

波形の周期を求める

波形の周期を求める方法について説明します。尚、精度はサンプリング周期に依存し、求めた周期の最小分解能（精度）はサンプリング周期となります。

<論理波形、もしくは論理波形に変換して求める方法>

論理波形の立ち上がり地点のデータ番号を取得し、求め結果の隣同士を引き算して、データ個数を求めてサンプリング周期を掛けます。

step1: パルス波形を整形する。

\$1 に格納されているパルス波形が"0"と"1"の論理値で構成されている保障が無い場合は、論理値に変換します。変換の方法として、幾つかの方法があります。

case1: 波形の平均値より等しいか大きい場合を"1"として変換する場合

\$2 "論理波形:" = GTE(\$1,MEA(\$1))

case2: 波形の閾値を決め、閾値より大きい場合を"1"として変換する場合

※ 閾値を 0 として演算式を記述します。

\$2 "論理波形:" = NOT(LTE(\$1,0))

case3: 閾値を設定して変換する場合

※ 閾値近傍のノイズ等の影響を避ける場合は"1"の閾値と、"0"の閾値を設定して論理波形変換する RWC 関数を使用します。下記例では 0.5 を上昇で過ぎた値を"1"とし、0.3 を下降で過ぎた値を"0"として演算式を記述します。

\$2 "論理波形:" = RWC(0.5,0.1,\$1)

step2: 周期を求める

論理波形の"0"から"1"への立ち上がり地点のデータ番号を\$5 に求め、データ個数は\$5 のデータ番号隣り同士の引き算した結果、周期はデータ個数結果にサンプリング周期を掛け求めます。

\$5 "立ち上がり位置:" = LST(1,\$2)

/*パルスの"0"から"1"に遷移した位置を取得*/

\$6 "個数:" = ERC(1,LEN(\$5)-1,\$5)-ERC(0,LEN(\$5)-2,\$5)

/*立ち上がり間のデータ個数*/

\$7 "周期:sec" = \$6*PRD()

/*区間データ個数に周期を掛ける*/

\$8 "周波数:Hz" = 1/\$7

/*周期の逆数を求め周波数に変換*/

\$9 "累積時間:sec" = ACC(\$7)

/*累積して経過時間を求める*/

※ 周期相当データ個数は\$5_{n+1}-\$5_n で求めます。

step3: 周期又は周波数を波形データとして生成する

step2 で求めた結果の\$7,\$8 のデータ個数は周期(又は周波数)の個数となり、時刻歴データはありません。時刻歴データに変換する場合、元のデータ個数と同じ大きさの配列にデータ番号を生成して置き、演算結果の周期(又は周波数)を Y とし、そのデータ位置(検出周期の開始点/終了点)を X とした補間テーブルを作成して補間関数(ITP 関数)を使用して時刻歴波形を生成します。

\$10 "num_data:" = LEN(\$1)

/*データ個数を取得*/

assign \$11 = \$10<1>

/*データ番号配列を用意*/

\$11 "index_number:" = ACC(\$11)-1

/*データ番号配列の生成 0,1,2,3,4,5...*/

\$12 "x:" = MGR(ERC(0,LEN(\$5)-2,\$5),ERC(1,LEN(\$5)-1,\$5)-1)

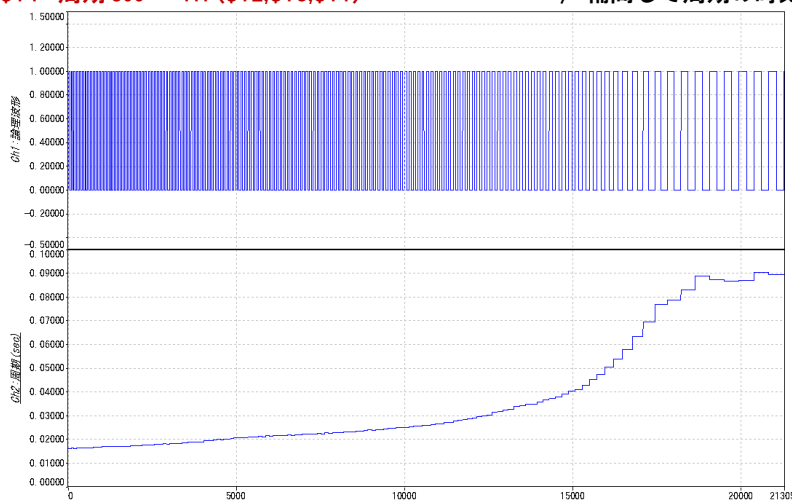
/*補間テーブル X 軸*/

\$13 "y:" = MGR(\$7,\$7)

/*補間テーブル Y 軸*/

\$14 "周期:sec" = ITP(\$12,\$13,\$11)

