

応力波形の極大値(peak)/極小値(valley)のそれぞれ順位指定した値を求める

応力波形の最大極大値、最小極小値を求める場合は、当該チャンネルの最大値が最大極大値、最小値が最小極小値ですので、最大値関数 (MAX 関数) 及び最小値関数(MIN 関数)を使用して簡単に求めることが出来ますが、例えば、大きい順に3番目の極大値とか、同じく小さい順に3番目の極小値などを求める場合は、工夫が必要です。ここでは、指定した順位の極大値、極小値を求める方法について説明します。

<方法概要>

始めに、解析対象波形を、山谷検索関数 (PVF 関数) を使用して、波形を山 (極大値) と谷 (極小値) のみの波形に変換します。変換した山谷波形 (極大値/極小値波形) を極大値、極小値別に分解し、分解した波形を、データ並び替え関数 (SRT 関数) を使用して昇順及び降順に並び替え、並び変えた波形のデータ番号を指定して値を抽出する方法で求めます。

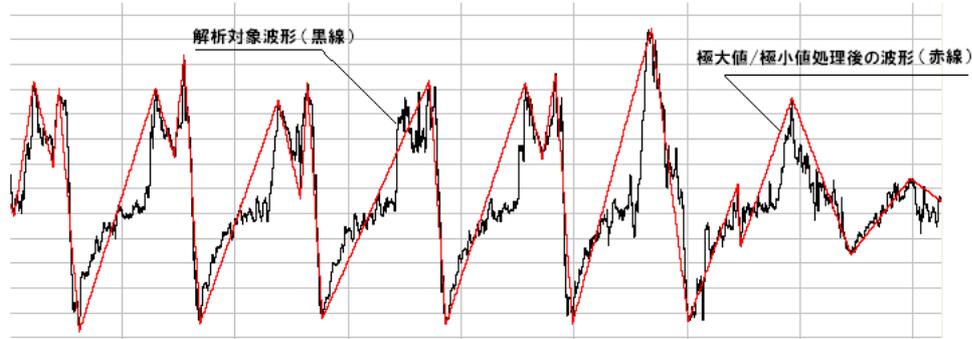
step1: 山谷検出関数を使用して極大値極小値だけのデータに変換します。

対象チャンネルを収録チャンネルの 1ch として説明します。

無効振幅値を仮に 5 とします。

$\$1$ "極大値_極小値:" = PVF(1,5,#1)

この結果、\$1 には極大値/極小値のみの数列が戻ります。



※ 上記図で描画した極大値/極小値波形は、重ねて表示する都合上、PVF のモードフラグ引数に-1 を指定して極大値極小値を直線補間して結んでいます。実際の処理手続きでは、記述の様にモードフラグを 1 と指定し、戻り数列は極大値極小値のみとします。

step2: \$1 の極大値/極小値数列を飛び越し抽出関数を使用して極大値と極小値に分解する

\$1 のデータ並びは、極大値⇒極小値⇒極大値...の並びか、極小値⇒極大値⇒極小値...の並びの何れかとなっています。そこで、\$1 を飛び越し抽出関数を使用して、2 種の数列に分解します。

$\$2$ = SEP(0,1,\$1)

$\$3$ = SEP(1,1,\$1)

この結果、\$2 は \$1 の index0,2,4,6.. の値の数列、\$3 は \$1 の index1,3,5,7... の数列となります。

step3: 極大値数列は降順並びに、極小値数列は昇順並びに変換します。

$\$4$ "極大値" = SRT(-1,\$2)*GTE(MAX(\$2),MAX(\$3))+SRT(-1,\$3)*NOT(GTE(MAX(\$2),MAX(\$3)))

$\$5$ "極小値" = SRT(1,\$2)*NOT(GTE(MAX(\$2),MAX(\$3))+SRT(1,\$3)*GTE(MAX(\$2),MAX(\$3))

