

収録データのリスAMPLING処理

収録データのリスAMPLING処理方法について説明します。

説明の都合上、リスAMPL対象チャンネルを収録チャンネル ch1 としています。

step1 : 収録したサンプリグ周波数とリスAMPLするサンプリグ周波数の最小公倍数を求めます。
`$1 "resampleFreq:Hz" = 3000 /* リサAMPL周波数を 3kHz として説明します*/`
`$2 "sampleFreq:Hz" = 1/PRD() /* 現在のサンプリグ周波数を取得します*/`
`$3 "lcm:Hz" = LCM(5,1/PRD(),1024) /*有効桁小数点以下 5 桁として最小公倍数を求めます*/`

step2 : 求めた最小公倍数から元波形のアップサンプルの必要があるか倍率を求めます。
`$4 "倍率:" = INT($3/$2)+0.5 /* LCM 周波数/収録サンプリグ周波数*/`

step3 : 最小公倍数周波数を使用してリスAMPL処理を行います。

case1 : 倍率が 1 以上の場合
 倍率が 1 以上の場合、一旦、最小公倍数で求めた周波数までアップサンプルする必要があります。アップサンプル処理はゼロを内挿してから、ローパスフィルタで波形接続した後、振幅を復元します。その後、設定されているリスAMPL周波数でサンプルします。
`case $4 > 1 /* 倍率が 1 以上の演算処理ブロック*/`

```
proc interpolation{
    $7 = INS($4-1,#1) /* ゼロ内挿後の波形データ*/
    $5 = 1/$3 /* 最小公倍数周波数を周期に変換*/
    def sampl_period $5 "sec" /* 最小公倍数周波数でカレント周期を定義*/
    $6 = LCF($2/2-1,LCF($2/2-1,$7)) /* ゼロ内挿した波形をローパスフィルタ処理して接続*/
    $6 = $6*MAX(#1)/MAX($6) /* 振幅復元*/
    $6 = SEP(0,$3/$1-1,$6) /* 振幅復元した波形からリスAMPL周期で抜き出し*/
    $1 = 1/$1 /* リサAMPL周波数を周期の変換*/
    def sampl_period $1 "sec" /* リサAMPL周期でカレント周期を定義*/
}interpolation
```

- ※ ゼロ内挿は倍率-1 個の 0 を元の波形データのサンプル間隔の間に挿入します。
- ※ フィルタ遮断周波数は元のサンプリグ周波数（収録サンプル周波数）の 1/2 とします。
- ※ 波形復元は元の最大値とローパスフィルタ処理後の最大値の比率を掛けます。
- ※ リサAMPL抜き出しは最小公倍数周波数とリスAMPL周波数の比-1 毎に振幅復元後の波形から抜き出します。

case2 : 倍率が 1 の場合
 倍率が 1 の場合、設定したリスAMPL周波数が元のサンプリグ周波数（収録サンプル周波数）の整数分の 1 であることを意味し、アップサンプルの必要はなく単純なダウンサンプルとなります。

```
case $4 = 1
proc downSample{
    $6 = LCF($1/2-1,LCF($1/2-1,#1)) /*設定されているリスAMPL周波数/2-1 でフィルタ処理*/
    $6 = SEP(0,$3/$1-1,$6) /* フィルタ処理後波形からリスAMPL周期で抜き出し*/
    $1 = 1/$1 /* リサAMPL周波数を周期の変換*/
    def sampl_period $1 "sec" /* リサAMPL周期でカレント周期を定義*/
}downsample
```

- ※ サンプリグ定理から導かれたリスAMPL関数 RSM 関数を使用して求めることもできますが、データ数が多くなると演算時間が掛る為、上記方法を推奨します。