/*補間して周期の時刻歴波形*/



※ 周期（周波数）波形は、FVC 関数により直接求めることも出来ます。

<論理波形に変換することなく収録波形からデータ番号検索関数を使用して周期を求める方法>

step1: 設定した閾値通過地点(index)を求めます。

case1: 設定した閾値を指定方向で過ぎてから再び同方向に過るまでの周期を求める場合
 DTM 関数を使用し設定した閾値を過ぎた地点(index)を取得します。下記例は通り方向を上昇、閾値を 0.5 として記述しています。

```
$5 "閾値通過位置:" = DTM(1,0.5,1,0,$1) /*通り方向上昇,閾値 0.5,検索方向 index 順,検索開始 index0 */
```

case2: 設定した閾値を過ぎた後、設定したデータ個数以上過った状態が持続している時に有効として再び同じ条件で閾値を過ぎるまでの周期を求める場合
 DTE 関数を使用し設定した条件で閾値を過ぎた地点 (index) を取得します。下記例は通り方向を上昇、閾値を 0.5 とし通過後保持データ個数を 100 として記述しています。

```
$5 "閾値通過位置:" = DTE(1,0.5,100,$1) /*通り方向上昇、閾値 0.5、持続データ個数 100 */
```

case3: 設定した閾値を過ぎた後、設定した閾値範囲内に、設定したデータ個数以上、過った状態が保持している時に有効として再び同じ条件で閾値を過るまでの周期を求める場合
 CTD 関数を使用し設定した条件で閾値を過ぎた地点(index)を取得します。下記例は通り方向を上昇、閾値を 0、閾値通過保持データ個数を 100、保持閾値範囲を 0.5±0.1 として記述しています。

```
$5 "閾値通過位置:" = CTD(1,0.5,1,100,0.1,$1) /*通り方向上昇、閾値 0.5、持続データ個数 100、閾値保持 0.1*/
```

step2: 周期を求める

求めた閾値通過地点データの隣接したデータ同士を引き算してデータ個数を求め、求めたデータ個数にサンプリング周期を掛けて周期を求めます。

```
$15 "通過個数:" = LEN($5)
```

```
case $15 >= 2
```

```
proc detect{
```

```
  $6 "個数:" = ERC(1,LEN($5)-1,$5)-ERC(0,LEN($5)-2,$5) /*立ち上がり間のデータ個数*/
```

```
  $7 "周期:sec" = $6*PRD() /*区間データ個数に周期を掛ける*/
```

```
  $8 "周波数:Hz" = 1/$7 /*周期の逆数を求め周波数に変換*/
```

```
  $9 "累積時間:sec" = ACC($7) /*累積して経過時間を求める*/
```

```
}detect
```

※ データ番号検索関数を使用する場合指定した閾値を過らなると戻り値は-1 となります。その為、関数の戻り値個数が 2 個以上存在しているかの確認が必要です。

<或るチャンネルが指定した閾値を過ぎてから、別のチャンネルが指定した閾値を過るまでの時間を求める場合>

説明の都合上 ch1 が閾値 10 を上昇で過ぎてから ch2 が閾値 5 を上昇で過るまでの時間を求めます。

step1: 設定閾値通過データ番号を取得します。

```
$1 "ch1 立ち上り位置:" = DTM(1,10,1,0,#1)
```

```
$2 "ch2 立ち上り位置:" = DTM(1,5,1,0,#2)
```

※ 上記例では閾値を過るデータ位置検索関数に DTM 関数を使用しています。

step2: 時間演算が可能か否か判定フラグを求めます。

```
$3 "判定フラグ:" = AND(NEG($1(0),-1),NEQ($2(0),-1))
```

※ 判定フラグが 0 の場合は設定した閾値を通過していない。

step3: 求めた立ち上り位置から論理波形を生成した後、論理"1"範囲の時間を求めます。

```
case_true $3 /*判定フラグ=1 の時のみ演算します*/
```

```
proc detect{
```

```
  $4 = LEN(#1)
```

```
  assign $5 = $4<0> /*ch1 の立ち上がり地点用空配列の生成*/
```

```
  $6 = $5 /*ch2 の立ち上がり地点用空配列の複製*/
```

```
  $4 = LEN($1)
```

```
  assign $7 = $4<1> /*ch1 立ち上がり地点数分 1 を生成*/
```

```
  $4 = LEN($2)
```

```
  assign $8 = $4<-1> /*ch2 の立ち上がり地点数分-1 を生成*/
```

```
  $5 = SBV($1,$7,$5) /*ch1 の立ち上がり地点を 1 とした配列*/
```

```
  $6 = SBV($2,$8,$6) /*ch2 の立ち上がり地点を-1 とした配列*/
```

```
  $9 = ACC(ASS($5,$6)) /*$5 と $6 から論理波形の組み立て*/
```

```
  $10 "区間データ数:" = LST(1,$9)-LST(0,$9) /*論理波形の立ち上がりから立下りデータ個数*/
```

```
  $11 "時間:sec" = $10*PRD() /*データ個数×サンプリング周期*/
```

```
}detect
```