値変換無しを意味します。記述省略された場合は 0 と見なします。従って、X で与える数列の全範囲を 1/3 オクターブ解析する場合は 0 と記述するか、又は記述省略します。0 以外を記述した場合は備考で記述します。</p> <p>[X]<必須><数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	OBF(\$1,\$4)
備考	<p>OBF 関数の戻り数列は、最小バンドから順に並び、仮想 2 次元数列となります。</p> <p>仮想行方向はバンド順に並び、仮想列方向は時間軸となります。但し、時定数に 0 記述した場合或いは記述省略した場合はバンド毎に実効値 1 個のデータとなりますので仮想行数は 1 列からなる一元数列となります。</p> <p>時定数に-1 を記述した場合は実効値変換せず、各バンド通過後の波形が戻ります。従って、仮想列方向は時間となり、列数は解析対象データ個数に等しくなります。なお、各仮想行方向はバンド毎の行となり、戻りデータ数は解析データ数×バンド数となり戻りデータ個数は増大します。従って、着目しているバンドのみ解析し、時間軸方向に着目したバンド範囲の波形がどの様になっているかを知りたい時などに-1 を指定します。</p> <p>時定数に実効値変換の時定数を記述した場合は、記述された時定数で移動実効値演算を行います。戻りデータ数は(解析データ数-時定数/サンプリング周波数)×バンド数となり戻りデータ数は増大します。</p> <p>なお、何れもメモリに余裕がある場合は全てのバンドを同時に処理しても問題ありません。</p> <p>つまり、時定数の 0 以外を指定した場合は、仮想 2 次元配列となりますので、受け取った後で 2 次元配列に変換すれば、行方向がバンドで列方向が時間経過となります。</p>



2. 11. 補正フィルタ関数

2. 11. 1. ISO5349-1 手腕振動知覚補正フィルタ

機能	XのISO5349-1 手腕振動暴露解析 X 軸/Y 軸/Z 軸補正フィルタ処理結果を戻します。
文法	HTV(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	HTV(#1)
備考	フィルタ特性図



ISO 5349-1:2001(E)

Band-limiting filter $f_1=6.310\text{Hz}$ $f_2=1258.9\text{Hz}$ $Q_1=0.71$

$$H_b = \frac{s^2 4\pi^2 f_2^2}{\left(s^2 + \frac{2\pi f_1 s}{Q_1} + 4\pi^2 f_1^2\right) \left(s^2 + \frac{2\pi f_2 s}{Q_1} + 4\pi^2 f_2^2\right)}$$

Frequency-weighting filter $f_3=15.915\text{Hz}$ $f_4=15.915\text{Hz}$ $Q_2=0.64$

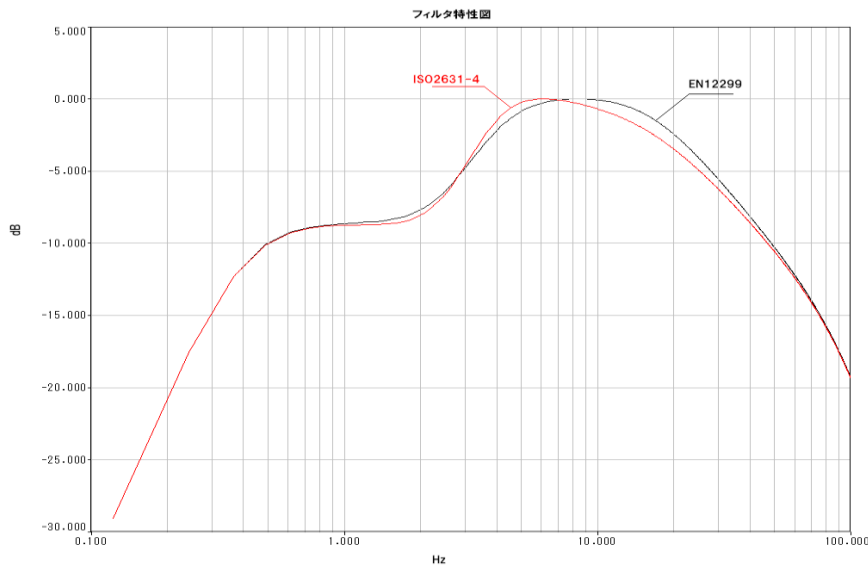
$$H_w = \frac{(s + 2\pi f_3) 2\pi f_4^2}{\left(s^2 + \frac{2\pi f_4 s}{Q_2} + 4\pi^2 f_4^2\right) f_3}$$

Overall frequency weighting

$$H(s) = H_b(s) H_w(s)$$

2. 11. 2. ISO2631-4:2001/EN12299:2009 Wb 振動知覚補正フィルタ

機能	記述するフラグ k により X の ISO2631-4:2001 Z 軸補正フィルタ処理、または、EN12299:2009 Z 軸補正フィルタ処理結果を戻します。
文法	WBB(k,X)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式> 処理するフィルタの選択フラグを記述します。k は 0 または 1 以外が設定された場合は、実行時 Error となります。 k⇒0 ISO2631-4 B フィルタ k⇒1 EN 12299:2009(E) B フィルタ 記述省略した場合は、k=0 と同じ意味を持ちます。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>
記述例	WBB(#1)
備考	



ISO2631-4:2001(E)

Band-limiting filter $f_1=0.4\text{Hz}$ $f_2=100\text{Hz}$ $Q_1=0.71$

$$H_b(p) = \frac{p^2 4\pi^2 f_2^2}{(p^2 + \frac{2\pi f_1}{Q_1} p + 4\pi^2 f_1^2)(p^2 + \frac{2\pi f_2}{Q_1} p + 4\pi^2 f_2^2)}$$

Weighting filter $f_3=16\text{Hz}$ $f_4=16\text{Hz}$ $f_5=2.5\text{Hz}$ $f_6=4\text{Hz}$ $Q_4=0.55$ $Q_5=0.90$ $Q_6=0.95$ $K=0.4$

$$H_w(p) = \frac{(p + 2\pi f_3)(p^2 + \frac{2\pi f_5}{Q_5} p + 4\pi^2 f_5^2)}{(p^2 + \frac{2\pi f_4}{Q_4} p + 4\pi^2 f_4^2)(p^2 + \frac{2\pi f_6}{Q_5} p + 4\pi^2 f_6^2)} \cdot \frac{2\pi K f_4^2 f_6^2}{f_3 \cdot f_5^2}$$

* $P=j\omega$

Overall frequency weighting

$$H(p) = H_b(p) \cdot H_w(p)$$

EN 12299:2009(E)

Band-limiting filter $f_1=0.4\text{Hz}$ $f_2=100\text{Hz}$ $Q_1=1/\sqrt{2}$

$$H_h(f) = \frac{1}{1 - (\frac{f_1}{f})^2 - i \frac{f_1}{Q_1 f}} \quad H_l(f) = \frac{1}{1 - (\frac{f}{f_2})^2 - i \frac{f}{Q_1 f_2}}$$

Acceleration to velocity translation $f_3=16\text{Hz}$ $f_4=16\text{Hz}$ $Q_2=0.63$

$$H_t = \frac{(1 + i \frac{f}{f_3})}{1 - (\frac{f}{f_4})^2 + i \frac{f}{Q_2 f_4}}$$

Upward gradient $f_5=2.5\text{Hz}$ $f_6=4\text{Hz}$ $Q_3=0.8$ $Q_4=0.8$ $K=0.4$

$$H_s = K \cdot \frac{1 - (\frac{f}{f_5})^2 + i \frac{f}{Q_3 f_5}}{1 - (\frac{f}{f_6})^2 + i \frac{f}{Q_4 f_6}}$$

Overall frequency weighting

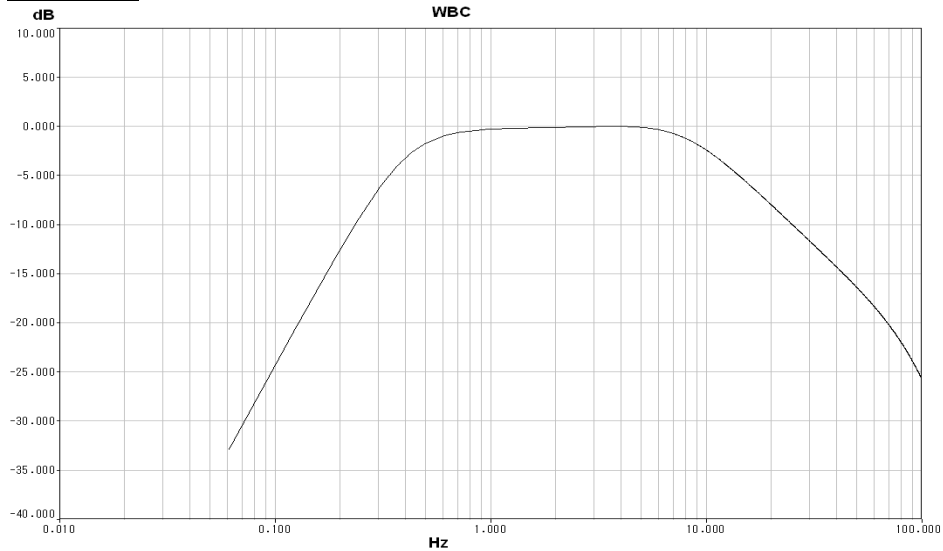
$$H(f) = H_h(f) \cdot H_l(f) \cdot H_t(f) \cdot H_s(f)$$

2. 11. 3. ISO2631-1 C 補正フィルタ

機能	X の ISO2631-1 全身振動暴露解析背もたれ X 軸補正フィルタ処理結果を戻します。
文法	WBC(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	WBC(#1)

備考

フィルタ特性図

ISO2631-1 W_c Band-limiting filter $f_1=0.4\text{Hz}$ $f_2=100\text{Hz}$

$$H_h(p) = \frac{1}{1 + \frac{\sqrt{2}\omega_1}{p} + \left(\frac{\omega_1}{p}\right)^2} \quad H_l(p) = \frac{1}{1 + \frac{\sqrt{2}p}{\omega_2} + \left(\frac{p}{\omega_2}\right)^2}$$

Acceleration-velocity translation $f_3=8.0\text{Hz}$ $f_4=8.0\text{Hz}$ $Q_4=0.63$

$$H_t(p) = \frac{1 + \frac{p}{\omega_2}}{1 + \frac{p}{Q_4\omega_4} + \left(\frac{p}{\omega_4}\right)^2}$$

Overall frequency weighting

$$H(p) = H_h(p) \cdot H_l(p) \cdot H_t(p)$$

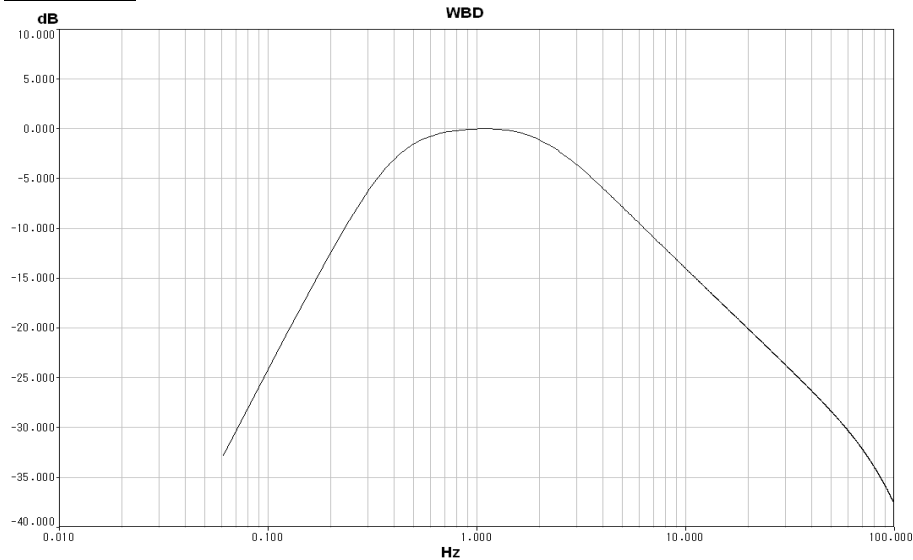
※ $\omega_n = 2\pi f_n$ $p = j\omega$

2. 11. 4. ISO2631-1 D 補正フィルタ

機能	X の ISO2631-1 全身振動暴露解析座位仰臥位 X,Y 軸補正フィルタ処理結果を戻します。
文法	WBD(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	WBD(#1)

備考

フィルタ特性図

ISO2631-1 W_d Band-limiting filter $f_1=0.4\text{Hz}$ $f_2=100\text{Hz}$

$$H_h(p) = \frac{1}{1 + \frac{\sqrt{2}\omega_1}{p} + \left(\frac{\omega_1}{p}\right)^2} \quad H_l(p) = \frac{1}{1 + \frac{\sqrt{2}p}{\omega_2} + \left(\frac{p}{\omega_2}\right)^2}$$

Acceleration-velocity translation $f_3=2.0\text{Hz}$ $f_4=2.0\text{Hz}$ $Q_4=0.63$

$$H_t(p) = \frac{1 + \frac{p}{\omega_2}}{1 + \frac{p}{Q_4\omega_4} + \left(\frac{p}{\omega_4}\right)^2}$$

Overall frequency weighting

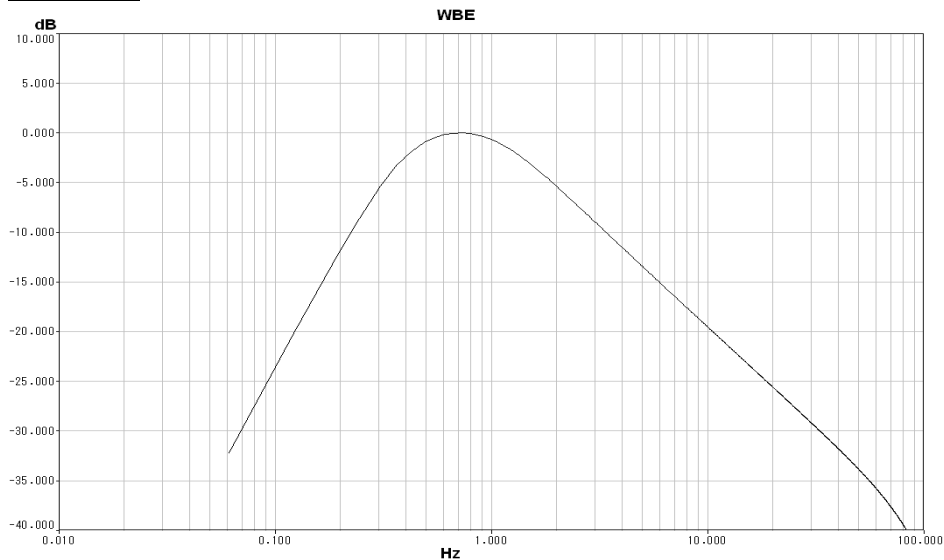
$$H(p) = H_h(p) \cdot H_l(p) \cdot H_t(p)$$

※ $\omega_n = 2\pi f_n$ $p = j\omega$

2. 11. 5. ISO2631-1 E 補正フィルタ

機能	X の ISO2631-1 全身振動暴露解析座位回転モーメント補正フィルタ処理結果を戻します。
文法	WBE(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	WBE(#1)

備考

ISO2631-1 W_e Band-limiting filter $f_1=0.4\text{Hz}$ $f_2=100\text{Hz}$

$$H_h(p) = \frac{1}{1 + \frac{\sqrt{2}\omega_1}{p} + \left(\frac{\omega_1}{p}\right)^2} \quad H_l(p) = \frac{1}{1 + \frac{\sqrt{2}p}{\omega_2} + \left(\frac{p}{\omega_2}\right)^2}$$

Acceleration-velocity translation $f_3=1.0\text{Hz}$ $f_4=1.0\text{Hz}$ $Q_4=0.63$

$$H_t(p) = \frac{1 + \frac{p}{\omega_2}}{1 + \frac{p}{Q_4\omega_4} + \left(\frac{p}{\omega_4}\right)^2}$$

Overall frequency weighting

$$H(p) = H_h(p) \cdot H_l(p) \cdot H_t(p)$$

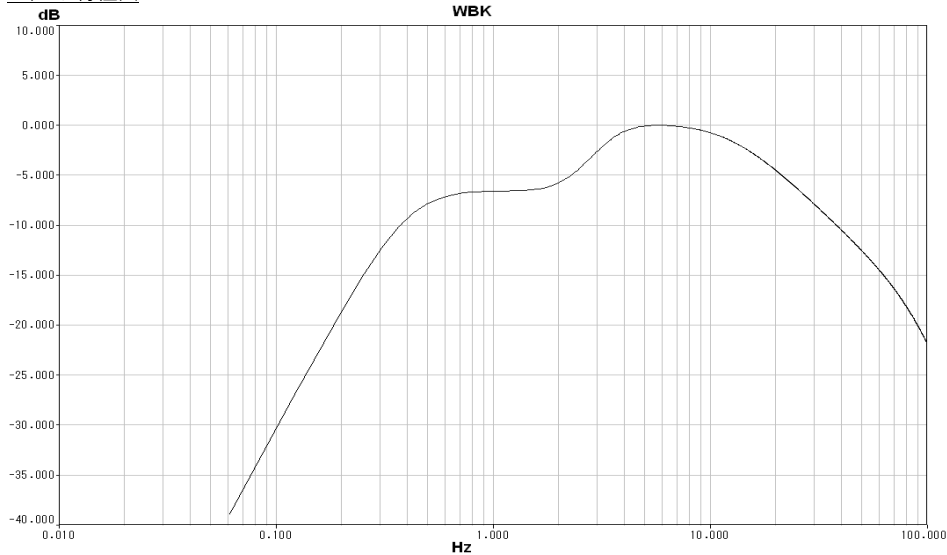
※ $\omega_n = 2\pi f_n$ $p = j\omega$

2. 11. 6. ISO2631-1 K 補正フィルタ

機能	X の ISO2631-1 全身振動暴露解析座位仰臥位 Z 軸補正フィルタ処理結果を戻します。
文法	WBK(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	WBK(#1)

備考

フィルタ特性図

ISO2631-1 W_k Band-limiting filter $f_1=0.4\text{Hz}$ $f_2=100\text{Hz}$

$$H_h(p) = \frac{1}{1 + \frac{\sqrt{2}\omega_1}{p} + \left(\frac{\omega_1}{p}\right)^2} \quad H_l(p) = \frac{1}{1 + \frac{\sqrt{2}p}{\omega_2} + \left(\frac{p}{\omega_2}\right)^2}$$

Acceleration-velocity translation $f_3=12.5\text{Hz}$ $f_4=12.5\text{Hz}$ $Q_4=0.63$

$$H_t(p) = \frac{1 + \frac{p}{\omega_2}}{1 + \frac{p}{Q_4\omega_4} + \left(\frac{p}{\omega_4}\right)^2}$$

Upward step $f_5=2.37\text{Hz}$ $f_6=3.35\text{Hz}$ $Q_5=0.91$ $Q_6=0.91$

$$H_s(p) = \frac{1 + \frac{p}{Q_5\omega_5} + \left(\frac{p}{\omega_5}\right)^2}{1 + \frac{p}{Q_6\omega_6} + \left(\frac{p}{\omega_6}\right)^2} \cdot \left(\frac{\omega_5}{\omega_6}\right)^2$$

Overall frequency weighting

$$H(p) = H_h(p) \cdot H_l(p) \cdot H_t(p) \cdot H_s(p)$$

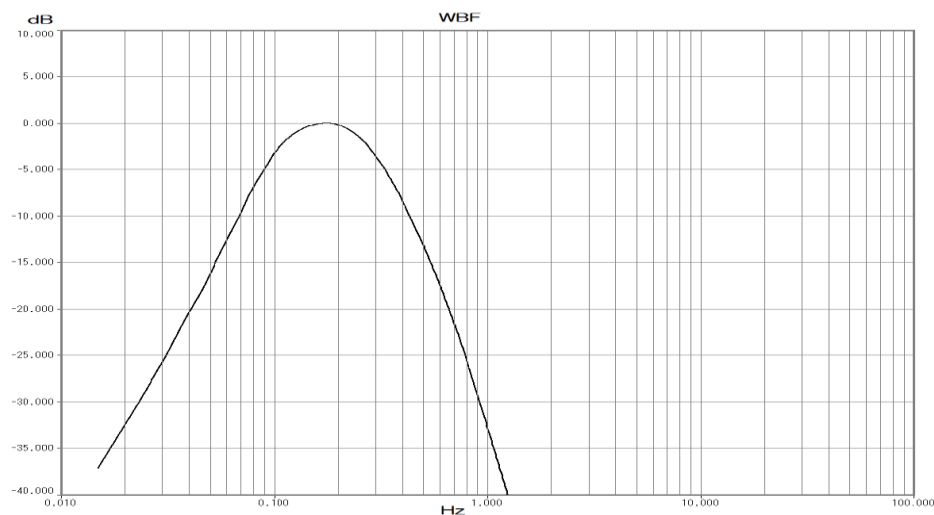
※ $\omega_n = 2\pi f_n$ $p = j\omega$

2. 11. 7. ISO2631-1 F 補正フィルタ

機能	X の ISO2631-1 Wf 動揺病(乗り物酔い)補正フィルタ処理結果を戻します。
文法	WBF(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	WBF(#1)

