

WIN システム用波形モニターツールの活用

鶴岡 弘*

Practical Use of The Waveform Monitor Tool for WIN System

Hiroshi TSURUOKA*

はじめに

地震研究所では、衛星テレメタリングシステムの副局が維持・管理されており、大学・気象庁・防災科学技術研究所等で管理運営している地震観測点からの 5000 チャンネルを越える地震波形データが WIN システムによってリアルタイムに集配信されている。衛星からのデータは IP パケットとして所内の専用のネットワークセグメントに配信され、震源決定等の基礎データとして利用されている。これらの解析はハードディスク上に保存されたデータに対して行われてきたが、今後地震波形のリアルタイムモニタリングの必要性が高まっていくであろうと考えられる。本報告では WIN システム用波形モニターツール（鶴岡・ト部, 2002, 2003）の活用方法について技術的な面から述べる。これらのツールは IXP プロットライブラリ（鶴岡, 2002）を利用しており、X, ポストスクリプト, PNG, JPG への出力が可能であるので幅広く活用できる。ここでは、(1)リアルタイム波形モニタリング, (2)波形モニター出力, (3) Web との連携 (CGI アプリケーション) への利用について簡潔に述べる。

WIN システムによる波形データ

まずは、ネットワークを流れる地震波形データが WIN

システムによってどのようにディスクに保存されるかについて述べる。モニターツールを活用する上でこれを把握しておくことは重要である。衛星から配信されるデータはプログラム recvt によって共有メモリ 11 番に書き込まれる。この段階では必ずしも時間順にソートされていないので、プログラム order で時間順に整列（バッファリング）され、共有メモリ 12 番に書き込まれる。最後に、プログラム wdisk によってハードディスクに 1 分ごとのファイルに分割されて特定のディレクトリ (/dat/raw) に保存される。図 1 にこれらの流れをまとめて示す。WIN システムでは、さらにディスク上に保存された波形データに対して自動検出、自動震源決定等が行われる。

WIN システム用波形モニターツールの特徴

WIN システム用波形モニターツールは図 1 における共有メモリ 11 番や 12 番およびディスク上の WIN フォーマットデータをテキストベースでデータ内容をダンプする既存のプログラム shmdump に波形モニター用の新たなフォーマット出力を機能拡張し、そのフォーマット出力を入力として波形表示するプログラム群（shmx, shmp, shmp1）から構成されている。このような機能分割によってプログラムの柔軟性・拡張性が高められている。

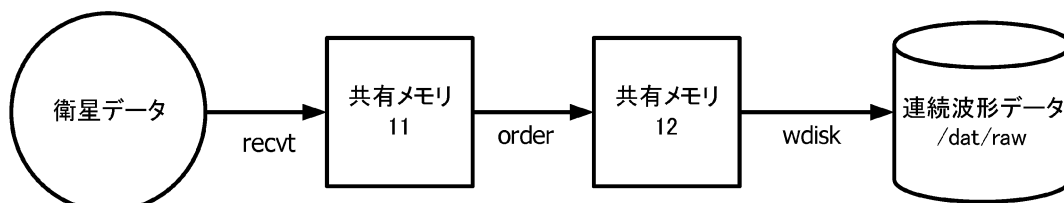


図 1. 衛星からの地震波形データがディスクに保存されるまでのフロー

2003 年 9 月 8 日受付, 2003 年 10 月 24 日受理.

* 東京大学地震研究所地震予知情報センター.

* Earthquake Information Center, Earthquake Research Institute, University of Tokyo.