この結果、\$4 には極大値の大きい順(降順)、\$5 には極小値の小さい順(昇順)に並び変わります。

※ \$2 と \$3 の何れが極大値かの判断は、最大値比較して大きい方が極大値数列となります。

step4: 抽出する順位のデータ値を取得します。

抽出する順位は極大値は大きい順の 3 番目、極小値は小さい順の 3 番目とします。

$\$6$ "3 番目の極大値:" = PTV(3-1,\$4)

$\$7$ "3 番目の極小値:" = PTV(3-1,\$5)

この結果、\$6 には、極大値の大きい順から 3 番目の値、\$7 には、極小値の小さい順から 3 番目の値が格納されます。

※ 説明の都合上、Step を分解して記述していますが、極大値順位での値を求める式と極小値順位での値を求める式の 2 式に集約して記述しても良い。但し、演算時間は分解した方が速いです。

順位指定して抽出した値の位置も求めたい場合について説明します。

<方法概要>

山谷検出関数 (PVF 関数) を使用して、極大値、極小値に波形データを縮退する時に、同時に、Index 値を求めて置き、それを極大値極小値位置に分解して並び替えた後に、順位のデータ番号を指定して読み出すことで求めます。

step1 : 対象波形を極大値、極小値だけのデータに変換します。

```
$1"極大値/極小値:" = PVF(1,5,#1) /*$1は極大値、極小値配列*/  
$8 "極大値/極小値位置:" = PVF(0.5,#1) /*$8は極大値、極小値位置配列*/
```

step2 : 極小値と極大値に分解する

```
$2 = SEP(0,1,$1) /*極大値極小値のデータ番号偶数の数列*/  
$3 = SEP(1,1,$1) /*極大値極小値のデータ番号奇数の数列*/  
$9 = SEP(0,1,$8) /*極大値極小値位置のデータ番号偶数の数列*/  
$10 = SEP(1,1,$8) /*極大値極小値位置のデータ番号奇数の数列*/
```

step3 : 極大値数列は降順並びに、極小値数列は昇順並びに変換します。

```
$11 = SRT(2,$2) /*極大値極小値の偶数番号数列の昇順キー*/  
$12 = SRT(-2,$2) /*極大値極小値の偶数番号数列の降順キー*/  
$13 = SRT(2,$3) /*極大値極小値の奇数番号数列の昇順キー*/  
$14 = SRT(-2,$3) /*極大値極小値の奇数番号数列の降順キー*/  
$4 "極大値" = QUE($12,$2)*GTE(MAX($2),MAX($3))+QUE($14,$3)*NOT(GTE(MAX($2),MAX($3)))  
$5 "極小値" = QUE($11,$2)*NOT(GTE(MAX($2),MAX($3)))+QUE($13,$3)*GTE(MAX($2),MAX($3))  
$15 "極大値位置" = QUE($12,$9)*GTE(MAX($2),MAX($3))+QUE($14,$10)*NOT(GTE(MAX($2),MAX($3)))  
$16 "極小値位置" = QUE($11,$9)*NOT(GTE(MAX($2),MAX($3)))+QUE($13,$10)*GTE(MAX($2),MAX($3))
```

step4 : 抽出する順位のデータ値を取得します。

抽出する順位は極大値は大きい順の3番目、極小値は小さい順の3番目とします。

```
$17 "3番目の極大値" = PTV(3-1,$4)  
$18 "3番目の極小値" = PTV(3-1,$5)  
$19 "3番目の極大値位置" = PTV(3-1,$15)  
$20 "3番目の極小値位置" = PTV(3-1,$16)
```

上記の\$19及び\$20はデータ番号ですが、時刻で求めたい場合は、サンプリング周期を掛けます。

```
$19 "3番目の極大値時刻:sec" = PTV(3-1,$15)*PRD()  
$20 "3番目の極小値時刻:sec" = PTV(3-1,$16)*PRD()
```