備考

フィルタ特性図

ISO2631-1 W_fBand-limiting filter $f_1=0.08\text{Hz}$ $f_2=0.63\text{Hz}$

$$H_b(p) = \frac{1}{1 + \frac{\sqrt{2}\omega_1}{p} + \left(\frac{\omega_1}{p}\right)^2} \quad H_l(p) = \frac{1}{1 + \frac{\sqrt{2}p}{\omega_2} + \left(\frac{p}{\omega_2}\right)^2}$$

Acceleration-velocity translation $f_3=\infty\text{Hz}$ $f_4=0.25\text{z}$ $Q_4=0.86$

$$H_t(p) = \frac{1 + \frac{p}{\omega_2}}{1 + \frac{p}{Q_4\omega_4} + \left(\frac{p}{\omega_4}\right)^2}$$

Upward step $f_5=0.0625\text{Hz}$ $f_6=0.1\text{z}$ $Q_5=0.80$ $Q_6=0.80$

$$H_s(p) = \frac{1 + \frac{p}{Q_5\omega_5} + \left(\frac{p}{\omega_5}\right)^2}{1 + \frac{p}{Q_6\omega_6} + \left(\frac{p}{\omega_6}\right)^2} \cdot \left(\frac{\omega_5}{\omega_6}\right)^2$$

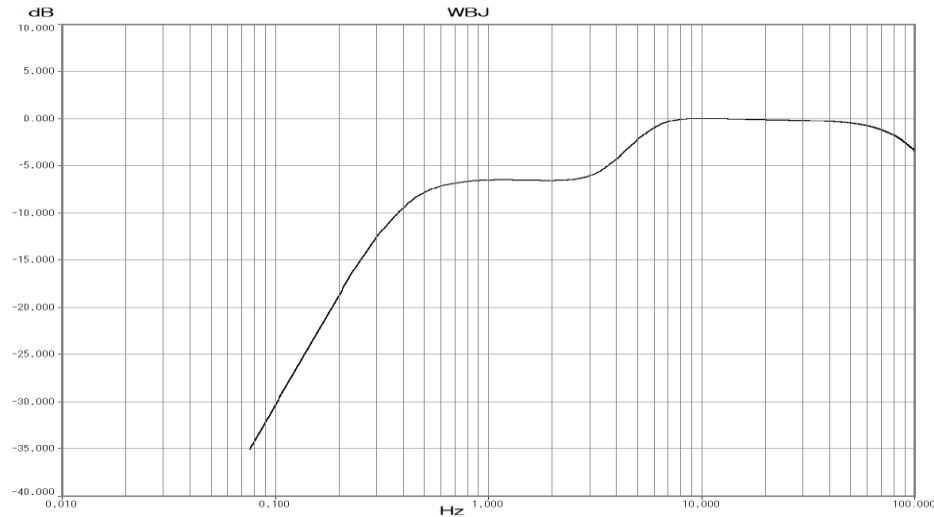
Overall frequency weighting

$$H(p) = H_b(p) \cdot H_l(p) \cdot H_t(p) \cdot H_s(p)$$

※ $\omega_n = 2\pi f_n$ $p = j\omega$

2. 11. 8. ISO2631-1 J 補正フィルタ

機能	X の ISO2631-1Wj 頭部暴露解析仰臥位補正フィルタ処理結果を戻します。
文法	WBJ(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	WBJ(#1)
備考	フィルタ特性図

ISO2631-1 W_j Band-limiting filter $f_1=0.4\text{Hz}$ $f_2=100\text{Hz}$

$$H_h(p) = \frac{1}{1 + \frac{\sqrt{2}\omega_1}{p} + \left(\frac{\omega_1}{p}\right)^2} \quad H_l(p) = \frac{1}{1 + \frac{\sqrt{2}p}{\omega_2} + \left(\frac{p}{\omega_2}\right)^2}$$

Acceleration-velocity translation $f_3=\infty\text{Hz}$ $f_4=\infty\text{Hz}$ $Q_4=-$

$$H_t(p) = \frac{1 + \frac{p}{\omega_2}}{1 + \frac{p}{Q_4\omega_4} + \left(\frac{p}{\omega_4}\right)^2}$$

Upward step $f_5=3.75\text{Hz}$ $f_6=5.32\text{Hz}$ $Q_5=0.91$ $Q_6=0.91$

$$H_s(p) = \frac{1 + \frac{p}{Q_5\omega_5} + \left(\frac{p}{\omega_5}\right)^2}{1 + \frac{p}{Q_6\omega_6} + \left(\frac{p}{\omega_6}\right)^2} \cdot \left(\frac{\omega_5}{\omega_6}\right)^2$$

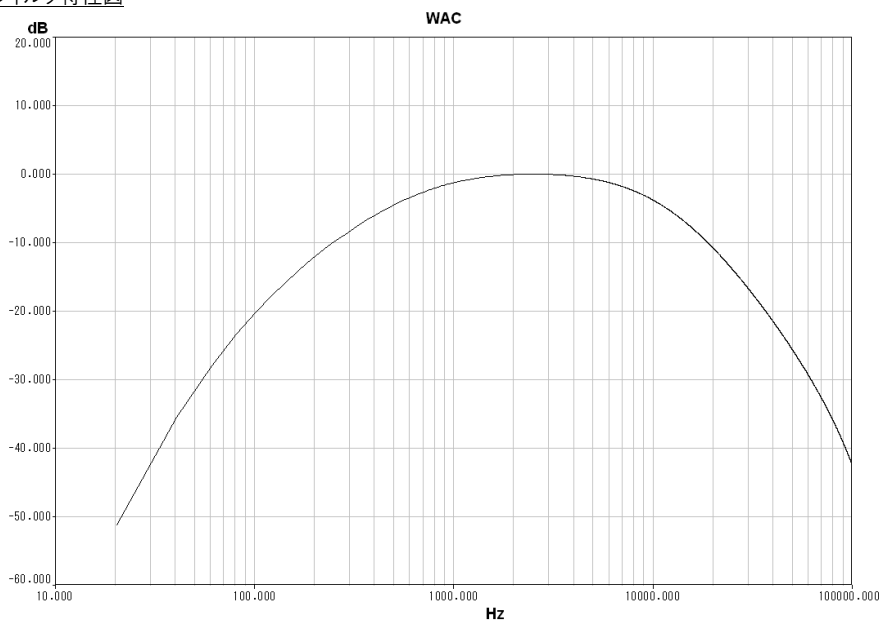
Overall frequency weighting

$$H(p) = H_h(p) \cdot H_l(p) \cdot H_t(p) \cdot H_s(p)$$

※ $\omega_n = 2\pi f_n$ $p = j\omega$

2. 11. 9. ANSI 聴感補正 A カーブ・フィルタ

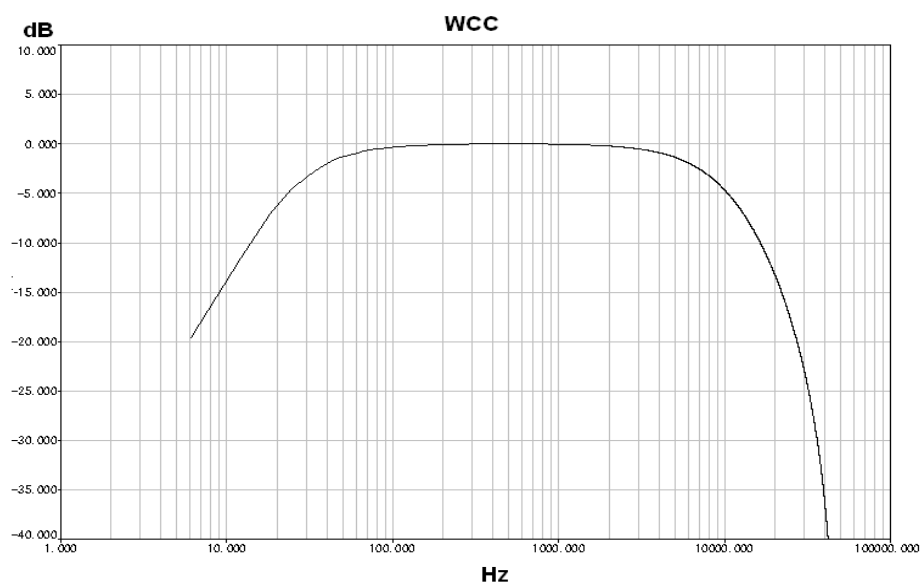
機能	X の聴感補正 A カーブ・フィルタ処理結果を戻します。
文法	WAC(X)
引数	[X]<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	WAC(#1)
備考	フィルタ特性図



フィルタ特性図は振幅最大値を 0dB としたグラフです。

2. 11. 10. ANSI 聴感補正 C カーブ・フィルタ


機能	X の聴感補正 C カーブ・フィルタ処理結果を戻します。
文法	WCC(X)
引数	[X]<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	WCC(#1)
備考	フィルタ特性図




フィルタ特性図は振幅最大値を 0dB としたグラフです。

2. 11. ロゼッタゲージ解析関数


2. 12. 1. 最小主ひずみ関数

機能	a,b,c をロゼッタゲージひずみ信号として、最小主ひずみを戻します。
文法	EMN(a,b,c)
引数	<p>【a】<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ a 軸ひずみ数列を記述します。</p> <p>【b】<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ b 軸ひずみ数列を記述します。</p> <p>【c】<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ c 軸ひずみ数列を記述します。 ただし、引数 a,b,c のデータ個数は等しくなくてはなりません。</p>
記述例	EMN(#1,#2,#3) → Ch1,2,3 で構成されたロゼッタゲージから最小主ひずみを戻します。
備考	$\varepsilon_{\min} = \frac{1}{2} \left\{ \varepsilon_a + \varepsilon_c - \sqrt{2 \left[(\varepsilon_a - \varepsilon_b)^2 + (\varepsilon_b - \varepsilon_c)^2 \right]} \right\}$ 

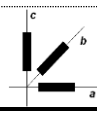
2. 12. 2. 最大主ひずみ関数

機能	a,b,c をロゼッタゲージひずみ信号として、最大主ひずみを戻します。
文法	EMX(a,b,c)
引数	<p>【a】<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ a 軸ひずみ数列を記述します。</p> <p>【b】<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ b 軸ひずみ数列を記述します。</p> <p>【c】<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ c 軸ひずみ数列を記述します。 ただし、引数 a,b,c のデータ個数は等しくなくてはなりません。</p>
記述例	EMX(#1,#2,#3) → Ch1,2,3 で構成されたロゼッタゲージから最大主ひずみを戻します。
備考	$\varepsilon_{\max} = \frac{1}{2} \left\{ \varepsilon_a + \varepsilon_c + \sqrt{2 \left[(\varepsilon_a - \varepsilon_b)^2 + (\varepsilon_b - \varepsilon_c)^2 \right]} \right\}$ 

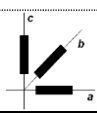
2. 12. 3. 主ひずみ方向関数

機能	a,b,c をロゼッタゲージひずみ信号として、主ひずみ方向を戻します
文法	ETH(a,b,c)
引数	<p>【a】<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ a 軸ひずみ数列を記述します。</p> <p>【b】<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ b 軸ひずみ数列を記述します。</p> <p>【c】<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ c 軸ひずみ数列を記述します。 ただし、引数 a,b,c のデータ個数は等しくなくてはなりません。</p>
記述例	ETH(#1,#2,#3)
備考	<p>演算結果は最大ひずみ方向を示し単位は degree となります。</p> $\theta = \frac{1}{2} \tan^{-1} \left\{ \frac{2\varepsilon_b - \varepsilon_a - \varepsilon_c}{\varepsilon_a - \varepsilon_c} \right\}$ 

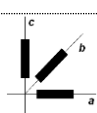
2. 12. 4. 最大せん断ひずみ関数

機能	a,b,c をロゼッタゲージひずみ信号として、最大せん断ひずみを戻します
文法	GMX(a,b,c)
引数	<p>[a]<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ a 軸ひずみ数列を記述します。</p> <p>[b]<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ b 軸ひずみ数列を記述します。</p> <p>[c]<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ c 軸ひずみ数列を記述します。 ただし、引数 a,b,c のデータ個数は等しくなくてはなりません。</p>
記述例	GMX(a,b,c)
備考	 $\varepsilon_{\max} = \sqrt{2[(\varepsilon_a - \varepsilon_b)^2 + (\varepsilon_b - \varepsilon_c)^2]}$

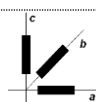
2. 12. 7. 最小主応力関数

機能	a,b,c をロゼッタゲージ信号として、最小主応力を戻します。
文法	SMN(E,v,a,b,c)
引数	<p>[E]<必須><即値/数値属性参照チャンネル> 縦弾性係数を記述します。</p> <p>[v]<必須><即値/数値属性参照チャンネル> ポアソン比を記述します。</p> <p>[a]<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ a 軸ひずみ数列を記述します。</p> <p>[b]<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ b 軸ひずみ数列を記述します。</p> <p>[c]<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ c 軸ひずみ数列を記述します。 ただし、引数 a,b,c のデータ個数は等しくなくてはなりません。</p>
記述例	SMN(\$1,\$2,#1,#2,#3)
備考	 $\sigma_{\min} = \frac{E}{2(1-\nu^2)} \left\{ (1+\nu) \times (\varepsilon_a + \varepsilon_c) - (1-\nu) \times \sqrt{2[(\varepsilon_a - \varepsilon_b)^2 + (\varepsilon_b - \varepsilon_c)^2]} \right\}$

2. 12. 8. 最大主応力関数

機能	a,b,c をロゼッタゲージひずみ信号として、最大主応力を戻します。
文法	SMX(E,v,a,b,c)
引数	<p>[E]<必須><即値/数値属性参照チャンネル> 縦弾性係数を記述します。</p> <p>[v]<必須><即値/数値属性参照チャンネル> ポアソン比を記述します。</p> <p>[a]<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ a 軸ひずみ数列を記述します。</p> <p>[b]<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ b 軸ひずみ数列を記述します。</p> <p>[c]<必須><即値/収録チャンネル/数値属性参照チャンネル> ロゼッタゲージ c 軸ひずみ数列を記述します。 ただし、引数 a,b,c のデータ個数は等しくなくてはなりません。</p>
記述例	SMX(\$1,\$2,#1,#2,#3)
備考	 $\sigma_{\max} = \frac{E}{2(1-\nu^2)} \left\{ (1+\nu) \times (\varepsilon_a + \varepsilon_c) + (1-\nu) \times \sqrt{2[(\varepsilon_a - \varepsilon_b)^2 + (\varepsilon_b - \varepsilon_c)^2]} \right\}$

2. 12. 9. 最大せん断応力関数

機能	a,b,c をロゼッタゲージ信号として、最大せん断応力を戻します。
文法	TMX(E,v,a,b,c)
引数	<p>【E】<必須><即値/数値属性参照チャネル> 縦弾性係数を記述します。</p> <p>【v】<必須><即値/数値属性参照チャネル> ポアソン比を記述します。</p> <p>【a】<必須><即値/収録チャネル/数値属性参照チャネル> ロゼッタゲージ a 軸ひずみ数値を記述します。</p> <p>【b】<必須><即値/収録チャネル/数値属性参照チャネル> ロゼッタゲージ b 軸ひずみ数値を記述します。</p> <p>【c】<必須><即値/収録チャネル/数値属性参照チャネル> ロゼッタゲージ c 軸ひずみ数値を記述します。ただし、引数 a,b,c のデータ個数は等しくなくてはなりません。</p>
記述例	TMX(\$1,\$2,#1,#2,#3)
備考	$\tau_{max} = \frac{E}{2(1+\nu)} \sqrt{2[(\epsilon_a - \epsilon_b)^2 + (\epsilon_b - \epsilon_c)^2]}$ 

2. 12. 10. ゲージ率補正関数

機能	ひずみデータのゲージ率補正結果を戻します。
文法	GRT(k,X)
引数	<p>【k】<必須><即値/数値属性参照チャネル> 補正するゲージ率を記述します。</p> <p>【X】<必須><即値/収録チャネル/数値属性参照チャネル> ゲージ率 2.0 のひずみ値数値を記述します。</p>
記述例	GRT(\$1,#1)
備考	<p>演算対象データ列をひずみ量としてゲージ率を補正します。</p> $Ans_n = \frac{2.0}{k} \times X_n$

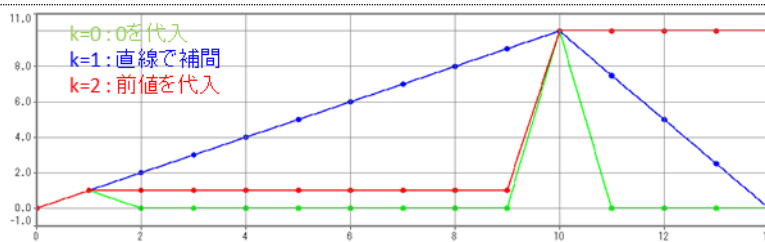
2. 13. 数列操作関数

2. 13. 1. 波形数列生成関数

機能	指定した波形データ列を生成した結果を戻します。										
文法	DAG(k,a,m,f,p)										
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 生成するデータ個数を記述します。</p> <p>【a】<必須><即値/数値属性参照チャネル/演算式> 生成する波形の波高値(片振幅)を記述します。</p> <p>【m】<省略可><即値/数値属性参照チャネル> 生成する波形種別フラグを記述します。記述する波形種別フラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>m 値</th> <th>生成される波形種別</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>正弦波</td> </tr> <tr> <td>1</td> <td>三角波</td> </tr> <tr> <td>2</td> <td>矩形波</td> </tr> <tr> <td>3</td> <td>鋸波</td> </tr> </tbody> </table> <p>波形種別フラグを記述省略した場合は以後の引数を記述できません。 波形種別フラグを記述省略した場合は、波高値を持つ直流数列が生成されます。</p> <p>【f】<省略可><即値/数値属性参照チャネル/演算式> 生成する波形の周波数を記述します。ただし、f の値は、現在設定されているサンプリング周期の 2 倍以上でなければなりません。 f >= 2/PRD()、周波数を記述省略した場合は以後の引数を記述できません。 周波数を記述省略した場合は、現在設定されているサンプリング周期の 100 倍の周期を持つ周波数が暗黙的に採られ、位相角は 0° となります。</p> <p>【p】<省略可><即値/数値属性参照チャネル/演算式> 生成する波形の開始時位相角を単位 deg(角度)で記述します。 記述省略された場合は 0° と見なします。</p>	m 値	生成される波形種別	0	正弦波	1	三角波	2	矩形波	3	鋸波
m 値	生成される波形種別										
0	正弦波										
1	三角波										
2	矩形波										
3	鋸波										
記述例	<p>DAG(5000,5,0,20,0) ⇒ $5 \times \sin(2 \times \pi \times 20)$ の数列を生成</p> <p>DAG(5000,10) ⇒ 要素数 5000 の全ての要素値 10 の数列を生成</p>										
備考	<p>指定された引数により、現在設定されているサンプリング周期を参照して波形データを生成した結果を戻します。結果データ数は k で指定した個数となります。</p>										