リアルタイム波形モニタリング

ディスクに保存される波形データの表示は最大でリアルタイムより1分+orderのバッファリング時間だけ遅れるため、リアルタイム波形表示には共有メモリ中にある波形データを利用することが必須である。まずは、時間順にデータがソートされて共有メモリ12番に書き込まれた波形データをモニターする手順について説明する。ツールとしてはshmdump, shmp (単一チャンネル), shmp1 (マルチチャンネル) を利用する。

- ・単一チャンネルモニター

チャンネルID=0252のデータをモニターする場合は以下のコマンドを実行する。

```
% shmdump -tq 12 0252 | shmp 0252 -m1 -n60
```

出力結果の様子を図2に示す。1トレース1分のデータが60行表示され、1時間の波形モニターとなっている。

- ・マルチチャンネルモニター

```
% shmdump -tq 12 0252 0253 0254 ... |
```

```
shmp1 0252 0253 0254 ... -m10
```

出力結果の様子を図3に示す。単一・マルチチャンネルとも1トレースの表示時間はオプションを指定することで設

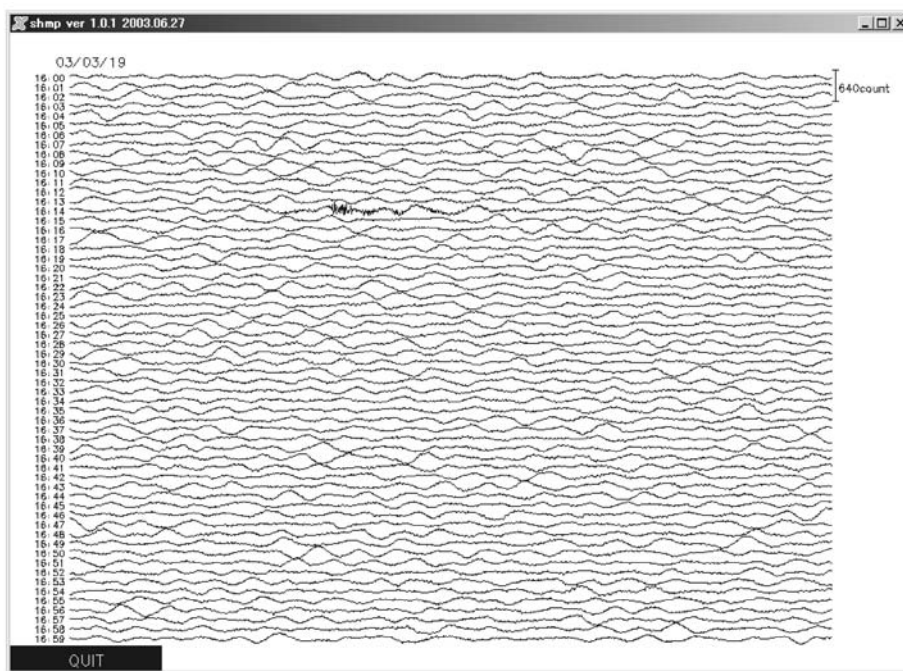


図 2. 単一チャンネルモニター出力例

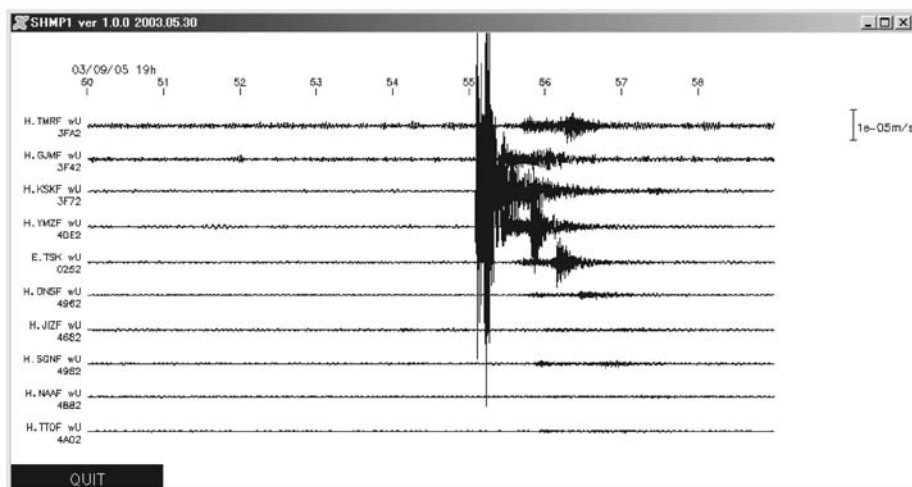


図 3. マルチチャンネルモニター出力例

定が可能である。また豊富なオプション群により柔軟なカスタマイズができるようになっている。

よりリアルタイム表示を求める場合には共有メモリ 11 番に書き込まれている波形データを利用する必要がある。そのためのツールとして shmx がある。shmx は wish というスクリプト言語で作成されたプログラムであるため shmp, shmp1 よりも CPU の負荷が高いが、よりリアルタイム表示が求められる場合に活用できる。利用には同様に shmdump と組み合わせて使用する。