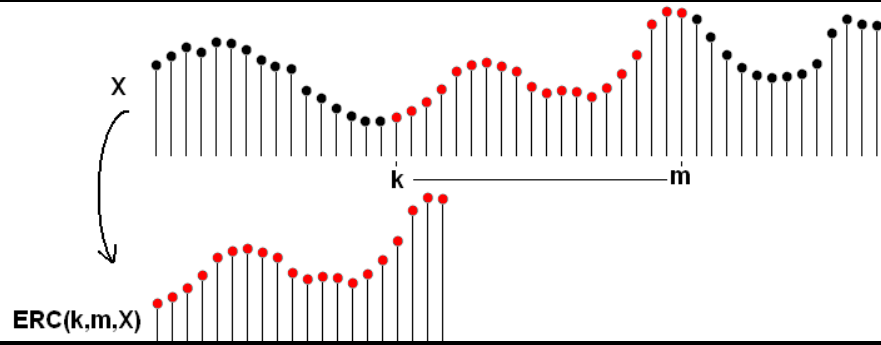
2. 13. 2. 数列生成関数

機能	データ番号配列とデータ値配列、および生成後のデータ個数から新たな配列を生成します。
文法	VRP(k,n,m,X)
引数	<p>【k】<必須><即値/数値属性参照チャネル> 補間有無指定フラグを記述します。</p> <p>k=0 の時: 代入されなかったデータ番号位置の値に 0 を代入します。</p> <p>k=1 の時: 代入されなかったデータ位置に前後のデータから直線補間した値を代入します。なお、先頭データ番号位置に代入されなかった場合は、先頭に 0 を代入してから最初に代入された位置の値までを直線補間します。同様に、終端データ番号に代入されなかった場合は終端の 0 を代入して終端から遡った最後に代入された位置の値までを直線補間します。</p> <p>k=2 の時: 代入されなかったデータ位置に前値を代入します。先頭データ番号位置に代入されなかった場合は、先頭に 0 を代入します。</p> <p>【n】<必須><即値/数値属性参照チャネル/演算式> 生成する配列のデータ個数を記述します。</p> <p>【m】<必須><数値属性参照チャネル/演算式> X で与えるデータ値のデータ番号列(Index)を記述します。記述するデータ番号は昇順かつ唯一無二で、その最大値は n-1 の必要があります。</p> <p>【X】<必須><数値属性参照チャネル/演算式> データ列を記述します。m(0)位置に X(0)が、同様に m(1)位置に X(1)、m(i)位置に X(i)が代入された数列が生成されます。</p>
記述例	VRP(1,1024,\$1,\$2)
備考	<p>戻りデータ個数は n となります。</p> <p>例)</p> <pre>assign \$1 = 1,10 \$2 = VRP(0,15,\$1,\$1) \$3 = VRP(1,15,\$1,\$1) \$4 = VRP(2,15,\$1,\$1)</pre> <p>とした場合に生成される各数列は左図の様になります。</p> <p>この例では始端、終端データ番号に数値を指定していないため、k=1 の時は始端、終端を 0 として扱い、与えたデータと直線補完しています。</p>



2. 13. 3. Index 範囲指定データ抽出関数

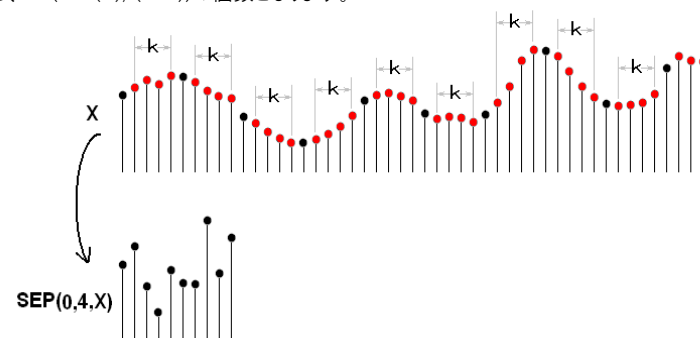
機能	X から指定した切り出し開始/終了データ番号範囲で切り出した結果を戻します。
文法	ERC(k,m,X)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 演算対象数列 X から切り出し開始位置のデータ番号を記述します。k は、X の Index 範囲内の必要があります。</p> <p>【m】<必須><即値/数値属性参照チャネル/演算式> 演算対象数列 X から切り出し終了位置のデータ番号を記述します。m は、X の Index 範囲内の必要があり、又、k<=m でなくてはなりません。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 切り出し対象数列を記述します。</p>
記述例	ERC(100,199,#1)
備考	演算対象データから k で示すデータ番号から m で示すデータ番号までを切り出した結果を戻します。



2. 13. 4. Index 指定データ抽出関数

機能	指定したデータ番号のデータ値を戻します。
文法	PTV(k,X)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> データを読み出すデータ番号を記述します。なお、k で指定するデータ番号は、X で記述する演算対象数列の Index 内でなければなりません。データ番号を複数要素持つ配列で記述した場合、結果も複数個となります。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	PTV(\$1,#2)
備考	データ番号が演算対象数列の Index 範囲にない場合は実行時エラーとなります。

2. 13. 5. Index 飛び越しデータ抽出関数

機能	X を指定した[飛び越し数毎のデータで再構成した結果を戻します。
文法	SEP(m,k,X)
引数	<p>【m】<必須><即値/数値属性参照チャンネル> X から抽出する開始位置のデータ番号を記述します。m は X の Index 範囲になくてはなりません。</p> <p>【k】<必須><即値/数値属性参照チャンネル> データ飛び越し数を記述します。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	SEP(0,4,#1)
備考	<p>演算対象数列 X から m で指定した開始位置から k 指定した飛び越し数で飛び越してデータを再構成した結果を戻します。戻りデータ個数は縮退し、$\text{INT}(\text{LEN}(X)/(k+1))$ の個数となります。</p>  <p>\$1 ← 1,2,3,4,1,2,3,4,1,2,3,4 \$1 = SEP(2,3,\$1) \$1 → 3,3,3</p>

2. 13. 6. 論理値指定データ抽出関数

機能	Y を X で指定した論理値"1"のデータ番号のみで再構成した結果を戻します。
文法	ZSP(X,Y)
引数	<p>【X】<必須><数値属性参照チャンネル/演算式> 抽出するデータ番号を論理数列で記述します。論理数列は抽出するデータ番号の値を"1"とし、削除するデータ番号の値を"0"とした数列を意味します。X のデータ個数は Y のデータ個数と等しくなければなりません。</p> <p>【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 抽出対象数列を記述します。</p>
記述例	ZSP(\$1,#1)
備考	演算対象データを X 指定した論理値"1"のデータ番号の Y のデータのみ抽出した結果を戻します。戻りデータ個数は、X で記述した論理数列の"1"の数に一致します。

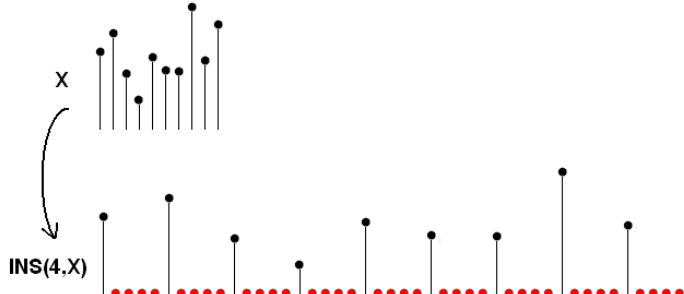
2. 13. 7. Index 範囲指定データ置換関数

機能	Y のデータを X で指定したデータに置き換えた結果を戻します。
文法	REP(k,X,Y)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> データを置き換え開始するデータ番号記述します。なお、k は、Y で記述する置き換え元データ番号の範囲でなければなりません。 存在しない場合は置き換えを行いません。</p> <p>【X】<必須><即値/収録チャンネル数値属性参照チャンネル/演算式> 置き換えるデータを記述します。</p> <p>【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	REP(0,\$1,#1)
備考	<p>Y で示すデータの k で指定するデータ番号から、X で示すデータに置き換えた結果を戻します。なお、X のデータ個数が、置き換えた結果 Y の最終データを越える場合は、余り分は無視されます。</p> <p>複数個所を一度に置き換えることはできません。</p>

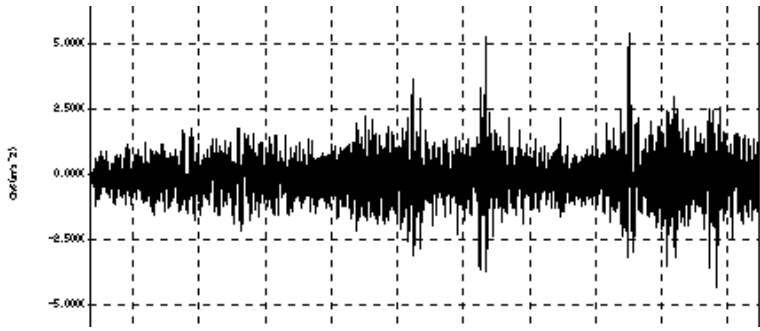
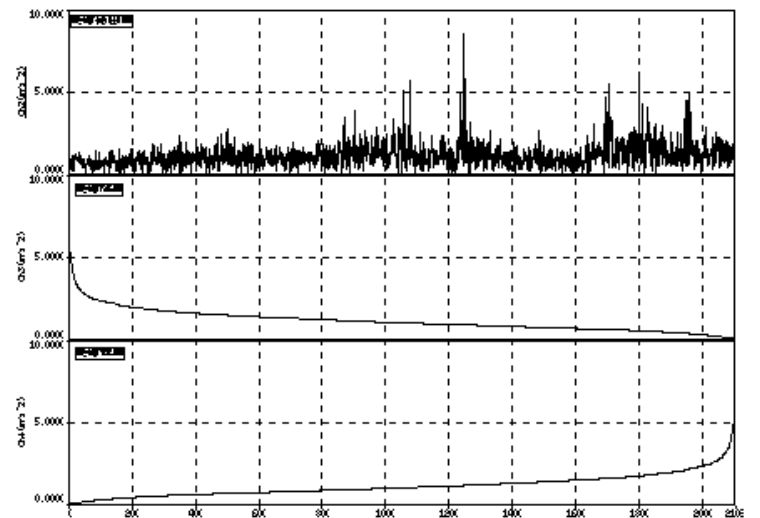
2. 13. 8. Index 指定データ置換関数

機能	指定したデータ番号位置のデータ値を指定したデータ値に置き換えます。
文法	SBV(k,m,X)
引数	<p>【k】<必須><即値/数値属性参照チャンネル/演算式> データを変更するデータ番号を記述します。k で指定するデータ番号は、X の Index 範囲内に必要があります。 又、記述するデータ番号は昇順の必要があります。</p> <p>【m】<必須><即値/数値属性参照チャンネル/演算式> 置換するデータ値を記述します。複数要素を持つ数列で記述する場合、k と同じデータ個数の必要があります。</p> <p>【X】<必須></数値属性参照チャンネル/演算式> 演算対象数列を記述します。</p>
記述例	SBV(\$1,\$2,#2)
備考	演算対象データ X の k で指定されたデータ番号のデータ値を m で指定するデータに変更します。変換するデータ番号 k が複数の要素を持つ配列の場合、結果も複数個となります。

2. 13. 9. Index 飛び越しゼロ挿関数

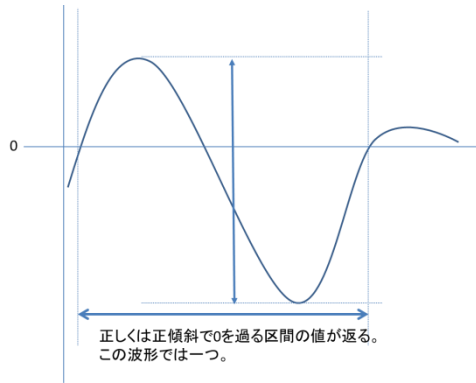
機能	X のデータ間に指定した数のゼロデータを挿入します。
文法	INS(k,X)
引数	<p>【k】<必須><即値/数値属性参照チャネル> 挿入するゼロ個数を記述します。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>
記述例	INS(\$1,\$2)
備考	<p>演算対象データ間に、k で指定された個数分のゼロ値を挿入した結果を戻します。戻りデータ個数は、(X のデータ個数-1)×(k+1)+1 となります。</p> 

2.13.10. 振幅値変換関数

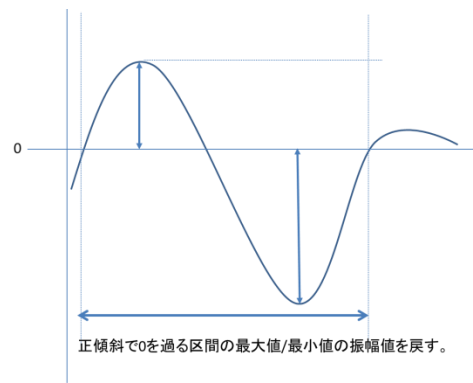
機能	X が正傾斜でゼロを過ぎてから再び正傾斜でゼロを過ぎるまでの最大値、および最小値から振幅を抽出し、全てのデータを振幅値または最大値最小値の絶対値に変換します。										
文法	AMP(k,X,m)										
引数	<p>【k】<必須><即値/数値属性参照チャネル> 結果並び形式フラグを記述します。記述する結果並びフラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>k</th> <th>結果並び</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>振幅値を昇順</td> </tr> <tr> <td>0</td> <td>振幅値を抽出順</td> </tr> <tr> <td>-</td> <td>振幅値を降順</td> </tr> <tr> <td>1</td> <td>振幅値を降順</td> </tr> </tbody> </table> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p> <p>【m】:<省略可><即値/数値属性参照チャネル> m=0 または省略時 正傾斜で 0 を過る区間の振幅値を戻す。 m<>1 の時 正傾斜で 0 を過る区間の最大値と最小値の振幅値を戻す。</p>	k	結果並び	1	振幅値を昇順	0	振幅値を抽出順	-	振幅値を降順	1	振幅値を降順
k	結果並び										
1	振幅値を昇順										
0	振幅値を抽出順										
-	振幅値を降順										
1	振幅値を降順										
記述例	<p>AMP(0,#1) → Ch1 のデータを振幅値に変換します。</p> <p>AMP(0,#1,1) → Ch1 のデータを最大値最小値の絶対値に変換します。</p> <p>AMP(1,#1) → Ch1 のデータを振幅値に変換後、昇順に並べます。</p> <p>AMP(-1,#1) → Ch1 のデータを振幅値に変換後、降順に並べ換えます。</p>										
備考	<p>引数 k の値によって、求めた振幅値配列が並び替えられます。ゼロを通過しない場合は、振幅は求まりません。また、結果のデータ個数は抽出した振幅値数に依存します。X 軸は便宜的に時間軸となりますが意味を持ちません。X 軸は振幅抽出個数属性を持ち、結果のデータ個数は抽出振幅個数となります。</p> <p>元波形</p>  <p>演算結果</p> <p>上段-振幅抽出順、中段-振幅降順、下段-振幅昇順</p> 										

フラグ m の違いにより戻り値は下記のようになります。

$m=0$



$m < > 0$



2. 13. 11. 著大値除去関数

機能	X がゼロ中心とした振幅値が設定した値を越えた場合、フラグ 0 または省略したはその範囲をデータから除去します。フラグが 0 以外の場合はその範囲の始端 Index 及び終端 Index のペアを戻します。
文法	LMT(k,X,m)
引数	<p>【k】<必須><即値/数値属性参照チャネル> 除去する著大振幅値を記述します。ただし、k は、$k > 0$ でなければなりません。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p> <p>【m】<省略可><即値/数値属性参照チャネル/演算式> 戻り値選択フラグを記述します。 $m=0$ の時: 記述省略した場合は、$m=0$ と等価です。 ゼロを上昇で過ってから再びゼロを上昇で過る範囲の両端振幅値が振幅閾値を超えた場合に該当範囲を削除します。 $m < 0$ の時: 同じく該当範囲の始端 Index と終端 Index のペアが対象範囲数分戻ります。閾値を超える範囲が存在しない場合は -1 が戻ります。</p>
記述例	LMT(12.34,#4)
備考	<p>演算対象データが正傾斜でゼロを通過してから、再び正傾斜でゼロを通過するまでの最大値および最小値から振幅値に変換し、設定されている引数 k で示す著大値と比較し、著大振幅を越える振幅が存在する場合、当該振幅範囲、(正傾斜でゼロを通過してから再び正傾斜でゼロを通過するまで)をデータ列から除去します。したがって、演算結果は著大値を除去したデータ個数分短くなります。</p> <p>なお、ゼロを通過しない場合は、除去しません。</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>$m=0$ $a_2 > k$</p> <p>除去区間</p> </div> <div style="text-align: center;"> <p>$m < 0$ $a_2 > k$</p> <p>戻り値 = X(m) X(n)</p> </div> </div> <p>演算例 上段⇒元波形 下段⇒除去後波形</p> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="margin-bottom: 20px;"> <p>元の波形</p> </div> <div> <p>著大値除去後の波形</p> </div> </div>

2. 13. 12. 数列連結関数

機能	X のデータ列に Y のデータ列を連結します。
文法	LNK(X,Y)
引数	<p>【X】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 連結対象数列1を記述します。</p> <p>【Y】<必須><即値/収録チャンネル/数値属性参照チャンネル/演算式> 連結対象数列2を記述します。</p>
記述例	LNK(\$1,\$2)
備考	<p>X の最終データの後ろに Y を連結した結果を戻します。結果データ個数は X のデータ個数+Y のデータ個数となります。</p> <p>\$1 ← 1,2,3,4,5,6,7 \$2 ← 8,9,10,11,12 \$3 = LNK(\$1,\$2) \$3 → 1,2,3,4,5,6,7,8,9,10,11,12</p>

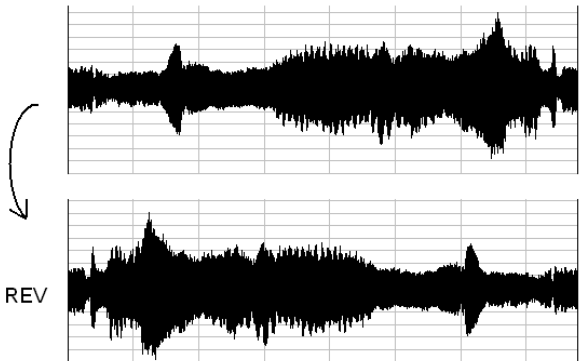
2. 13. 13. 数列縫合関数

機能	X のデータと Y のデータを交互に参照した結果を戻します。
文法	MGR(X,Y,k)
引数	<p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 縫合対象数列1を記述します。</p> <p>【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 縫合対象数列2を記述します。</p> <p>【k】<省略可><即値/数値属性参照チャンネル/演算式> 縫合順フラグを記述します。記述する縫合順フラグの内容は下記となります。 k=0 の時:記述省略した場合は、k=0 等価です。 数列 X と数列 Y の Index 順に縫合します。(備考1 参照) k=1 の時: 数列 X と数列 Y の大小関係が比較され、小さい方が優先して縫合されます。(備考2参照) k=-1 の時: 数列 X と数列 Y の大小関係が比較され、大きい方が優先して縫合されます。(備考3参照)</p>
記述例	MGR(\$1,#1)
備考	<p>備考1:(k=0 または省略時) 演算対象データ X と Y をデータ番号順に交互に参照して結果を戻します。演算結果のデータ個数は X のデータ個数+Y のデータ個数となります。なお、X のデータ個数と Y のデータ個数の何れか多い方は縫合されずそのまま連結されます。</p> <p>\$1 ← 1,2,3,4,5,6,7 \$2 ← 8,9,10,11,112,13,14,15 \$3 = MGR(\$1,\$2) \$3 → 1,8,2,9,3,10,4,11,5,12,6,13,7,14,15</p> <p>備考2:(k=1の時) 演算対象数列 X と Y の大小関係が比較され小さい方を優先して格納します。初めに X0<=Y0 が比較され成立すると X0 が格納され次に X1<=Y0 が比較され、成立すると X1 が格納され、その次は X2<=Y0 が比較され不成立の場合は Y0 が格納されます。数列 X、Y の何れも昇順並びの場合は格納結果も昇順並びとなります。</p> <p>\$1 ←1,3,5,9,8 \$2 ← 0,2,4,10,7, \$3 = MGR(\$1,\$2,1) \$3 →0,1,2,3,4,5,9,8,10,7 ※ \$5,\$4,\$6 の3個の数列を昇順並びで縫合する場合 \$3 = MGR(MRG(SRT(1,\$4),SRT(1,\$5)),SRT(1,\$6))</p> <p>備考3:(k=-1の時) 演算対象数列 X と Y の大小関係が比較され大きい方を優先して格納します。初めに X0>=Y0 が比較され成立すると Y0 が格納され次に X0>=Y1 が比較され、成立すると次の Y1 が格納され、その次は X0<=Y2 が比較され不成立の場合は X0 が格納されます。数列 X、Y の何れも降順並びの場合は格納結果も降順並びとなります。</p> <p>\$1 ←1,3,5,9,8 \$2 ← 0,2,4,10,7, \$3 = MGR(\$1,\$2,-1) \$3 →1,3,5,9,8,0,2,4,10,7 ※ \$5,\$4,\$6 の3個の数列を降順並びで縫合する場合 \$3 = MGR(MRG(SRT(-1,\$4),SRT(-1,\$5)),SRT(-1,\$6))</p>

2. 13. 14. 数列整列関数

機能	Y を X で指定したデータ番号に従って並び替えた結果を戻します。
文法	QUE(X,Y)
引数	【X】<必須><数値属性参照チャンネル/演算式> 並び替えるデータ番号数列を記述します。 X のデータ個数は、Y のデータ個数と等しくなければなりません。又、データ番号は唯一無二である必要があります。 【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列を記述します。
記述例	QUE(\$1,#1)
備考	他のチャンネルのデータの昇順、または降順に並び替える場合などに使用します。 \$1 = SRT(1,#1) ←#1 を昇順にした場合のデータ番号並びを取得 \$2 = QUE(\$1,#2) ←取得したデータ番号並び順に#2 を並び替えた結果を取得 \$1 ← 1,3,2,4,0 \$2 ← 1,2,3,4,5 \$2 = QUE(\$1,\$2) \$2 → 2,4,3,5,1

2. 13. 15. 数列反転関数

機能	数列 X の並び順を反転した結果を戻します。
文法	REV(X)
引数	【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 演算対象数列記述します。
記述例	REV(#1)
備考	演算対象データの並びを逆にした結果を戻します。  \$1 ← 1,2,3,4,5, \$1 = REV(\$1) \$1 → 5,4,3,2,1

2. 13. 16. 数列並び替え関数

機能	X のデータ列を並び替えた結果を戻します。並び替えモードは、昇順/降順以外に、データは並び替えないが、並び替えたとした場合のデータ番号を戻すことができます。										
文法	SRT(k,X)										
引数	<p>【k】<必須><即値/数値属性参照チャネル> 並び替えモード記述します。記述する並び替えモードを下表に示します。</p> <table border="1"> <thead> <tr> <th>K 値</th> <th>並び替えモード</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>データ昇順</td> </tr> <tr> <td>2</td> <td>データ昇順でデータ番号戻り</td> </tr> <tr> <td>-1</td> <td>データ降順</td> </tr> <tr> <td>-2</td> <td>データ降順でデータ番号戻り</td> </tr> </tbody> </table> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>	K 値	並び替えモード	1	データ昇順	2	データ昇順でデータ番号戻り	-1	データ降順	-2	データ降順でデータ番号戻り
K 値	並び替えモード										
1	データ昇順										
2	データ昇順でデータ番号戻り										
-1	データ降順										
-2	データ降順でデータ番号戻り										
記述例	SRT(1,#1)										
備考	<p>演算対象データを k で指定したモードに従って並び替えた結果を戻します。 k に 2 または 2 が指定された場合は、データの昇順または降順に並び替えたとした場合のデータ番号を戻します。</p> <p>\$1 ← 2,1,4,3,5 \$2 = SRT(1,\$1) \$2 → 1,2,3,4,5 \$2 = SRT(2,\$1) \$2 → 1,0,3,2,4 \$2 = SRT(-1,\$1) \$2 → 5,4,3,2,1 \$2 = SRT(-2,\$1) \$2 → 4,2,3,0,1</p>										

2. 13. 17. データ個数関数

機能	X のデータ個数を戻します。k で記述した区間毎のデータ個数を戻します。
文法	LEN(k,X)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式> 演算範囲を Index で記述します。即値、または要素 1 個の参照チャネルで記述した場合、指定した値から X で記述するデータ列の最終データ迄を演算範囲とし、複数要素の参照チャネルで記述した場合、要素で示す Index 範囲毎を演算範囲とします。なお、複数要素の場合、k の要素は昇順並びの必要があります。又、k で指定した Index が存在しない場合は、実行時 Error となります。</p> <p>記述省略した場合は、X で記述するデータ列の全て範囲を演算対象とします。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>
記述例	LEN(#1) LEN(\$ 1,#1)
備考	引数 k が省略された場合、或いは指定されても要素が 1 個の場合は、戻り値の要素数は 1 個となります。引数 k が記述された場合は、戻り値の要素数は k の要素数に依存し、例えば、要素数 3 の場合は戻り値の要素数は 2 個となり、その演算範囲は Ans1 = k(0)~k(1)-1 の範囲、Ans2 =k(1)~k(2)-1 の範囲のデータ個数値が戻ります。

2. 13. 18. 数列指定 Index 削除関数

機能	指定した Index の要素を削除します。
文法	DEL(k,X)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 削除する Index を記述します。記述する Index は唯一無二で削除対象数列に存在し、昇順並びの必要があります。存在していない Index が記述された場合、実行 error となります。</p> <p>【X】<必須><数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>
記述例	DEL(12,\$1) DEL(SRT(1,\$2),\$3) /* \$2 を昇順並びに整列させて設定*/
備考	<p>戻り数列の長さは、削除した要素数分短くなります。数列の全ての要素を削除することはできません。」</p> <p>\$1←2,4,6,8,10,12,14,16 \$2←1,3,5 \$3 = DEL(\$2,\$1) \$3→2,6,10,14,16 /*戻り数列の内容は指定された Index が削除された結果となる*/</p>

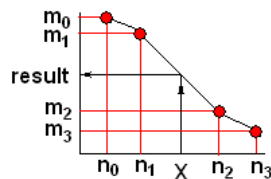
2. 13. 19. 数列同値削除関数

機能	演算対象数列から同じ値を削除し唯一無二の値で構成した数列を戻します。
文法	UNQ(X)
引数	【X】<必須><数値属性参照チャネル/演算式> 演算対象数列を記述します。データ並びは昇順で無ければなりません。
記述例	UNX(\$1)
備考	全て同じ値の場合は要素数1個となります。 また、データ並びが昇順並びで無い場合は、演算結果は保証されません。 演算対象数列のデータ並びが昇順並びでない場合は、数列並べ替え関数を使用して次の様に記述します。 UNQ(SRT(1,X))

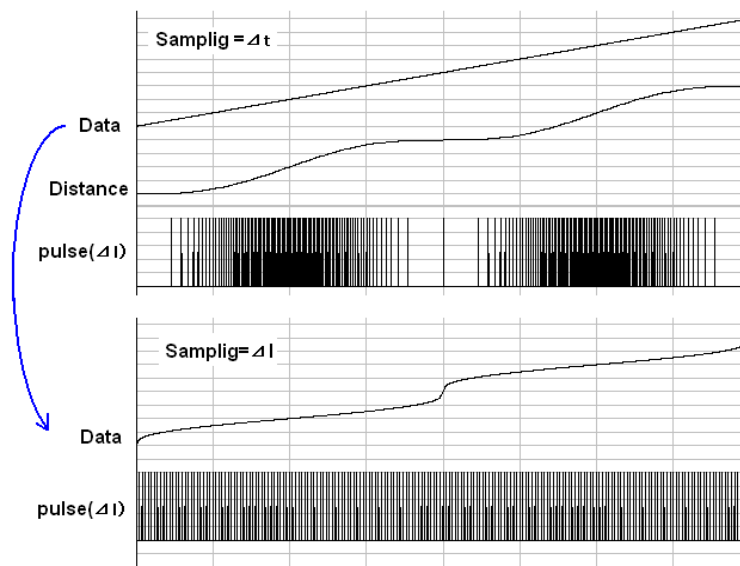
2. 13. 20. テーブル補間関数

機能	補間テーブルより直線補間した値を戻します。
文法	ITP(n,m,X)
引数	【n】<必須><数値属性参照チャネル/演算式> 補間テーブル X 軸値数列を記述します。n のデータ個数は、 $n > 2$ でなければなりません。また、データ値は、唯一無二で昇順並びでなければなりません。 【m】<必須><数値属性参照チャネル/演算式> 補間テーブル Y 軸値数列を記述します。m のデータ個数は、n のデータ個数と等しくなくてはならず、並び順は n のデータ番号順に対応した値でなければなりません。 【X】<必須><即値/収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。X のデータ属性は n と同じと見なします。
記述例	ITP(\$1,\$2,100)

備考 演算対象データ X のデータ番号順に、n で設定された補間テーブル X 軸列から該当する区間を求め、その両端のデータから直線補間して、X のデータ値に対応した値を求めて戻します。なお、X の値が補間テーブル X 軸の範囲に存在しない場合は直近の 2 点を延長した直線補外して求めて戻します。結果のデータ個数は演算対象データの個数と等しくなります。



例えば、頻度解析結果から S-N 近似線テーブルを与え被害度を演算する場合や、他の関数と組み合わせて時間サンプリングした走行距離データから等距離サンプリングに変換する処理に適用できます。



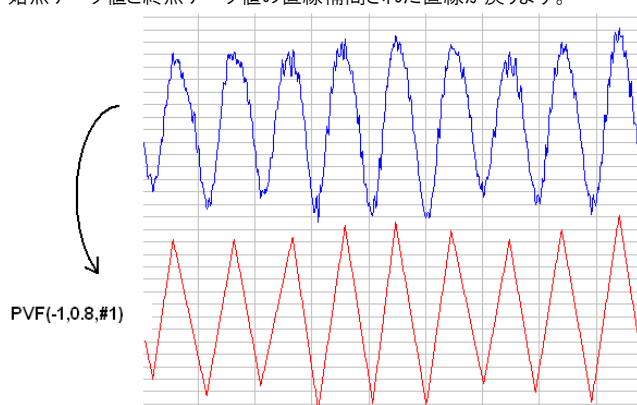
2. 13. 21. 戻り値選択関数

機能	引数 k により戻る値を X または Y とします。															
文法	RVS(k,X,Y)															
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 戻り値を X とするか Y とするか選択コードを記述します。</p> <table border="1"> <thead> <tr> <th>戻り値選択コード</th> <th>比較演算</th> <th>戻り値</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>-</td> <td>k(i)=0 のとき X(i) を戻す。k の個数が 1 のときは X 全体を戻す。</td> </tr> <tr> <td>1</td> <td>-</td> <td>K(i)=1 のとき Y(i) を戻す。k の個数が 1 のときは Y 全体を戻す。</td> </tr> <tr> <td>2</td> <td>X>Y</td> <td>不成立時: X を戻し、成立時: Y を戻す</td> </tr> <tr> <td>3</td> <td>X<Y</td> <td>不成立時: X を戻し、成立時: Y を戻す</td> </tr> </tbody> </table> <p>フラグ k が 0 または 1 で複数要素を持つ場合は、要素ごとに 0/1 を判別しますので、k,X,Y の要素数は等しい必要があります。フラグ k が 2 または 3 の時は、k が複数要素を持つ数列で記述しても参照されるのは Index0 のみとなります。k は 0~3 の範囲で無ければなりません。それ以外の場合、実行時 Error となります。</p> <p>【X】<必須><即値/数値属性参照チャネル/演算式> 戻す値を記述します。</p> <p>【Y】<必須><即値/数値属性参照チャネル/演算式> 戻す値を記述します。</p>	戻り値選択コード	比較演算	戻り値	0	-	k(i)=0 のとき X(i) を戻す。k の個数が 1 のときは X 全体を戻す。	1	-	K(i)=1 のとき Y(i) を戻す。k の個数が 1 のときは Y 全体を戻す。	2	X>Y	不成立時: X を戻し、成立時: Y を戻す	3	X<Y	不成立時: X を戻し、成立時: Y を戻す
戻り値選択コード	比較演算	戻り値														
0	-	k(i)=0 のとき X(i) を戻す。k の個数が 1 のときは X 全体を戻す。														
1	-	K(i)=1 のとき Y(i) を戻す。k の個数が 1 のときは Y 全体を戻す。														
2	X>Y	不成立時: X を戻し、成立時: Y を戻す														
3	X<Y	不成立時: X を戻し、成立時: Y を戻す														
記述例	RVS(2,\$1+\$2,\$1-\$2)															
備考	<p>フラグ k が 0 で複数要素の時は、k の要素ごとに参照され X(i) または Y(i) が要素ごとに選択されて戻ります。フラグ k が 2 または 3 の時は、X と Y の要素ごとに比較演算され、X(i) または Y(i) が要素ごとに選択されて戻ります。また、フラグ k が 0 または 1 の場合で要素数が 1 の場合は、戻り数列の要素数は戻り値の要素数に依存します。</p> <p>\$3 = (\$1+\$4)/4*EQU(\$1,\$2)+(\$1+\$5)/3*NEQ(\$1,\$2) の演算式は \$3 = RVS(EQU(\$1,\$2),(\$1+\$5)/3,(\$1+\$4)/4) として記述できます。 フローチャートで示せば下図と等価です。\$1 と \$2 の要素毎に比較演算され TRUE の場合は当該 Index n 位置に (\$1(n)+\$4(n))/4 が FALSE の場合は (\$1(n)+\$5(n))/3 が代入されます。</p>															

2. 13. 22. 収録時間経過関数

機能	現在のサンプリング周期から経過時間データ列を戻します。
文法	SPB(k)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 経過時間初期値を意味し、即値、先頭文字 "\$" に続きチャネル番号で表す数値属性参照チャネル、または演算式で記述します。なお、記述する時間の単位は sec です。</p>
記述例	SPB(0)
備考	<p>現在設定されている解析範囲の経過時間を初期値 k に加算して戻します。右肩上がりの直線となります。結果のデータ個数は現在解析範囲に指定されているデータ個数と同じ個数となります。</p> <p>SPB(k) は、収録チャネルのデータ個数を利用して、(ACC((#1*0)+1)-1)*PRD()+k と同じ意味を持ちます。</p> <p>この関数は、例えば、時間軸サンプリングから距離軸サンプリングに変更する処理などを行う場合、経過時間として意味を持ちます。</p> <p>なお、生成されるデータ個数は、現在の解析範囲に指定されている範囲のデータ個数となります。解析範囲が指定されていない場合は使用できません。</p>

2. 13. 23. 山谷検索関数

機能	演算対象波形の山谷値または山谷位置データ番号を検索します。																
文法	PVF(k,m,X)																
引数	<p>[k]<必須><即値/数値属性参照チャネル> 演算モードフラグを記述します。記述可能なフラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>k の値</th> <th>演算種別</th> </tr> </thead> <tbody> <tr> <td>-1</td> <td>Peak/Valley 値を直線補間した波形が戻ります。</td> </tr> <tr> <td>0</td> <td>Peak/Valley 位置データ番号が戻ります。</td> </tr> <tr> <td>1</td> <td>Peak/Valley 位置の値が戻ります。</td> </tr> <tr> <td>2</td> <td>Peak 位置データ番号が戻ります。</td> </tr> <tr> <td>3</td> <td>Peak 位置の値が戻ります。</td> </tr> <tr> <td>4</td> <td>Valley 位置データ番号が戻ります。</td> </tr> <tr> <td>5</td> <td>Valley 位置の値が戻ります。</td> </tr> </tbody> </table> <p>[m]<必須><即値/数値属性参照チャネル/演算式> 無効振幅値を記述します。無効振幅値は正数(0<m)でなければなりません。 PeakValley 検出後、隣接した Peak/Valley(Valley/Peak)のペアから求めた振幅値がここで設定された無効振幅値 m を越えて居ない場合は当該ペアが除去されます。</p> <p>[X]<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>	k の値	演算種別	-1	Peak/Valley 値を直線補間した波形が戻ります。	0	Peak/Valley 位置データ番号が戻ります。	1	Peak/Valley 位置の値が戻ります。	2	Peak 位置データ番号が戻ります。	3	Peak 位置の値が戻ります。	4	Valley 位置データ番号が戻ります。	5	Valley 位置の値が戻ります。
k の値	演算種別																
-1	Peak/Valley 値を直線補間した波形が戻ります。																
0	Peak/Valley 位置データ番号が戻ります。																
1	Peak/Valley 位置の値が戻ります。																
2	Peak 位置データ番号が戻ります。																
3	Peak 位置の値が戻ります。																
4	Valley 位置データ番号が戻ります。																
5	Valley 位置の値が戻ります。																
記述例	PVF(0,2,\$1)																
備考	<p>演算対象波形の先頭から、山谷を検索します。検出した山谷データから無効振幅除去を行った後、 k=-1 の時は、残った山谷位置の値で、直線補間した結果を格納します。結果のデータ個数は、X のデータ個数となり変化しません。 k=0 の時は、残った山谷位置のデータ番号を格納します。結果のデータ個数は、山谷の検出個数 k=1 の時は、残った山谷位置の値を格納します。結果のデータ個数は、山谷の検出個数 k=2 の時は、残った山位置のデータ番号を格納します。結果のデータ個数は、山の検出個数 k=3 の時は、残った山位置の値を格納します。結果のデータ個数は、山の検出個数 k=4 の時は、残った谷位置のデータ番号を格納します。結果のデータ個数は、谷の検出個数 k=5 の時は、残った谷位置の値を格納します。結果のデータ個数は、谷の検出個数 なお、先頭データは山、または谷として使用されます。山とする場合は先頭から次のデータが負傾斜になる場合、谷とする場合は先頭から次のデータが正傾斜となる場合です。 検出できなかった時の処理は、設定した演算種別フラグ k によって異なり。 k=0,2,4 の時は、-1 が戻ります。 k=1,3,5 の時は、演算 Error となります。 k=-1 の時は、始点データ値と終点データ値の直線補間された直線が戻ります。</p> 																

2. 13. 24. 飛越連結関数

機能	数列同士を Index 順に連結した数列を生成します。
文法	INL(X1,X2,X3,,,X10)
引数	<p>[X]<必須><数値属性参照チャネル/演算式> 連結する数列を記述します。記述可能な数列の数は最大 10 個までとなります。但し、記述する数列の要素数は等しい必要があります。記述した数列で要素数が異なる数列を含む場合は実行時 ERROR となります。</p>
記述例	INL(\$1,\$2,\$3,\$4)
備考	<p>戻り数列の要素数は記述した数列の合計要素数となります。 assign \$1 = 1,2,3,4 assign \$2 = 5,6,7,8 \$3 = INL(\$1,\$1,\$2,\$2) \$3 の要素値は index 順に、1,1,5,5,2,2,6,6,3,3,7,7,4,4,8,8 と戻ります。</p>

2. 14. Index 検索関数

2. 14. 1. 閾値通過 Index 検索関数 1

機能	設定した閾値を越えたデータ番号を戻します。												
文法	DTD(k,m,j,n,X)												
引数	<p>【k】<必須><即値/数値属性参照チャネル> 閾値通過方向フラグを記述します。記述する閾値通過方向フラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>k 値</th> <th>種別</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>閾値を下降で過る</td> </tr> <tr> <td>1</td> <td>閾値を上昇で過る</td> </tr> </tbody> </table> <p>【m】<必須><即値/数値属性参照チャネル/演算式> 閾値を記述します。</p> <p>【j】<必須><即値/数値属性参照チャネル> 検索方向フラグを記述します。記述する検索方向フラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>j 値</th> <th>検索方向</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Index 逆順(時間軸逆方向)</td> </tr> <tr> <td>1</td> <td>Index 順(時間軸方向)</td> </tr> </tbody> </table> <p>【n】<必須><即値/数値属性参照チャネル/演算式> 検索開始データ番号を記述します。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 検索対象数列を記述します。</p>	k 値	種別	0	閾値を下降で過る	1	閾値を上昇で過る	j 値	検索方向	0	Index 逆順(時間軸逆方向)	1	Index 順(時間軸方向)
k 値	種別												
0	閾値を下降で過る												
1	閾値を上昇で過る												
j 値	検索方向												
0	Index 逆順(時間軸逆方向)												
1	Index 順(時間軸方向)												
記述例	DTD(1,\$1,1,\$2,#1)												
備考	<p>n で示されたデータ番号位置を起点とし j で設定された検索方向で検索開始し、k で指定された閾値通過方向を m で示された閾値を越えた最初のデータ番号を戻します。結果のデータ個数は 1 個となります。 存在しない場合は、-1 を戻します。</p> <p>閾値と等しい場合は閾値を越えていないと判断しますので、戻りデータ番号-1 は閾値通過直前のデータ番号、または閾値と等しいデータ番号となります。</p>												