モニター波形ポストスクリプト出力

データ管理の観点から 1 チャンネルのデータ 1 日分の出力を自動的にプリンターに出力することもこれらのツールを利用すれば簡単である。リスト 1 のシェルスクリプト PLOT1D.sh を作成して、crontab にリスト 2 のように登録すれば、図 4 に示される高品質の自動出力が可能である。リスト 2 の設定では、毎日午前 9 時にチャンネル 0252, 0253 の波形データが出力される。

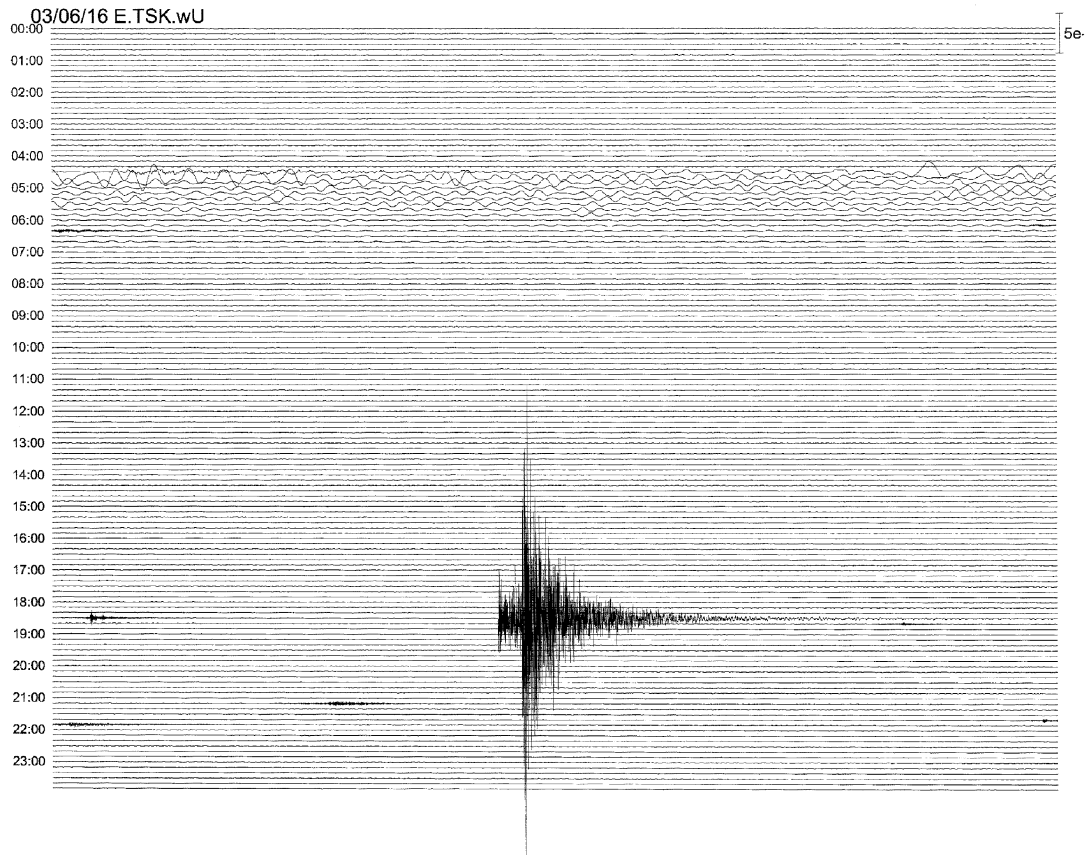


図 4. 波形モニター出力例

Web との連携

活用例の最後として、任意の単一チャンネル・時間の波形データを動的に作成する Web アプリケーションの構築について述べる。Web ページはリスト 3, 4, これらのページから呼び出される CGI スクリプトがリスト 5 である。利