2. 14. 2. 閾値通過 Index 検索関数 2

機能	閾値通過データ番号関数 1 (DTD 関数) が閾値を越えた最初のデータ番号を戻すのに対して、閾値通過データ番号関数 2 (DTM 関数) は閾値を越えた毎のデータ番号を戻します。												
文法	DTM(k,m,j,n,X)												
引数	<p>【k】<必須><即値/数値属性参照チャンネル> 閾値通過方向フラグを記述します。記述する閾値通過方向フラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>k 値</th> <th>種別</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>閾値を下降で過る</td> </tr> <tr> <td>1</td> <td>閾値を上昇で過る</td> </tr> </tbody> </table> <p>【m】<必須><即値/数値属性参照チャンネル/演算式> 閾値または差分値を記述します。o: 閾値か差分値かの種別フラグが 0 以外の場合差分値を意味し検索閾値はここで記述された値に検索開始データ番号の値を加算されます。</p> <p>【j】<必須><即値/数値属性参照チャンネル> 検索方向フラグを記述します。記述する検索方向フラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>j 値</th> <th>検索方向</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Index 逆順(時間軸逆方向)</td> </tr> <tr> <td>1</td> <td>Index 順(時間軸方向)</td> </tr> </tbody> </table> <p>※ 検索方向は複数要素で指定されても参照される Index は 0 のみとなります。</p> <p>【n】<必須><即値/数値属性参照チャンネル/演算式> 検索開始データ番号記述します。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 検索対象数列を記述します。</p> <p>【o】<省略可><即値/数値属性参照チャンネル・演算式> 閾値または差分値種別フラグを記述します。 m:=0 の場合、m は閾値を意味し、 m<>0 の場合は差分値を意味します。 なお、記述省略された場合は 0 と同じ意味を持ちます。</p>	k 値	種別	0	閾値を下降で過る	1	閾値を上昇で過る	j 値	検索方向	0	Index 逆順(時間軸逆方向)	1	Index 順(時間軸方向)
k 値	種別												
0	閾値を下降で過る												
1	閾値を上昇で過る												
j 値	検索方向												
0	Index 逆順(時間軸逆方向)												
1	Index 順(時間軸方向)												
記述例	DTM(1,\$1,1,\$2,#1)												
備考	<p>検索方向(j)以外の引数は複数要素を持つ配列で記述できます。 閾値通過方向(k)、閾値(m)、検索開始データ番号(n)がそれぞれ複数要素を持つ配列型で指定された場合で、要素数が異なる場合は、最も多い要素数以外の引数は、複数要素であっても参照される Index は 0 のみとなります。 検索は、設定された検索方向にその要素数回行われ、検索毎の最初に成立したデータ番号を戻します。成立しない場合は、-1 が戻ります。戻りのデータ個数は検索回数(要素数の最も多い引数の数分)となります。 引数の要素数が何れも 1 個の場合、n で示されたデータ番号位置を起点とし j で設定された検索方向で検索開始し、k で指定された閾値通過方向を m で示された閾値を越える毎にデータ番号を戻します。戻りのデータ個数は閾値を過ぎた個数となります。存在しない場合は-1 が戻ります。</p> <p>閾値と等しい場合は、閾値を越えて居ないと判断しますので、戻りデータ番号-1 が閾値通過直前のデータ番号、または閾値と等しいデータ番号となります。</p>												

2. 14. 3. 閾値通過持続 Index 検索関数1

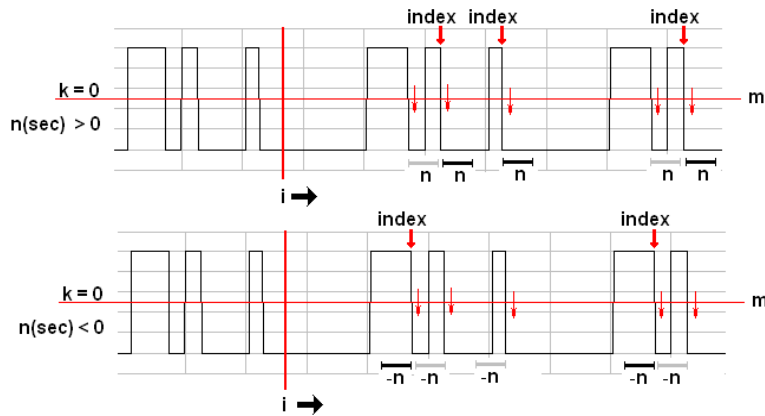
機能	設定した閾値を通過後、同じく設定した閾値許容範囲を持続した時の閾値通過データ番号を戻します。														
文法	CTD(k,m,j,s,n,d,X)														
引数	<p>【k】<必須><即値/数値属性参照チャンネル> 閾値通過判定フラグを記述します。記述する閾値通過フラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>k 値</th> <th>種別</th> </tr> </thead> <tbody> <tr> <td>-1</td> <td>閾値 m-許容範囲 d を過る</td> </tr> <tr> <td>0</td> <td>閾値 m を過る</td> </tr> <tr> <td>1</td> <td>閾値 m+許容範囲 d を過る</td> </tr> </tbody> </table> <p>閾値通過方向は関係なく、上昇または下降の何れかで過ぎった時に閾値通過したものを見なします。</p> <p>【m】<必須><即値/数値属性参照チャンネル/演算式> 閾値を記述します。</p> <p>【j】<必須><即値/数値属性参照チャンネル> 検索方向フラグを記述します。記述する検索方向フラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>j 値</th> <th>検索方向</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Index 逆順(時間軸逆方向)</td> </tr> <tr> <td>1</td> <td>Index 順(時間軸方向)</td> </tr> </tbody> </table> <p>【s】<必須><即値/数値属性参照チャンネル/演算式> 検索開始データ番号を意味し、即値、先頭文字"\$"に続きチャンネル番号で記述する数値属性参照チャンネル、または演算式で記述します。</p> <p>【n】<必須><即値/数値属性参照チャンネル/演算式> 閾値通過後のデータ個数を記述します。</p> <p>【d】<必須><即値/数値属性参照チャンネル/演算式> 閾値の許容範囲を記述します。 閾値の許容範囲 d は、m で指定された閾値通過判定時に $m \pm d$ で判定されると同時に閾値通過後保持時間内が $m \pm d$ であること を意味します。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 検索対象数列を記述します。</p>	k 値	種別	-1	閾値 m-許容範囲 d を過る	0	閾値 m を過る	1	閾値 m+許容範囲 d を過る	j 値	検索方向	0	Index 逆順(時間軸逆方向)	1	Index 順(時間軸方向)
k 値	種別														
-1	閾値 m-許容範囲 d を過る														
0	閾値 m を過る														
1	閾値 m+許容範囲 d を過る														
j 値	検索方向														
0	Index 逆順(時間軸逆方向)														
1	Index 順(時間軸方向)														
記述例	CTD(\$1,\$2.0,\$3,\$4,\$5,#1)														
備考	<p>検索対象データの s で設定されたデータ番号から j で設定された方向に検索を開始し、$m \pm d$ 閾値を越えた地点から n で設定された保持データ個数以上、閾値 m の土許容範囲 d を保持した時に成立し、閾値許容範囲内に達した地点のデータ番号を戻します。閾値通過判定は、k=1 の時は、$m-d$ を越えた地点、k=-1 の時は $m+d$ を越えた地点、k=0 の時は $m \pm d$ を越えた地点を閾値通過地点とします。戻り値は、この通過地点のデータ番号を戻り値とします。戻り結果データ個数は、s で設定される開始番号が複数要素を持つ場合、要素毎に最初の成立点を戻し、結果のデータ個数は s のデータ個数となります。また s の要素数(データ数)が 1 の場合は検出した個数となります。解析範囲に存在しない場合は-1 を戻します。</p>														

2. 14. 4. 閾値通過持続 Index 検索関数 2

機能	設定した閾値を越えた状態が設定したデータ個数分保持された時のデータ番号を戻します。						
文法	DTE(k,m,n,X)						
引数	<p>【k】<必須><即値/数値属性参照チャネル> 閾値通過方向フラグを記述します。</p> <table border="1"> <thead> <tr> <th>k 値</th> <th>種別</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>閾値を下降で過る</td> </tr> <tr> <td>1</td> <td>閾値を上昇で過る</td> </tr> </tbody> </table> <p>【m】<必須><即値/数値属性参照チャネル/演算式> 閾値を記述します。</p> <p>【n】<必須><即値/数値属性参照チャネル/演算式> 閾値通過後の持続データ個数を記述します。 閾値通過持続データ個数は m で指定された閾値を k で指定された方向で過ぎてから、閾値通過状態を継続するデータ個数です。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 検索対象数列を記述します。</p>	k 値	種別	0	閾値を下降で過る	1	閾値を上昇で過る
k 値	種別						
0	閾値を下降で過る						
1	閾値を上昇で過る						
記述例	DTE(0,\$1,\$2,#1)						
備考	<p>m で設定した閾値を k で設定した方向に過ぎた状態が n で指定したデータ個数分持続した時の閾値通過位置のデータ番号を戻します。なお、戻りデータ番号は閾値通過直前のデータ番号となります。</p> <p>なお、閾値通過後持続データ個数の符号は意味を持ちます。閾値通過後持続データ個数 n が、n<0 の場合は閾値通過後の持続データ個数(n の絶対値)に満たずに閾値を過ぎても無視し、その後、閾値通過後のデータ持続個数を満たすと、先に検出したデータ番号を戻します。結果の戻りデータ個数は閾値を越えた有効回数分となります。存在しない場合は-1 を戻します。</p>						

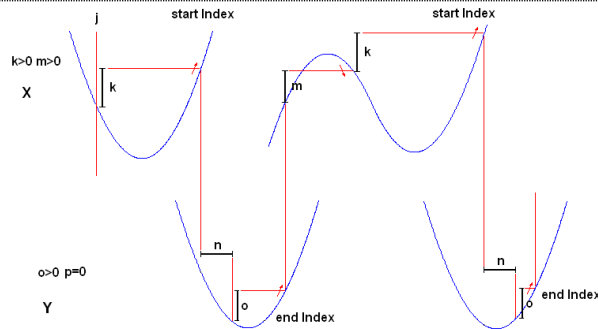
2. 14. 5. 閾値通過持続 Index 検索関数 3

機能	設定した閾値を越えた状態が設定した時間持続した時の閾値通過データ番号または、閾値を越える直前に設定した時間閾値を越えなかった時のデータ番号を戻します。						
文法	DTF(j,k,m,n,X)						
引数	<p>【j】<必須><即値/数値属性参照チャンネル/演算式> 検索開始データ番号記述します。なお、X で指定される検索対象数列の Index 範囲の必要があります。</p> <p>【k】<必須><即値/数値属性参照チャンネル> 閾値を過る方向フラグを記述します。記述する方向フラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>k 値</th> <th>種別</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>閾値を下降で過る</td> </tr> <tr> <td>1</td> <td>閾値を上昇で過る</td> </tr> </tbody> </table> <p>【m】<必須><即値/数値属性参照チャンネル/演算式> 閾値を記述します。</p> <p>【n】<必須><即値/数値属性参照チャンネル/演算式> 閾値通過後の持続時間(単位=sec)を記述します。なお、nの符号は意味を持ち、n>0の場合は、閾値を過ぎてから持続する時間を意味し、n<0の場合は、閾値を過る前に持続していた時間を意味します。</p> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 検索対象数列を記述します。</p>	k 値	種別	0	閾値を下降で過る	1	閾値を上昇で過る
k 値	種別						
0	閾値を下降で過る						
1	閾値を上昇で過る						
記述例	DTF(\$1.0,\$2,\$3,#1)						
備考	<p>j で設定された検索開始番号から m で設定した閾値を k で設定した方向に過ぎった地点(検出点)を探し、n>0 の場合はその地点から n 時間持続した場合、有効として過った地点-1 を抽出し、n<0 の場合はその地点から n 時間遡った範囲継続した場合、有効として過ぎって地点-1 を抽出します。</p> <p>存在しない場合は、-1 が戻ります。</p>						



2. 14. 6. 2 信号閾値通過 Index 検索関数

機能	検索対象チャンネルを 2ch 分設定し、指定したチャンネルの閾値越えた位置のデータ番号を検索し、次のデータ番号は、もう一方のチャンネルの閾値を越えたデータ番号を検索し、相補的に開始データ番号終了データ番号を検索します。																								
文法	DTC(j,k,m,X,n,o,p,Y)																								
引数	<p>【j】<必須><即値/数値属性参照チャンネル> 開始点検索開始データ番号を意味し、即値、先頭文字"\$"に続きチャンネル番号で表す数値属性参照チャンネル、または演算式で記述します。</p> <p>【k】<必須><即値/数値属性参照チャンネル/演算式> 開始変化閾値を記述します。開始閾値は開始点検索対象データ X の検索開始データ番号位置の値に開始閾値変化 k を加算となります。開始点閾値通過方向は、k の符号により決定されます。</p> <table border="1"> <tr> <td>k 値</td> <td>初回開始点閾値通過方向</td> </tr> <tr> <td>k<0</td> <td>閾値を下降で過る</td> </tr> <tr> <td>k>0</td> <td>閾値を上昇で過る</td> </tr> </table> <p>【m】<必須><即値/数値属性参照チャンネル/演算式> 次回以降の検索開始データ番号検索閾値を記述します。初回以降の開始データ番号は、終了点から、ここで設定した検索開始データ番号閾値を越えた地点を開始点検索データ番号として、開始点を検索します。なお、開始点閾値は、開始点検索対象データ X の検索開始データ番号位置の値に開始閾値変化量を加算した値となります。又、閾値通過方向は、m の符号により決定されます。</p> <table border="1"> <tr> <td>m 値</td> <td>開始点閾値通過方向</td> </tr> <tr> <td>m<0</td> <td>閾値を上昇で過る</td> </tr> <tr> <td>m>0</td> <td>閾値を下降で過る</td> </tr> </table> <p>【X】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 開始点検索対象数列を記述します。</p> <p>【n】<必須><即値/数値属性参照チャンネル/演算式> 開始点通過後の遅延データ個数を記述します。開始点通過データ番号+遅延データ個数=終了点検索開始データ番号を意味します。</p> <p>【o】<必須><即値/数値属性参照チャンネル/演算式> 終了閾値変化量を記述します。終了閾値は終了点検索対象データ Y の終了点検索開始データ番号位置の値に終了閾値変化量を加算した値となります。終了点閾値通過方向は、o の符号により決定されます。</p> <table border="1"> <tr> <td>o 値</td> <td>終了点閾値通過方向</td> </tr> <tr> <td>o<0</td> <td>閾値を下降で過る</td> </tr> <tr> <td>o>0</td> <td>閾値を上昇で過る</td> </tr> </table> <p>【p】<必須><即値/数値属性参照チャンネル> 終了点検索方向フラグを記述します。記述する終了点検索方向フラグを下表に示します。</p> <table border="1"> <tr> <td>p 値</td> <td>終了点検索方向</td> </tr> <tr> <td>0</td> <td>Index 順</td> </tr> <tr> <td>1</td> <td>Index 逆順</td> </tr> </table> <p>なお、開始点検索方向は Index 順(時間軸方向)に固定されています。</p> <p>【Y】<必須><収録チャンネル/数値属性参照チャンネル/演算式> 終了点検索対象数列を記述します。</p>	k 値	初回開始点閾値通過方向	k<0	閾値を下降で過る	k>0	閾値を上昇で過る	m 値	開始点閾値通過方向	m<0	閾値を上昇で過る	m>0	閾値を下降で過る	o 値	終了点閾値通過方向	o<0	閾値を下降で過る	o>0	閾値を上昇で過る	p 値	終了点検索方向	0	Index 順	1	Index 逆順
k 値	初回開始点閾値通過方向																								
k<0	閾値を下降で過る																								
k>0	閾値を上昇で過る																								
m 値	開始点閾値通過方向																								
m<0	閾値を上昇で過る																								
m>0	閾値を下降で過る																								
o 値	終了点閾値通過方向																								
o<0	閾値を下降で過る																								
o>0	閾値を上昇で過る																								
p 値	終了点検索方向																								
0	Index 順																								
1	Index 逆順																								
記述例	DTC(\$1,\$2,\$3,#1,\$4,\$5,0,\$4)																								
備考	<p>開始点検索対象データ列から開始点を検索、終了点検索対象データ列から終了点を検索します。検索結果は、開始点データ番号と終了点データ番号が交互に求まります。</p> <p>開始点閾値は終了点検索結果に依存するため、初回開始点閾値と以降の開始点閾値は検索毎に異なります。同様に終了点閾値も検索毎に異なります。検索条件に一致する条件が存在しない場合は、-1 が戻ります。</p>																								



2. 14. 7. 直近山谷 Index 検索関数

機能 閾値を越えた直後の山、または谷のデータ番号を返します。

文法 DTP(k,m,X)

引数 【k】<必須><即値/数値属性参照チャネル>

検索モードフラグ記述します。記述する検索モードを下表に示します。

K 値	検索方向	検索対象位置
0	Index 逆順	最初の谷
1	Index 逆順	最初の山
2	Index 順	最初の谷
3	Index 順	最初の山

【m】<必須><即値/数値属性参照チャネル/演算式>

検索開始データ番号を記述します。なお、検索開始データ番号 m が複数の要素を持つ場合、検索はその要素毎に行われます。

【X】<必須><収録チャネル/数値属性参照チャネル/演算式>

検索対象数列を記述します。

記述例 DTP(0,\$1,#1)

備考

k = 0 の時

X で与えられる波形データの m で与えられるデータ番号から時間軸をさかのぼり、最初に出会う谷底データ番号を返します。一定値が連続する場合、検索方向から見て一定値となる一番手前のデータ番号を返します。

k = 1 の時

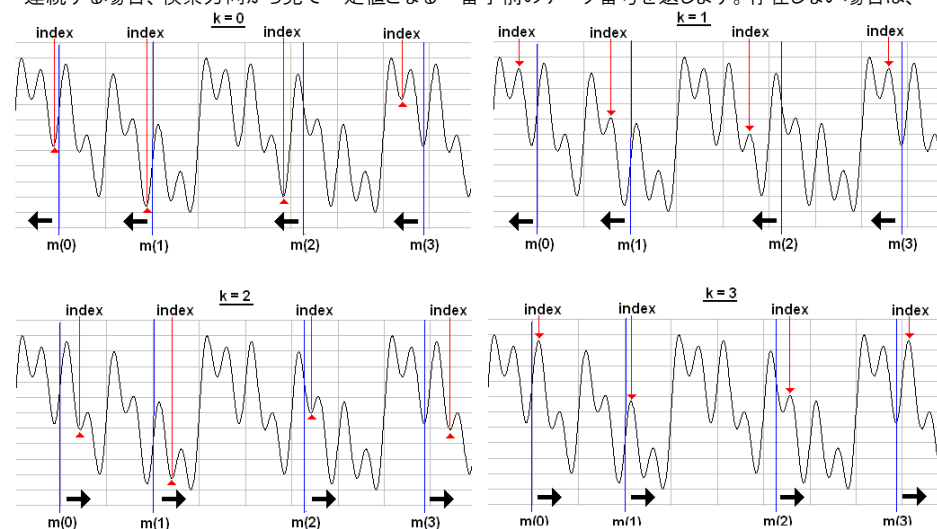
検索対象データ列の m で与えられるデータ番号から時間軸をさかのぼり、最初に出会う山頂データ番号を返します。一定値が連続する場合、検索方向から見て一定値となる一番手前のデータ番号を返します。

k = 2 の時

検索対象データ列波形データの m で与えられるデータ番号から時間軸を進み、最初に出会う谷底データ番号を返します。一定値が連続する場合、検索方向から見て一定値となる一番手前のデータ番号を返します。

k = 3 の時

検索対象データ列波形データの m で与えられるデータ番号から時間軸を進み、最初に出会う山頂データ番号を返します。一定値が連続する場合、検索方向から見て一定値となる一番手前のデータ番号を返します。存在しない場合は、-1 を返します。



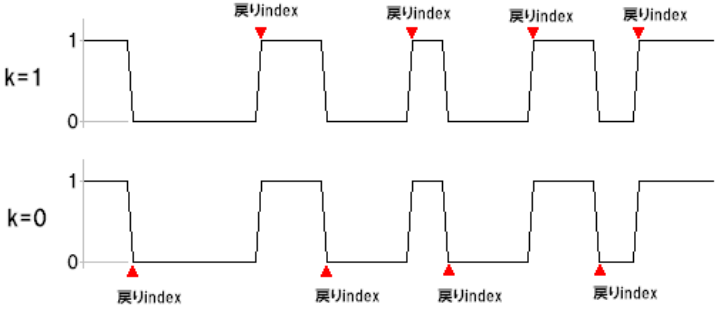
2. 14. 8. 値一致 Index 検索関数

機能	与えた値と一致したデータ番号、または一致データ個数を返します。								
文法	EQP(k,m,X)								
引数	<p>【k】<必須><即値/数値属性参照チャネル> 演算モードフラグを記述します。記述する演算モードを下表に示します。</p> <table border="1"> <thead> <tr> <th>k</th> <th>演算内容</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>一致したデータ番号を返し、一致しなかったデータ番号位置は-1を返す。</td> </tr> <tr> <td>1</td> <td>一致したデータ数を返す。</td> </tr> <tr> <td>2</td> <td>一致したデータ番号を返し、データ番号が1個でも一致しない場合は、-1を返す。</td> </tr> </tbody> </table> <p>【m】<必須><即値/数値属性参照チャネル/演算式> 検索データ値を記述します。複数の要素を持つ数値列で記述した場合、データは唯一無二でなければならない。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 検索対象数値列を記述します。</p>	k	演算内容	0	一致したデータ番号を返し、一致しなかったデータ番号位置は-1を返す。	1	一致したデータ数を返す。	2	一致したデータ番号を返し、データ番号が1個でも一致しない場合は、-1を返す。
k	演算内容								
0	一致したデータ番号を返し、一致しなかったデータ番号位置は-1を返す。								
1	一致したデータ数を返す。								
2	一致したデータ番号を返し、データ番号が1個でも一致しない場合は、-1を返す。								
記述例	EQP(0,\$1,\$2)								
備考	<p>検索データ値 m の要素毎に、検索対象データの先頭から並び順に m のデータと X のデータが等しい X の地点(データ番号)を検索します。</p> <p>k=0 の時は、等しい場合そのデータ番号を格納し、存在しない場合は、-1 を格納します。 なお、等しいデータ番号が複数存在した場合は、先頭のデータ番号を格納します。結果のデータ個数は、検索データ m の個数と一致します。</p> <p>k=1 の時は、等しい個数を格納します。 存在しない場合は 0 を格納します。結果のデータ個数は検索データ m の個数と同じになります。</p> <p>k=2 の時は、k=0 と基本的に同じ動作をしますが、m 要素に一致するデータが一つでも存在しない場合は-1 を返します。 その場合、結果のデータ個数は 1 個となります。</p>								