用の様子を図 5 に示す。ユーザは表示させたいチャンネルの ID および時間等を入力して PLOT ボタンをクリックするだけでよい。Web アプリケーションにすることによって、ユーザはアプリケーションのインストール、データ管理から解放されるので、複数ユーザが利用する場合は有効な活用の仕方である。

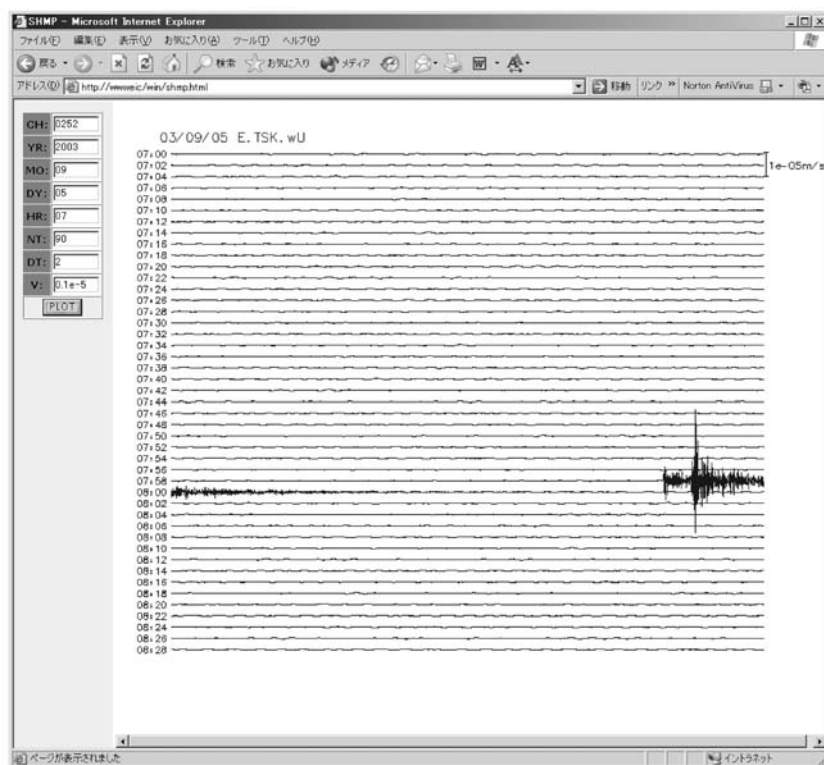


図 5. WWW 利用例

おわりに

WIN システム用波形モニターツールの活用例について述べてきたが、本報告以外の活用例もあるだろう。例えば、SAC ファイルから shmp 等で表示可能なフォーマット出力を行うプログラムを作成すれば WIN フォーマット以外にも利用可能であろう。また、最近では、WIN フォーマットでの出力が可能なテレメタリングシステムが市販されているので、これらのシステムからのデータに異常がないかななどのチェックにも活用できる。今後は波形データのモニタリングが重要である。ツールを有効に利用していただき

たい。

謝 辞：阿部勝征教授と渡辺秀文教授の査読は本稿の改善に役立ちました。ここに記して感謝します。

文 献

- 鶴岡 弘, 2002, Web に適したプロットライブラリの改良, 震研技報, No. 8, 46-49.
- 鶴岡 弘・ト部 卓, 2002, WIN システム用波形モニターツールの開発, 地震学会講演予稿集, 2, P089.
- 鶴岡 弘・ト部 卓, 2003, WIN システム用波形モニターツールの開発 (2), 地球惑星科学合同大会, S048
- WIN システム <http://eoc.eri.u-tokyo.ac.jp/WIN/>.