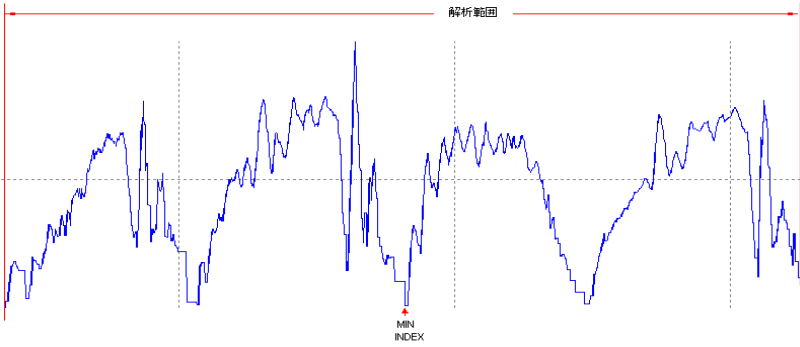
2. 14. 9. データ照合関数

機能	X に Y が含まれているかを検査します。指定する k により、存在したデータ個数、または X の位置を返します。
文法	FND(k,X,Y)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 戻り値の種別フラグを記述します。 k=0 の時は、X に Y が存在した個数、存在しない場合は 0 が戻ります。 k=1 の時は、X に Y が存在した X の Index が戻り、存在しない場合は-1 が戻ります。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 被検索対象数値列を記述します。</p> <p>【Y】<必須><即値/収録チャネル/数値属性参照チャネル/演算式> 照合数値列を記述します。</p>
記述例	FND(0,\$1,\$2) FND(1,#1,\$2)
備考	<p>設定する k=1 の時、X に Y が複数存在した場合で Y が複数要素の場合は、最初に一致した X の Index を返します。又、Y の要素が 1 個の場合は一致した全ての X の Index を返します。</p> <p>演算例 1: 照合数値列が複数要素の場合 X(0)=1, x(1)=2, X(2)= 3, X(3)=3, x(4) = 5 Y(0)=3 Y(1) = 4 とした場合 k=0(存在データ個数を求める場合) Ans = CPM(0,X,Y)の結果は、Ans(0)⇒2、Ans(1)⇒0 が戻ります。 Y(0)が X に 2 個存在した。Y(1)は X に存在しないことを意味します。 k=1(存在した X の Index を求める場合) Ans = CMP(1,X,Y)の結果は、Ans(0)⇒2、Ans(1)⇒-1 が戻ります。 Y(0)の 3 が X の Index 2 に存在、Y(1)の 4 は X に存在しないことを意味します。 注: データ値 3 は X に 2 個存在していますが、戻り Index は初めに一致した Index のみとなります。</p> <p>演算例 2: 照合数値列が単一要素の場合 X(0)=1, x(1)=2, X(2)= 3, X(3)=3, x(4) = 5 Y(0)=3 とした場合 k=0(存在データ個数を求める場合) Ans = CPM(0,X,Y)の結果は、Ans(0)⇒2 k=1(存在した X の Index を求める場合) Ans = CMP(1,X,Y)の結果は、Ans(0)⇒2 Ans(1)⇒3 が戻ります。 注 データ値 3 が X の Index 2 と Index 3 に存在したことを意味します。</p>

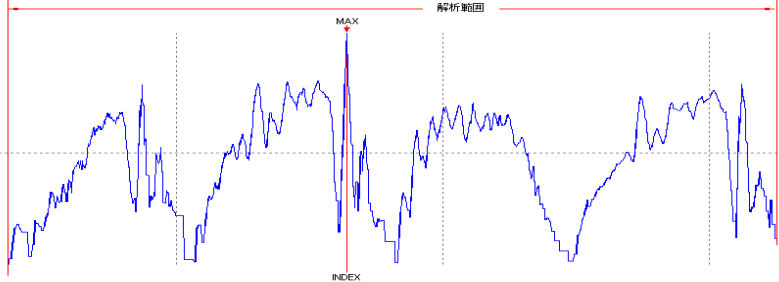
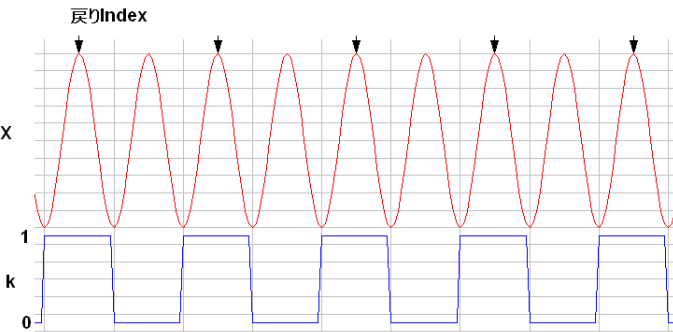
2. 14. 10. 論理値遷移 Index 検索関数

機能	X の論理値が"0"⇒"1"に遷移したデータ番号、または"1"⇒"0"に遷移したデータ番号を戻します。
文法	LST(k,X)
引数	<p>[k]<必須><即値/数値属性参照チャネル/演算式> 遷移方向フラグを記述します。 k=0 の時、論理値が"1"⇒"0"に遷移したデータ番号が戻ります。 k=1 の時、論理値が"0"⇒"1"に遷移したデータ番号が戻ります。</p> <p>[X]<必須><即値/数値属性参照チャネル/演算式> 演算対象論理数列を記述します。論理数列は、"0"または"1"で構成された論理値からなる数列を意味します。</p>
記述例	LST(0,\$2)
備考	<p>k=1 は、$(X_{n-1} <= "0") \& (X_n >= "1")$の時の Index n を戻し、 k=0 は、$(X_{n-1} >= "1") \& (X_n <= "0")$の時の Index n を戻す。</p> 

2. 14. 11. 最小値 Index 検索関数

機能	X の最小値位置のデータ番号を戻します。
文法	MNP(k,X)
引数	<p>[k]<省略可><即値/数値属性参照チャネル/演算式> 検索対象範囲を記述します。検索対象範囲は記述した k の要素数により意味が異なります。又、省略記述した場合は X 全て範囲を演算対象とします。 k の要素数が X の要素数より少ない場合 即値、または要素 1 個の参照チャネルで記述した場合 k を検索開始 Index と見なし当該 Index から X で記述するデータ列の最終データ迄を演算範囲とし、その最小値位置(Index)を戻します。 複数要素の参照チャネルで記述した場合 要素で示す Index 範囲毎を演算範囲とします。k の要素は昇並びの必要があります。又、k で指定した Index が存在しない場合は実行時 Error となります。 例えば、k が 100,200,300 の値の場合、100~199、200~299、300~最終データ番号と 3 区間の最小値位置を戻します。 k の要素数が X の要素数と等しい場合 k で記述された検索範囲は、Index 指定ではなく"0"と"1"からなる論理値数列と見なし、その論理"1"の範囲での最小値位置(Index)を戻します。</p> <p>[X]<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>
記述例	MNP(#1)
備考	<p>演算対象データの最小値位置のデータ番号を戻します。結果のデータ個数は 1 となります。</p> 

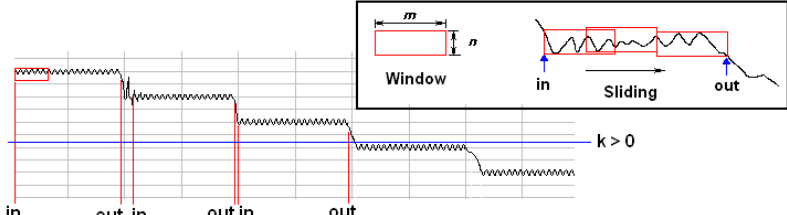
2. 14. 12. 最大値位置 Index 検索関数

機能	X の最大値位置のデータ番号を戻します。
文法	MXP(k,X)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式> 検索対象範囲を記述します。検索対象範囲は記述した k の要素数により意味が異なります。又、省略記述した場合は X 全て範囲を演算対象とします。</p> <p>k の要素数が X の要素数より少ない場合 即値、または要素 1 個の参照チャネルで記述した場合 k を検索開始 Index と見なし当該 Index から X で記述するデータ列の最終データ迄を演算範囲とし、その最小値位置(Index)を戻します。</p> <p>複数要素の参照チャネルで記述した場合 要素で示す Index 範囲毎を演算範囲とします。k の要素は昇並びの必要があります。又、k で指定した Index が存在しない場合は実行時 Error となります。 例えば、k が 100,200,300 の値の場合、100~199、200~299、300~最終データ番号と 3 区間の最小値位置を戻します。</p> <p>k の要素数が X の要素数と等しい場合 k で記述された検索範囲は、Index 指定ではなく"0"と"1"からなる論理値数列と見なし、その論理"1"の範囲での最小値位置(Index)を戻します。</p> <p>【X】<必須><収録チャネル/数値属性参照チャネル/演算式> 演算対象数列を記述します。</p>
記述例	MXP(#1)
備考	<p>k を記述省略した場合: 演算対象データの最大値位置 Index を戻します。結果のデータ個数は 1 となります。</p>  <p>k を論理パルス列で記述した場合: 論理"1"区間毎に最大値位置 Index を戻します。結果のデータ個数は、パルス列の"1"区間数に j 従属します。</p> 

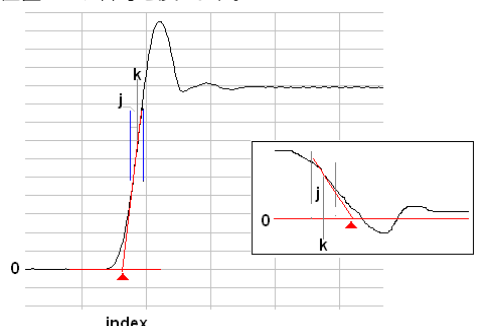
2. 14. 13. 区間最大値/最小値 Index 検索関数

機能	区間検出用閾値を越えてから再び逆傾斜で越えるまでを区間とし、その最大値位置または最小値位置を戻します。なお、最大値位置または最小値位置の何れを検索するかは引数で与えます。																		
文法	PMP(j,k,m,X)																		
引数	<p>【j】<必須><即値/数値属性参照チャネル> 最小値位置、または最大値位置の種別フラグを記述します。記述する種別フラグを下表に示します。</p> <table border="1"> <thead> <tr> <th>j 値</th> <th>種別</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>最小値位置データ番号</td> </tr> <tr> <td>1</td> <td>最大値位置データ番号</td> </tr> </tbody> </table> <p>【k】<必須><即値/数値属性参照チャネル/演算式> データ番号検索区間抽出閾値を記述します。 データ番号検索区間は、区間抽出閾値(k)<=有効閾値(m)の場合は区間抽出閾値を上昇で過ぎてから下降で過るまで、区間抽出閾値(k)>有効閾値(m)の場合は区間抽出閾値(k)を下降で過ぎてから上昇で過るまでを検索対象区間と見なします。なお、検索対象区間に含まれるデータ個数が 5 個以上存在しない場合は検索区間と見なしません。</p> <p>【m】<必須><即値/数値属性参照チャネル/演算式> 有効閾値を記述します。有効閾値は検索対象区間の最大値/最小値が有効か否か判定する閾値です。</p> <table border="1"> <thead> <tr> <th>j</th> <th>種別</th> <th>K と m の関係</th> <th>有効</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>最小値位置データ番号</td> <td>$k > m$</td> <td>最小値 $< m$</td> </tr> <tr> <td>1</td> <td>最大値位置データ番号</td> <td>$k < m$</td> <td>最大値 $> m$</td> </tr> </tbody> </table> <p>【X】<必須><収録チャネル/参照チャネル/演算式> 検索対象数列を記述します。</p>	j 値	種別	0	最小値位置データ番号	1	最大値位置データ番号	j	種別	K と m の関係	有効	0	最小値位置データ番号	$k > m$	最小値 $< m$	1	最大値位置データ番号	$k < m$	最大値 $> m$
j 値	種別																		
0	最小値位置データ番号																		
1	最大値位置データ番号																		
j	種別	K と m の関係	有効																
0	最小値位置データ番号	$k > m$	最小値 $< m$																
1	最大値位置データ番号	$k < m$	最大値 $> m$																
記述例	<p>\$20=PMP(1,50,100,#1) \$21=PTV(\$20,#1) ⇒ 求めたデータ番号からデータ値に変換しています。</p>																		
備考	<p>$k < m$ の場合 k を上昇で過ぎてから下降で過るまでを区間と見なし 最大値指定の場合 ⇒ 当該区間内で m を上昇で過ぎた最大値 最小値指定の場合 ⇒ 当該区間内で m を下降で過ぎた最小値</p> <p>$k > m$ の場合 k を下降で過ぎてから上昇で過るまでを区間と見なし 最大値指定の場合 ⇒ 当該区間内で m を上昇で過ぎた最大値 最小値指定の場合 ⇒ 当該区間なおで m を下降で過ぎた最小値 なお、存在しない場合は-1 が戻ります。</p> <p>引数区抽出閾値(k)と有効閾値(m)が数列で指定された場合 要素数が等しい場合、検索は区間抽出閾値(k)の要素数分行われ、存在しない場合は対応した戻り値要素に-1 を戻し、次回の区間検索開始 Index は更新されません。存在した場合は区間検索開始 Index が自動更新されます。 区間検索は複数存在しても、Index が求まる最初の区間のみとなります。 要素数が等しく無い場合何れも参照される Index は 0 のみとなり、従来通り、戻り値個数は、存在した個数と一致しません。</p> <table border="1"> <thead> <tr> <th></th> <th>j=1 MAX</th> <th>j=0 MIN</th> </tr> </thead> <tbody> <tr> <th>k ≤ m</th> <td></td> <td></td> </tr> <tr> <th>k > m</th> <td></td> <td></td> </tr> </tbody> </table>		j=1 MAX	j=0 MIN	k ≤ m			k > m											
	j=1 MAX	j=0 MIN																	
k ≤ m																			
k > m																			

2. 14. 14. 一定値区間 Index 検索関数

機能	波形上に設定した窓をスライディングさせ、窓内に入ったデータ番号と外れたデータ番号を戻します。
文法	SWP(m,n,k,X)
引数	<p>[m]<必須><即値/数値属性参照チャンネル/演算式> 検索窓の横方向幅を単位 sec で記述します。複数要素を持つ数列で記述しても参照される値は先頭のみとなります。</p> <p>[n]<必須><即値/数値属性参照チャンネル/演算式> 検索窓の縦方向幅を物理量値で絶対値で記述します。複数要素を持つ数列で記述しても参照される値は先頭のみとなります。検索窓は、矩形でその大きさは、m と k で設定されます。</p> <p>[k]<必須><即値/数値属性参照チャンネル/演算式> 検索窓のスライディング動作閾値を記述します。検索窓は、ここで設定した動作閾値>=0 の場合はデータが閾値を越えている領域を検索し、動作閾値<0 の場合はデータが閾値を下回っている領域を検索します。</p> <p>[X]<必須><収録チャンネル/数値属性参照チャンネル/演算式> 検索対象数列を記述します。</p>
記述例	SWP(\$1,\$2,\$3,#1)
備考	<p>スライディング動作閾値の要素数が 1 個の場合、検索対象データを時間軸方向に検索窓が移動し、検索窓横方向で設定した時間幅内の全データの(最大値-最小値)が検索窓縦方向で設定値以内に入っている状態を成立とし、不成立⇒成立の時の窓左端のデータ番号を IN、同様に成立⇒不成立の時の窓右端のデータ番号を OUT として戻します。戻りデータ個数は成立箇所数×2 となります。</p>  <p>検索対象データの先頭から成立した場合は、先頭データ番号を最初の IN とします。検索結果の戻り値並びは、先頭データは IN/OUT 個数、以下 IN→OUT→IN→OUT→IN→…の順にデータ番号を戻します。結果の戻りデータ個数は成立区間の個数×2+1 となります。結果の戻り先頭データが-1 の場合は成立区間が存在しなかったことを意味します。スライディング動作閾値の要素が複数個の場合、検索は、要素数分繰り返して行われ、検索開始から最初に成立した窓 IN データ番号と外れたデータ番号のペアで検索回数分+1 が戻ります。先頭データは IN/OUT 個数となります。なお、検索開始データ番号は、直前の検索で成立した場合には次の検索開始データ番号は、直前の検索の窓 OUT データ番号から行われます。</p>

2. 14. 15. ゼロ交点 Index 検索関数

機能	指定された検索範囲の最大傾斜を延長してゼロとの交点データ番号を戻します。
文法	ZIS(k,j,X)
引数	<p>[k]<必須><即値/数値属性参照チャンネル/演算式> 交点検索中心データ番号を記述します。</p> <p>[j]<必須><即値/数値属性参照チャンネル/演算式> 交点検索中心データ番号からの土幅を単位 sec で記述します。</p> <p>[X]<必須><収録チャンネル/数値属性参照チャンネル/演算式> 検索対象数列を記述します。</p>
記述例	ZIS(\$1,\$2,#1)
備考	<p>検索対象データの k で指定されたデータ番号位置の j で指定した土幅を検索対象範囲とし、検索対象範囲の最大傾斜を延長してゼロとの交点位置データ番号を戻します。</p> 

2. 14. 16. 存在 Index 取得関数

機能	k で与えられた検索条件を満たす検索数列の要素値が検索対象数列に含まれる場合その全ての Index 列を戻します。														
文法	CMP(k,X,Y)														
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式> 検索条件コードを記述します。記述する検索コード表を下記に示します。</p> <table border="1"> <thead> <tr> <th>検索条件コード</th> <th>戻り値</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X=Y の Index を戻す</td> </tr> <tr> <td>1</td> <td>X<>Y の Index を戻す</td> </tr> <tr> <td>2</td> <td>X<=Y の Index を戻す</td> </tr> <tr> <td>3</td> <td>X<Y の Index を戻す</td> </tr> <tr> <td>4</td> <td>X>=Y の Index を戻す</td> </tr> <tr> <td>5</td> <td>X>Y の Index を戻す</td> </tr> </tbody> </table> <p>記述省略した場合は 0 と見なす。又、コード表以外の記述は実行時 Error となります。</p> <p>【X】<必須><即値/数値属性参照チャネル/演算式> 検索数列を記述します。</p> <p>【Y】<必須><即値/数値属性参照チャネル/演算式> 検索対象数列を記述します。</p>	検索条件コード	戻り値	0	X=Y の Index を戻す	1	X<>Y の Index を戻す	2	X<=Y の Index を戻す	3	X<Y の Index を戻す	4	X>=Y の Index を戻す	5	X>Y の Index を戻す
検索条件コード	戻り値														
0	X=Y の Index を戻す														
1	X<>Y の Index を戻す														
2	X<=Y の Index を戻す														
3	X<Y の Index を戻す														
4	X>=Y の Index を戻す														
5	X>Y の Index を戻す														
記述例	CMP(\$1,\$2)														
備考	<p>比較演算との違いは、比較演算が対応する要素毎の比較に対して、X の要素毎に X の全ての要素と比較し成立した Y の Index を戻す点異なります。従って、比較演算で何れか片方の要素数が 1 個の時に近似した機能となります。</p> <p>\$1 ← 1,2,3,4 \$2 ← 0,1,3,2,2,5,4,1 \$3 = CMP(\$1,\$2) \$3 → 1,7,3,4,2</p> <p>X の要素が Y 要素と一致した Index を求める場合は、EQP(0,\$1,\$2)と近似しますが、複数一致した場合に最初に一致した Index を戻す機能に対して、CMP(X,Y)とした場合、一致した全ての Index を戻します。つまり EQP が戻り値の要素数は X の要素数に一致しますが、本関数は無関係となります。</p> <p>なお、一致した個数を X の要素毎に求める場合は、EQP(1,X,Y)として求めます。</p>														

2. 14. 17. PEAK 検出関数

機能	波形上の Peak を検出し Index を戻します。
文法	PDT(k,m,X,f)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 検索窓の深さを $k > 0$ で記述します。</p> <p>【m】<必須><即値/数値属性参照チャネル/演算式> 検索窓の幅をデータ個数奇数で記述します。偶数で記述された場合+1 されて参照されます。</p> <p>【X】<必須><数値属性チャネル> 検索対象波形数列为記述します。</p> <p>【f】<省略可><即値/数値属性参照チャネル/演算式> 内部で参照される検索窓の深さを k で指定した深さをそのまま使用するか、検索対象波形数列の値に設定した深さを掛算して参照するかのフラグを意味します。 0 又は記述省略した時 k で設定された値をそのまま使用します。 0 以外を記述した時 k で設定した値に検索対象数列の値を掛算して($k' = X(i) * k$) 修飾した値を深さとし、深さは浮動する事になります。但し、検索対象数列の値が 0 の場合は、0 と記述した時と同じで深さ k をそのまま使用します。</p>
記述例	PDT(1.25,11,#1)
備考	<p>検索対象数列の先頭から検索窓が入る地点から検索を開始し、検索対象数列の終端から検索窓が入る地点までをスライディングして検索します。従って、検索対象数列の要素数を n とし、検索窓の幅を m とした場合、検索対象開始 Index は $\text{INT}(m/2)$、終了 Index は $n - \text{INT}(m/2)$ となります。</p> <p>成立条件は、検索窓底面から検索窓に入り、検索窓底面から出るとなります。言い換えると、検索窓内最大値 X_{\max} とすると、底面閾値は $X_{\max} - \text{深さ}$ となり、検索窓内に全データは底面閾値 $\leq X(i) \leq X_{\max}$ の範囲に入っている事が成立条件です。</p> <p>なお、成立した場合、次の検索開始位置は直前の検索窓から出た Index に飛びます。</p> <p>戻り値は、成立検索窓の最大値位置 ($n1$)、検索窓底面閾値を上方で過った地点 (底面閾値 $\leq X(n2)$ の $n2$) と検索窓底面閾値を下方で過った地点 (底面閾値 $= X(n3)$ の $n3$) の 3 点をペアで成立箇所分戻ります。</p> <p>下図に示す何れのパターンでも成立となります。</p> <p>条件が不成立の場合戻り値は 1 個で -1 が戻ります。</p>

2. 15. 文字列取扱関数

2. 15. 1. 文字列範囲指定削除関数

機能	指定した文字位置から文字数分を削除します。char delete 文と等価な機能です。
文法	CDEL(k,n,&m)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> 削除開始文字位置を記述します。負数で記述した場合は終端からの文字位置を意味します。記述省略した場合は、1 と等価となります。なお、対象文字列の要素数と等しい配列で記述した場合は、要素毎に開始文字位置を指定できます。</p> <p>【n】<必須><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> 削除文字数を記述します。対象文字列&n の要素が指定された文字数に満たない場合は、開始文字位置から先頭或いは終端まで削除されます。対象文字列と要素数が等しい配列で記述した場合は、要素毎に削除文字数を指定できます。</p> <p>【&m】<必須><文字属性参照チャネル/戻り文字属性演算式><間接指定可> 対象文字列を記述します。</p>
記述例	&DEL(\$1,\$2,&1)
備考	戻り属性は文字属性となります。

2. 15. 2. 文字列切り出し関数

機能	文字列から指定した範囲の文字列を切り出します。char extract 文と等価な機能です。
文法	CEXT(k,n,&m)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> 切り出し開始文字位置を半角換算で記述します。なお、負数で記述した場合は、文字列の終端からの文字位置となります。</p> <p>【n】<必須><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> 切り出す文字数を半角換算で記述します。開始位置から切り出す文字数に不足した場合は、終端或いは先頭文字までが切り出されます。</p> <p>【&m】<必須><文字列即値/文字属性参照チャネル/戻り文字属性演算式><Index 指定可/間接指定可> 切り出し対象文字列を記述します。</p>
記述例	CEXT(&1,5,8)
備考	戻り属性は文字属性です。 切り出し対象文字列を配列で記述した場合は、戻り要素数は配列の要素数に等しい配列で戻ります。

2. 15. 3. 文字列検索関数

機能	検査対象文字列に指定した文字列を含んでいるかを検査し、存在した場合は先頭文字位置を戻します。 char find 文と等価な機能です。
文法	CFND(k,&n,&m)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> 検査開始文字位置を半角換算で記述します。負数で記述した場合は、検査対象文字列の終端からの文字位置となります。</p> <p>【&n】<必須><文字列即値/文字属性参照チャネル/戻り文字属性演算式><Index 指定可/間接指定可> ふくまれているかの検査文字列を記述します。</p> <p>【&m】<必須><文字属性参照チャネル/戻り文字属性演算式><Index 指定可/間接指定可> 検査対象文字列を記述します。</p>
記述例	CFND(1,"abc",&1)
備考	戻り属性は数値属性です。

2. 15. 4. 文字列挿入関数

機能	対象文字列に指定した文字列を挿入します。 char insert 文と等価な機能です。
文法	CINS(k,&n,&m)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> 挿入開始文字位置を記述します。負数で記述した場合は終端からの文字位置を意味します。記述省略した場合は、1 と等価となります。なお、対象文字列の要素数と等しい配列で記述した場合は、要素毎に開始文字位置を指定できます。</p> <p>【&n】<必須><文字列即値/文字属性参照チャネル/戻り文字属性演算式><Index 指定可/間接指定可> 挿入する文字列を記述します。対象文字列と要素数が等しい配列で記述した場合は、要素毎に削除文字数を指定できます。開始文字位置が対象文字列の長さを越えた場合は、終端或いは先頭に挿入されます。</p> <p>【&m】<必須><文字属性参照チャネル/戻り文字属性演算式><間接指定可> 対象文字列を記述します。</p>
記述例	CINS(\$1,&1,&2)
備考	戻り属性は文字属性となります。

2. 15. 5. 文字列配列再構成関数

機能	文字列配列を指定したフラグが 0 の時は同じく指定した再構成論理は"1"の要素のみで再構成し、フラグ 1 の時は再構成論理の値を Index と見なし、Index 順に並び替えます。 char recomposition 文と等価な機能です。
文法	CREC(k,n,&m)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式> フラグを意味し、0 または 1 と記述します。0 の時は m の値の論理"1"のみの対応する Index の要素のみを切り出します。1 の時は n の値を Index と見なし、Index 順に並び替えます。記述省略した場合は、0 と見なしします。</p> <p>【n】<必須><数値属性参照チャネル/演算式> k が 0 の時は要素の値は 0 または 1 で構成された配列で記述します。k が 1 の時は、要素の値は Index で記述し、0 から対象文字列の要素数-1 までの値で唯一無二の必要があります。なお、要素数は、対象文字列&nの要素数と等しい必要があります。</p> <p>【&m】<必須><文字属性参照チャネル/戻り文字属性演算式> 対象文字列を記述します。</p>
記述例	CREC(0,\$1,&2)
備考	戻り属性は数値属性です。

2. 15. 6. 文字列置換関数

機能	対象文字列から指定した検査文字列と一致した文字列を指定した置換文字列に置き替えます。 char replace 文と等価な機能です。
文法	CREP(k,&s,&n,&m)
引数	<p>【引数】<省略可><即値/数値属性参照チャネル/演算式> 検査開始文字位置を記述します。負数で記述した場合は終端からの文字位置を意味します。記述省略した場合は 1 と等価となります。なお、対象文字列と要素数と等しい配列で記述した場合は、要素毎に開始文字位置を指定できます。また、検索対象文字列を記述省略した場合は意味が異なり、置換開始文字位置を意味します。</p> <p>【&s】<省略可><文字列即値/文字属性参照チャネル> 対象文字列から置換する検査文字列を記述します。なお、対象文字列と要素数と等しい配列で記述した場合は、要素毎に検査文字列を指定できます。記述省略した場合、k で示す文字位置から&n で記述する文字列長さ分が置換されます。</p> <p>【&n】<必須><文字列即値/文字属性参照チャネル> 置き換える文字列を記述します。対象文字列と要素数と等しい配列で記述した場合は、要素毎に置き換える文字列を指定できます。</p> <p>【&m】<必須><文字属性参照チャネル> 対象文字列を記述します。</p>
記述例	CREP(1,"abc","def",&1)
備考	<p>戻り属性は文字属性となります。</p> <p>assign &1 = "012345678901234567890"</p> <p>&2 = CREP(1,"012","abcd",&1) ⇒ &2 には"abcd3456789abcd34567890"が格納されます。</p> <p>&2 = CREP(1,"abcd",&1) ⇒ &2 には"abcd45678901234567890"が格納されます。</p>

2. 15. 7. 文字列配列文字数取得関数

機能	文字数を半角換算で求めます。 char length 文と等価な機能です。
文法	CLN(&m)
引数	【&m】<必須><文字列即値/文字属性参照チャネル/文字属性戻り演算式><Index 指定可/間接指定可> 文字列の長さ(要素数)を求めたい文字列を記述します。
記述例	CLN(&1)
備考	戻りは数値属性です。与えた文字列が複数要素を持つ場合は、戻り値も複数となります。 文字列の要素数を求める場合は、LEN(&CLN(&1))と記述することで求められます。

2. 15. 8. 文字列要素数取得関数

機能	文字属性参照チャネルの要素数を求めます。 char num_element 文と等価な機能です。
文法	ELM(&m)
引数	【&m】<必須><文字属性参照チャネル/文字属性戻り演算式><間接指定可> 文字列配列の要素を求めたい文字属性参照チャネルを記述します。
記述例	ELM(&1)
備考	戻り値は数値属性となります。 数値属性参照チャネルの数列長さを求める LEN(X)と等価な機能です。

2. 15. 9. 文字列数値変換関数

機能	文字列から数値に変換します。
文法	CTN(&m)
引数	【&m】<必須><文字列即値/文字属性参照チャネル/戻り文字属性演算式> 数値変換対象文字列を記述します。変換は対象文字列の先頭から数値文字記述範囲までが対象となります。
記述例	CTN(&1)
備考	戻り属性は数値属性となります。 対象文字列が数値変換不可の場合は実行時 Error となります。

2. 15. 10. 文字列大文字小文字変換関数

機能	文字列を半角英数字と見なし、大文字を小文字に変換、または小文字を大文字に変換します。
文法	CBSC(&m,k)
引数	【&m】<必須><文字列即値/文字属性参照チャネル/戻り文字属性演算関数> 変換対象文字列を記述します。文字列に全角文字が含まれていた場合、全角文字は変換されません。 なお、複数要素で記述すると戻り要素も複数となります。 【k】<省略可><即値/数値属性参照チャネル/演算式> 変換フラグを記述します。変換フラグは 0 または 1 で、0 の時は変換対象文字列に含まれる大文字を小文字に変換、1 の時は小文字を大文字に変換します。記述省略した場合及び 0 または 1 で無い時は 0 と見なします。なお、複数要素で記述しても参照される Index は 0 のみとなります。
記述例	CBSC("Abc",0)
備考	&1 = CBSC("Abc",0) &1 → "abc" &1 = CBSC("Abc",1) &1 → "ABC"

2. 15. 11. 文字列要素置換関数

機能	与えられた Index の文字列を指定した文字列に置換します。
文法	CERP(k,&n,&m)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> 置換する Index を記述します。ここで記述された Index が置換対象文字列に存在した場合に置換します。 なお、Index 並びは順不同で良い、また、Index は唯一無二の必要はありませんが、同じ Index が存在した時は最後に置換された内容が戻ります。 【&n】<必須><文字列即値/文字属性参照チャネル/戻り文字属性演算式> 一致した Index が置換対象文字列に存在した場合に置き換える文字列を記述します。複数要素を持つ配列で記述した場合、置換順に参照され置換数に不足した場合は循環して参照されます。 【&m】<必須><文字属性参照チャネル/戻り文字属性演算式> 対象文字列を記述します。
記述例	CERP(\$1,"",&2)
備考	

2. 15. 12. 文字列比較関数

機能	文字列同士を比較して結果論理値を返す。比較は要素毎に行い、k=0 の時、一致の場合0、不一致の場合 0 を返す。戻り個数は比較対象文字列&n の要素数に一致します。k=1 の時、一致した&n の Index を返します。
文法	CEQ(&m,&n,k)
引数	<p>【&m】<必須><文字列即値/文字属性参照チャネル/戻り文字属性演算関数> 比較文字列を記述します。</p> <p>【&n】<必須><文字属性参照チャネル/戻り文字属性演算式> 比較対象文字列を記述します。比較は Index 毎に行います。 なお、複数要素で記述すると戻り要素も複数となります。</p> <p>【k】<省略可><即値/数値属性参照チャネル/演算式> 論理値を返すか Index を返すかのフラグを記述します。k=0 の時は論理値を返します。 戻り論理値指定の場合で\$m の要素数が不足した場合、不足分は不一致 0 となります。&m 要素数が多い場合、余った要素は比較されません k=1 の時は&n の Index を返します。Index 指定の場合で全て一致しない場合は-1 を返します。なお、記述省略された場合は 0 と見なします。</p>
記述例	CEQ(&1,&2)
備考	<pre>& ← "abc","cde","efg","ghi" &2 ← "cde","cde","efg" \$1 = CEQ(&2,&1) \$1 → 0,1,1,0</pre>

2. 15. 13. 文字列指定文字削除関数

機能	対象文字列から削除指定した文字列を検索開始文字位置から対象文字列から削除します。
文法	CDEC(k,n,&n,&m)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式> 検索開始文字位置を記述します。正数で指定した場合は先頭からの検索開始文字位置、負数で記述した後方から検索開始文字位置を意味します。何れも、開始文字位置が不正な場合は、何もしません。(削除文字列が当該文字列に存在していない場合と等価)又、複数要素を持つ数列で指定された場合は、循環して参照されます。 記述省略された場合は、1 と見なします。</p> <p>【n】<必須><即値/数値属性参照チャネル/演算式> 削除演算終了フラグを意味し、0 または 1 と記述します。0 または 1 以外は実行時 Error となります。 0 と記述した場合は、検索開始文字位置から検索対象文字列を検索し、一致しない文字列に出会うとそこまでの文字列を削除して終了します。1 と記述した場合は、同じく検索開始文字位置から文字列全てを検索し、一致した全ての文字列を削除します。</p> <p>【&n】<必須> 削除文字列を記述します。複数要素を持つ配列で指定された場合は、循環して参照されます。 削除対象文字列に複数箇所、一致した場合は、複数箇所削除されます。なお、削除結果で全ての文字列が削除された場合は、半角スペース 1 文字に置き換わります。(null にはなりません)</p> <p>【&m】<必須><文字属性参照チャネル/戻り文字属性演算式> 対象文字列を記述します。</p>
記述例	CDEC(0,"",&1) ⇒ 文字列先頭から半角スペース部分を削除する
備考	<pre>戻り属性は文字属性となります。 & ← " abc ", " cde ", " efg ", " ghi" &2 = CDEC(1,"",&1) &2 → "abc","cde","efg","ghi" &3 = CDEC(0,"",&1) &3 → "abc ", "cde ", "efg ", "ghi"</pre>

2. 15. 14. 文字列 Index 指定抽出関数

機能	Index を指定して、Index 位置の文字列を戻します。指定した Index が存在しない場合、実行時 ERROR となります。
文法	CPTV(k,&m)
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> 抽出する Index(データ番号)を記述します。記述する Index は昇順並びで対象文字列配列に全て存在している必要があります。 戻り個数は k の要素数に依存します。</p> <p>【&m】<必須><文字属性参照チャネル/戻り文字属性演算式> 対象文字列を記述します。</p>
記述例	CPTV(3,&1)
備考	<p>戻り属性は文字属性となります。</p> <p>\$1 ← 1,3 &1 ← "abc","cde","efg","ghi" &2 = CPTV(\$1,&1) &2 → "cde","ghi"</p>

2. 15. 15. 文字列分離関数

機能	与えた文字列を指定した区切り文字で複数の文字列に分離し戻します										
文法	CSEP((k,&m)										
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式> セパレータ(区切り文字)をコードで記述します。</p> <table border="1"> <thead> <tr> <th>分離コード</th> <th>セパレータ</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>改行</td> </tr> <tr> <td>1</td> <td>半角カンマ</td> </tr> <tr> <td>2</td> <td>tab</td> </tr> <tr> <td>3</td> <td>半角スペース</td> </tr> </tbody> </table> <p>記述省略した場合は 0 と見なします。 複数要素を持つ数列で記述しても参照される Index は 0 のみとなります。 セパレータに半角スペースが選択された場合で半角スペースが連続した場合は、1 個のセパレータと見なし、文字列から半角スペースは無くなります。又、文字列の先頭或いは末尾のスペースはセパレータと見なせず、半角スペースのみ削除されます。</p> <p>【&m】<必須><文字属性参照チャネル/戻り文字属性演算式> 分離対象文字列を記述します。複数要素を持つ配列で指定されても参照される Index は 0 のみとなります。</p>	分離コード	セパレータ	0	改行	1	半角カンマ	2	tab	3	半角スペース
分離コード	セパレータ										
0	改行										
1	半角カンマ										
2	tab										
3	半角スペース										
記述例	CSEP(&1) /*chr 属性で読み出したテキストを行分解する*/										
備考	<p>戻り属性は文字属性となります。</p> <p>分離した文字列にセパレータ文字を含みません。戻り要素数は、分離対象文字列構成に従属します。</p> <p>セパレータに半角スペースを指定した場合</p> <p>&1 = " AB CD "</p> <p>&2 = CSEP(3,&1)</p> <p>&2(0) ⇒ "AB"</p> <p>&2(1) ⇒ "C"</p> <p>&2(2) ⇒ "D"</p>										

2. 15. 16. 文字列結合関数

機能	要素ごとに格納された文字列を結合し一つの文字列にします。												
文法	CNCT(k,&m)												
引数	<p>【k】<必須><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> 結合に挿入するセパレータ(区切り文字)をコードで記述します。</p> <table border="1"> <thead> <tr> <th>区切りコード</th> <th>区切り文字</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>改行</td> </tr> <tr> <td>1</td> <td>半角カンマ</td> </tr> <tr> <td>2</td> <td>tab</td> </tr> <tr> <td>3</td> <td>半角スペース</td> </tr> <tr> <td>4</td> <td>無し</td> </tr> </tbody> </table> <p>記述省略した場合は 0 と見なします。 複数要素を持つ数列で記述しても参照される Index は 0 のみとなります。</p> <p>【&m】<必須><文字属性参照チャネル/戻り文字属性演算式><Index 指定可/間接指定可> 結合対象文字列を記述します。</p>	区切りコード	区切り文字	0	改行	1	半角カンマ	2	tab	3	半角スペース	4	無し
区切りコード	区切り文字												
0	改行												
1	半角カンマ												
2	tab												
3	半角スペース												
4	無し												
記述例	CNCT(3,&1)												
備考	<p>戻り属性は文字属性となります。戻り要素数は 1 個となります。</p> <p>&1(0)←"abc" &1(1)←"cde" &1(2)←"fgh" &2 = CNCT(1,&1) &2→ "abc,cde,fgh"</p>												

2. 15. 17. 現在年月日取得関数

機能	現在年月日に戻します
文法	CNDT()
引数	なし
記述例	CNDT()
備考	戻り属性は文字属性となり、フォーマットは MM-DD-YYYY となります。

2. 15. 18. 現在時刻取得関数

機能	現在時刻に戻します。
文法	CNTM()
引数	なし
記述例	CNTM()
備考	戻り属性は文字属性となり、フォーマットは hh:mm:ss となります。

2. 15. 19. 数値ASCII文字列変換関数

機能	与えられた数列の要素毎1文字と見なし ASCII 文字に変換します。
文法	CVTA(k,X,f)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> 変換コード表コードを意味します。0: Ascii コード ※ 現在は ASCII コード以外サポートしていません。記述する場合は必ず 0 とします。</p> <p>【X】<必須><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> 変換対象数列を記述します。k=0 又は記述省略された場合 X の値は 32~126 範囲の必要があります。</p> <p>【f】<省略可><即値/数値属性参照チャネル/演算式><Index 指定可/間接指定可> X の要素毎に変換した文字を連結して文字属性参照チャネルに戻すか、1 文字毎に文字属性参照チャネルの要素に戻すかを意味します。0 の時:連結する。0 以外の時:連結しないを意味します。 記述省略した場合は 0 と見なします。但し、記述する場合は先行従って、戻り要素数はこのフラグに従属します。</p>
記述例	CVTA(\$1)
備考	バイナリファイル読み出しで文字列を byte 属性で読み出し、読み出し結果数値を文字列に変換する時などに使用します。

2. 16. 20. 文字列繰り返し生成関数

機能	与えられた文字を指定された繰り返し数分接続した文字列を戻します。
文法	CSTR(k,&m)
引数	<p>[k]<省略可><即値/数値属性参照チャンネル/演算式><Index 指定可/間接指定可> 繰り返し数を記述します。記述は正整数で>0 で無ければなりません。<=0 の場合は実行時 Error となります。 記述省略した場合は 1 と見なします。</p> <p>[&m]<必須><文字列即値/文字属性参照チャンネル/戻り文字属性演算式> 繰り返し文字列を記述します。</p>
記述例	CSTR(3,"abcd") 戻りは abcdabcdabcd となります。
備考	&m が複数要素を持つ配列で記述された場合は配列の要素毎に複数繰り返されて生成されます。

2. 16. 21. 文字列接続関数

機能	文字列を要素毎に連結します。
文法	CCON(&n,&m,&k,...)
引数	<p>[&n]<必須><文字列即値/文字属性参照チャンネル/演算式><Index 指定可/間接指定可> 連結する文字列を記述します。文字列属性参照チャンネルで記述した場合、内容は null でも問題ありません、</p> <p>[&m]<必須><文字列即値/文字属性参照チャンネル/演算式><Index 指定可/間接指定可> 連結する文字列を記述します。最大 10 個まで記述できます。</p>
記述例	CCON("abc","def") → 戻り文字列は"abcdef"となります。
備考	戻り属性は文字属性です。戻り文字列要素数は記述された接続文字列要素の最も多い配列の個数と等しく、短い配列は循環して参照されます。

2. 16. 22. 文字列配列連結関数

機能	文字列を要素毎に連結します。
文法	CLNK(&n,&m,&k,...)
引数	<p>[&n]<必須><文字列即値/文字属性参照チャンネル/演算式><Index 指定可/間接指定可> 連結する文字列を記述します。文字列属性参照チャンネルで記述した場合、内容は null でも問題ありません、</p> <p>[&m]<必須><文字列即値/文字属性参照チャンネル/演算式><Index 指定可/間接指定可> 連結する文字列を記述します。最大 10 個まで記述できます。</p>
記述例	&1 = CLNK("abc","def") → 戻り文字列配列の要素数は 2 個、&1(0):="abc" &1(1):="def"となります。
備考	戻り属性は文字属性です。戻り文字列要素数は記述された連結文字列配列の合計要素数となります。

2. 16. 23. 数値文字列変換関数

機能	数値を指定した型式で文字列に変換します。								
文法	CNSC(k,m,n,X)								
引数	<p>[k]<省略可><即値/数値属性参照チャンネル/演算式><Index 指定可/間接指定可> 変換型式コードを意味します。コードは 0,1,2 の何れかを記述します。記述省略された場合は 0 と見なします。</p> <table border="1" data-bbox="284 1370 691 1491"> <thead> <tr> <th>コード</th> <th>変換形式</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>F 形式</td> </tr> <tr> <td>1</td> <td>指数形式指数部 2 桁</td> </tr> <tr> <td>2</td> <td>指数形式指数部 3 桁</td> </tr> </tbody> </table> <p>[m]<必須><即値/数値属性参照チャンネル/演算式><index 指定可/間接指定可> 変換結果の全文字数を記述します。数値文字列変換結果が記述された全文字数に満たない場合は、全文字数を満たす様に先行半角スペースを付加します。全文字数に溢れた場合は、実行時エラーとなります。 但し、0 と記述された場合は、特別な意味を持ち、数値文字列変換結果を戻し、満たない場合の先行半角スペースの付加を行いません。</p> <p>[n]<必須><即値/数値属性参照チャンネル/演算式><Index 指定可/間接指定可> 変換結果の小数点以下桁数を記述します。数値文字列変換に際して小数点以下桁数の最終桁+1 を四捨五入します。なお、k で指数形式が指定された場合で 0 と記載した場合は小数点以下桁を 3 桁と見なします。</p> <p>[X]<必須><文字列即値/文字属性参照チャンネル/演算式><Index 指定可/間接指定可> 文字列変換する数値を記述します。</p>	コード	変換形式	0	F 形式	1	指数形式指数部 2 桁	2	指数形式指数部 3 桁
コード	変換形式								
0	F 形式								
1	指数形式指数部 2 桁								
2	指数形式指数部 3 桁								
記述例	<p>CNSC(0.0,0,-1234.567) 戻り文字列は"-1235"となります。</p> <p>CNSC(0,6,0,-1234.567) 戻り文字列は"-1235"となります。</p> <p>CNSC(1,12,4,-1234.567) 戻り文字列は"-1.2346E+03"となります。</p>								
備考	戻り属性は文字属性です。戻り文字列要素数は記述された数値の要素数と等しい。								

2. 16. 24. 文字列文字数整合関数

機能	文字列を指定した文字数に整合します。
----	--------------------

文法	CADJ(k,&m)
引数	<p>【k】<必須><即値/数値属性属性参照チャネル/演算式><Index 指定可/間接指定可> 整合する文字数を記述します。記述する文字数は 0 以上の必要があります。少数点以下を含む場合は小数点以下四捨五入して参照されます。</p> <p>【&m】<必須><文字列即値/文字属性参照チャネル/演算式><index 指定可/間接指定可> 整合する文字列を記述します。k で記述した文字列に満たない場合は後端に半角スペースを付加して整合します。溢れた場合は文字列の先頭から k で記述した文字数分が切り取られます。</p>
記述例	<p>CADJ(10,"abc1234") 戻り文字列は"abc1234" となります。</p> <p>CNSC(3.3,"abc1234") 戻り文字列は"abc"となります。</p>
備考	戻り属性は文字属性です。戻り文字列要素数は記述された文字列配列の要素数と等しい。

2. 16. 解析情報取得関数

2. 16. 1. 収録ファイル名取得関数

機能	現在解析対象となっているファイル名を取得します。 read file_name 文と等価な機能です。
文法	CFNM()
引数	引数なし
記述例	CFNM()
備考	戻り属性は文字属性となります。 解析対象ファイルが読み出されていない場合は実行時 ERROR となります。

2. 16. 2. 収録チャンネル信号名取得関数

機能	解析対象ファイルのチャンネルに付けられている信号名を取得します。 read ch_name 文と等価な機能です。
文法	CHNM(k)
引数	【k】<必須><即値/数値属性参照チャンネル/演算式> 信号名を取得するチャンネル番号を記述します。0と記述した場合は、全ての信号名を取得します。 なお、信号名が付けられていないチャンネルは先頭文字半角"#"に続きチャンネル番号が戻ります。
記述例	CHNM(0)
備考	戻り属性は文字属性となります。 解析対象ファイルが読み出されていない場合は実行時 ERROR となります。

2. 16. 3. 収録チャンネル単位名取得関数

機能	解析対象ファイルのチャンネルに付けられている単位を取得します。 read ch_name 文と等価な機能です。
文法	CUNT(k)
引数	【k】<必須><即値/数値属性参照チャンネル/演算式> 単位を取得するチャンネル番号を記述します。0と記述した場合は、全ての単位を取得します。 なお、単位が付けられていないチャンネルは半角スペース" "が戻ります。
記述例	CUNT()
備考	戻り属性は文字属性となります。 解析対象ファイルが読み出されていない場合は実行時 ERROR となります。

2. 16. 4. 収録チャンネル数取得関数

機能	収録チャンネル数を取得します。 read num_ch 文と等価な機能です。
文法	NCH()
引数	引数なし
記述例	NCH()
備考	戻り属性は数値属性となります。 解析対象ファイルが読み出されていない場合は実行時 ERROR となります。

2. 16. 5. 収録チャンネル番号取得関数

機能	解析対象ファイルの収録チャンネル番号を取得します。 read ch_series 文と等価な機能です。
文法	CHS()
引数	引数なし
記述例	CHS()
備考	戻り属性は数値属性となります。 解析対象ファイルが読み出されていない場合は実行時 ERROR となります。

2. 16. 6. マーク個数取得関数

機能	解析範囲に付けられている MARK 個数を取得します。 read num_mark 文と等価な機能です。
文法	NMK()
引数	引数なし
記述例	NMK()
備考	戻り属性は数値属性となります。 解析対象ファイルが読み出されていない場合は 0 を戻します。

2. 16. 7. マークメモ取得関数

機能	指定したマーク番号に付けられているマークメモを取得します。read mark_memo 文と等価な機能です。
文法	CMMR(k)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> マーク番号を記述します。マーク番号に 0 を記述した場合、全てのマークメモが戻ります。
記述例	CMMR(0)
備考	戻り属性は文字属性となります。 解析対象ファイルが読み出されていない場合は実行時 ERROR となります。

2. 16. 8. マーク番号 Index 取得関数

機能	収録時、または収録後の解析事前処理で波形に付けられた MARK 位置のデータ番号を戻します。
文法	MRK(k)
引数	【k】<必須><即値/数値属性参照チャネル/演算式> 検索する MARK 番号を意味し、即値、先頭文字"\$"に続きチャネル番号で表す数値属性参照チャネル、または演算式で記述します。なお、MARK 番号は解析範囲に含まれる最初のマークを 1 としたインクリメント番号で、解析対象範囲に該当する MARK 番号が存在していなければなりません。なお、MARK 番号に 0 を記述した場合は意味が異なり、解析範囲に付けられている全ての MARK 位置のデータ番号が戻ります。
記述例	MRK(2)
備考	k で指定された相対的 MARK 番号位置のデータ番号を戻します。 相対的マーク番号とは、解析範囲に指定された中で最も若い番号を 1 とした番号となります。 * 解析対象ファイルを読み込んでいないか、指定した MARK 番号が存在しない場合は、実行時エラーとなります。 MARK は、レコーダで収録中に付加する方法以外に、PcWaveForm の虫めがね機能により Post 処理過程でも付加することができます。解析対象ファイルが読み出されていない場合は実行時 ERROR となります。

2. 16. 9. 絶対マーク番号取得関数

機能	現在指定されている解析範囲に付けられている先頭 MARK 番号 1 が、解析対象ファイルの先頭から何番目に相当するかの絶対 MARK 番号を戻します。
文法	MKS()
引数	なし
記述例	MKS()
備考	解析対象範囲に MARK が付けられていない場や解析対象ファイルが読み出されていない場合は 0 が戻ります。

2. 16. 10. サンプリング周期取得関数

機能	現在設定されているサンプリング周期を戻します。
文法	PRD()
引数	なし
記述例	PRD()
備考	解析対象ファイルを読み出している場合は、当該ファイルのヘッダー部に記録されているサンプリング周期が戻り、Script 処理で解析対象ファイルを読み出していないか、def sampl_period 文で周期が定義されていない場合は実行時 ERROR となります。

2. 16. 11. 解析範囲開始 Index 取得関数

機能	解析対象範囲に指定された開始データ位置を、当該データの先頭から換算したデータ番号で戻します。
文法	STA()
引数	なし
記述例	STA()
備考	現在設定されている解析範囲の開始データ番号を戻します。解析範囲を指定して演算を行った場合、その開始データ番号は 0 に振り替えられています。STA 関数の戻り値は、解析開始データ番号が解析対象ファイルの先頭から振られたデータ番号を意味します。

2. 16. 12. サンプリング単位取得関数

機能	サンプリング単位を戻します。
文法	CXUT()
引数	なし
記述例	CXUT()
備考	戻り属性は文字属性となります。 サンプリング単位はヘッダーファイルの HORZ_UNIT 行に記載されている内容となります。

2. 16. 13. 収録開始年月日取得関数

機能	解析対象ファイルの収録開始年月日を戻します。
文法	CSDT()
引数	なし
記述例	CSDT()
備考	戻り属性は文字属性となりません。 収録開始年月日はヘッダーファイルの DATE 行に記載されている内容で MM-DD-YYYY の形式で戻ります。

2. 16. 14. 収録開始時刻取得関数

機能	解析対象ファイルの収録開始時刻を戻します。
文法	CSTM()
引数	なし
記述例	CSTM()
備考	戻り属性は文字属性となりません。 収録開始時刻はヘッダーファイルの TIME 行に記載されている内容で hh:mm:ss の形式で戻ります。

2. 16. 15. 収録開始 X 軸オフセット取得関数

機能	解析対象先頭データの X 軸オフセット値を戻します。
文法	XOF()
引数	なし
記述例	XOF()
備考	収録がプレトリガ収録されている場合など、先頭データは収録開始時刻より前の場合があり、オフセットとしてサンプリング周期の単位でヘッダーファイルの X_OFFSET に記載されています。

2. 16. 16. 収録データ数取得関数

機能	解析対象ファイルの収録データ数を戻します。
文法	NDT()
引数	なし
記述例	NDT()
備考	ヘッダーファイルの NUM_SAMPS 行に記載されている収録データ個数を取得します。

2. 16. 17. 解析対象ファイルコメント取得関数

機能	現在解析対象に指定されているヘッダーファイルに記載されているコメントを取得します。
文法	CCMT(k)
引数	【k】<必須><即値/数値属性参照チャンネル/演算式> 取得するコメント行を記述します。コメント行は 1~3 が有効となります。なお、0 を記述した場合は特別な意味を持ち、コメント行の全てを戻します。なお、コメント行が存在しない場合或いは存在しても記載されていない場合は半角スペース 1 文字を戻します。
記述例	CCMT(0)
備考	戻り属性は文字属性となります。 Script 文: 解析対象ファイルコメント取得文: read comment 文と同じ機能です。

2. 16. 18. カレントフォルダパス名取得関数

機能	カレントフォルダパス名を取得します。Script 文:カレントフォルダパス名取得文:read folder_path と同じ機能です。
文法	CFPN(k)
引数	<p>【k】<省略可><即値/数値属性参照チャネル/演算式> 取得するフォルダ先をコードで記述します。</p> <p>0: カレントフォルダ名を取得する 1: カレントフォルダ名と当該フォルダに含まれるフォルダ名を取得する 2: カレントフォルダ名と当該フォルダに含まれるフォルダ名及びそのフォルダに含まれるフォルダ名を取得する 記述省略した場合は 0 と見なします。</p>
記述例	CFPN(0)
備考	<p>戻り属性は文字属性となります。</p> <p>例えば、現在のフォルダ構成が以下の場合、フラグにより戻り内容が異なります。</p> <pre>c: ¥abc ← 現在のカレントフォルダ ¥xxx ¥zzz ¥yyy ¥aaa ¥bbb ¥ccc ¥ddd ¥eee</pre> <p>フラグ 0 の場合、戻りは 1 個で c:¥abc</p> <p>f フラグ 1 の場合、戻りは 3 個で c:¥abc c:¥abc¥xxx c:¥abc¥aaa</p> <p>フラグ 2 の場合、戻りは 9 個で c:¥abc c:¥abc¥xxx c:¥abc¥xxx¥zzz c:¥abc¥xxx¥zzz¥yyy c:¥abc¥aaa c:¥abc¥aaa¥bbb c:¥abc¥aaa¥ccc c:¥abc¥aaa¥ccc¥ddd c:¥abc¥aaa¥ccc¥ddd¥eee</p>

2. 16. 19. ファイルステータス取得関数

機能	カレントパスに指定されたファイルのステータスを取得します。
文法	RFC(&m)
引数	<p>【&m】<必須><文字列即値/文字属性参照チャネル/戻り文字属性演算式> ステータスをチェックするファイル名を記述します。</p> <p>複数要素を持つ文字列配列で記述した場合、戻りは数列となります。</p> <p>戻り値 ⇒ 0 存在していて、Open 可能 戻り値 ⇒ 1 存在しているが、現在使用中 Open 不可 戻り値 ⇒ 2 存在していない</p>
記述例	RFC(&1)
備考	<p>戻り属性は数値属性となります。</p> <p>Script 文:ファイル存在確認文:read file_check 文と同じ機能です。</p>

2. 16. 20. カレントチャンネル番号取得関数

機能	解析範囲に指定されたカレントチャンネル番号を取得します。
文法	CCH()
引数	なし
記述例	CCH()
備考	解析範囲が指定されていない場合は-1を戻し、解析範囲が指定されている(PcWaveForm 波形 Window で反転表示域存在する)場合は現在のカレントチャンネル番号を戻します。

関数索引

A		CNT(X)..... 47	ETH(a,b,c)..... 65
ABS(X)..... 8		CNTM()..... 102	EXP(X)..... 14
ACC(X)..... 8		CNV(k,m,n,f)..... 29	
ACS(X)..... 6		COR(X,Y)..... 22	F
AMP(k,X,m)..... 73		COS(X)..... 6	FFT(k,n,m,j,X)..... 20
AND(X,Y)..... 39		CPTV(k,&m)..... 101	FIR(k,X)..... 54
ASN(X)..... 6		CREC(k,n,&m)..... 98	FND(k,X,Y)..... 89
ASS(X,Y)..... 46		CREP(k,&s,&n,&m)..... 98	FTI(X)..... 38
ATN(X)..... 7		CRR(X,Y)..... 28	FVC(k,X,m)..... 48
ATS(k,n,m)..... 37		CSDT()..... 107	
B		CSEP((k,&m)..... 101	
		CSTM()..... 107	G
		CSTR(k, &m)..... 103	GMM(X)..... 29
		CTD(k,m,j,s,n,d,X)..... 84	GMX(a,b,c)..... 66
		CTN(&m)..... 99	GOV(k,X)..... 11
BIT(k,X)..... 45		CUNT(k)..... 105	GRT(k,X)..... 67
		CVTA(k,X,f)..... 102	GT_(X,Y,k)..... 42
C		CXUT()..... 106	GTE(X,Y,k)..... 43
			GUN(k,X)..... 11
CADJ(k,&m)..... 104		D	
CBSC(&m,k)..... 99		DAG(k,a,m,f,p)..... 68	H
CCH()..... 108		DEG(X)..... 7	HPF(k,X)..... 51
CCMT(k)..... 107		DEL(k,X)..... 78	HPR(k,X)..... 51
CCON(&n,&m,&k,....)..... 103		DIF(X)..... 9	HTV(X)..... 56
CDEC(k,n,&n,&m)..... 100		DIV(X)..... 10	
CDEL(k,n,&m)..... 97		DOP(k,m,X)..... 24	I
CEQ(&m,&n,k)..... 100		DTC(j,k,m,X,n,o,p,Y)..... 87	ING(k,X)..... 12
CERP(k,&n,&m)..... 99		DTD(k,m,j,n,X)..... 82	INL(X1,X2,X3,....,X10)..... 81
CEXT(k,n,&m)..... 97		DTE(k,m,n,X)..... 85	INS(k,X)..... 72
CFND(k,&n,&m)..... 97		DTF(j,k,m,n,X)..... 86	INT(X)..... 15
CFNM()..... 105		DTM(k,m,j,n,X)..... 83	INV(X)..... 13
CFPN(k)..... 108		DTP(k,m,X)..... 88	ITP(n,m,X)..... 79
CFT(k,X,m)..... 19			
CHNM(k)..... 105		E	L
CHS()..... 105		ECP(k,m,X)..... 41	LBF(k,X)..... 52
CHT(k,X,Y)..... 9		EFF(k,X)..... 16	LCF(k,X)..... 52
CINS(k,&n,&m)..... 98		ELM(&m)..... 99	LCM((k,X,Y)..... 13
CLN(&m)..... 99		EMN(a,b,c)..... 65	LEN(k,X)..... 78
CLNK(&n,&m,&k,....)..... 103		EMX(a,b,c)..... 65	LGT(X)..... 14
CMM(k,m,n,X)..... 96		ENV(k,X)..... 10	LMT(k,X,m)..... 75
CMMR(k)..... 106		EQP(k,m,X)..... 89	LNK(X,Y)..... 76
CMP(k,X,Y)..... 94		EQU(X,Y,k)..... 41	
CNCT(k,&m)..... 102		ERC(k,m,X)..... 69	
CNDT()..... 102			
CNSC(k,m,n,X)..... 103			

株式会社 デイシー

〒198-0024 東京都青梅市新町 9-2190

電話: 0428-34-9860

メール: info@deicy.co.jp

© Copyright 2006- DEICY Corporation