リスト1. モニター出力用シェルスクリプト.

```
#!/bin/sh
if [ x$1 = x ] ; then
  echo "Usage $0 CHID"
  echo "      $0 CHID YYMMDD"
  echo ""
  echo " e.g. $0 0252"
  echo "      $0 0252 030901"
  exit
fi
PATH=/bin:/usr/bin:/usr/local/win/bin
export PATH
CHID=$1
YYMMDD=$2
if [ x$YYMMDD = x ]; then
  YYMMDD=`date --date '1 days ago' '+%y%m%d'`
fi
RAWDIR=/dat/raw
cat $RAWDIR/${YYMMDD}???.?? | ¥
shmdump -tq - $CHID | ¥
shmp $CHID -w10 -oshmp.$CHID.$YYMMDD -m10 -n144
lpr shmp.$CHID.$YYMMDD.ps
```

リスト2. crontab の設定

```
0 9 * * * /home/auto/PLOT1D.sh 0252 > /dev/null 2>&1
0 9 * * * /home/auto/PLOT1D.sh 0253 > /dev/null 2>&1
```

リスト3. shmp.html

```
<html>
<head><title>SHMP</title></head>
<frameset cols="120,*" border=1>
  <frame src="form.html" name="FORM">
  <frame name="PLOT">
</frameset>
</html>
```

リスト4. form.html

```
<html><body bgcolor="#eeeeee"><font face="Verdana">
<form action="/cgi-bin/shmp.pl" method="post" target="PLOT">
<table cellpadding=2 cellspacing=0 border=1>
  <tr><th bgcolor="#3399cc">CH:</th>
    <td bgcolor="#ffffff"><input name=CH size=8 value="0252"></td></tr>
  <tr><th bgcolor="#3399cc">YR:</th>
    <td bgcolor="#ffffff"><input name=YR size=8 value="2003"></td></tr>
  <tr><th bgcolor="#3399cc">MO:</th>
    <td bgcolor="#ffffff"><input name=MO size=8 value="08"></td></tr>
  <tr><th bgcolor="#3399cc">DY:</th>
    <td bgcolor="#ffffff"><input name=DY size=8 value="22"></td></tr>
  <tr><th bgcolor="#3399cc">HR:</th>
    <td bgcolor="#ffffff"><input name=HR size=8 value="18"></td></tr>
  <tr><th bgcolor="#3399cc">NT:</th>
    <td bgcolor="#ffffff"><input name=NT size=8 value="60"></td></tr>
  <tr><th bgcolor="#3399cc">DT:</th>
    <td bgcolor="#ffffff"><input name=DT size=8 value="1"></td></tr>
  <tr><th bgcolor="#3399cc">V:</th>
    <td bgcolor="#ffffff"><input name=V size=8 value="0.1e-5"></td></tr>
  <tr><th align="center" colspan=2><input type="submit" value="PLOT"></th></tr>
</table>
</form>
</font></body></html>
```

リスト 5. CGI スクリプト

```
#!/usr/bin/perl
$| = 1;
$WCAT='/dat/web/bin/wcat';
$SHMDUMP='/usr/local/win/bin/shmdump';
$SHMP='/usr/local/win/bin/shmp';
$TABLE = '/dat/etc/channels.tbl';
require '/dat/web/lib/cgi-lib.pl';
&ReadParse(*in);
$YR=$in{'YR'}; $MO=$in{'MO'}; $DY=$in{'DY'}; $HR=$in{'HR'};
$NT=$in{'NT'}; $DT=$in{'DT'}; $CH=$in{'CH'}; $V=$in{'V'};
if ( $V eq '' ) { $V=0.1e-5; } $N=$NT/$DT;
system("$WCAT /dat/raw $YR $MO $DY $HR 0 $NT |
      $SHMDUMP -tq - $CH |
      $SHMP -Aab -m$DT -n$N -w11 -t$TABLE -um/s -v$V -o/tmp/shmp.$$ $CH
      > /dev/null");
printf("Content-type:image/png\n\n");
system("/bin/cat /tmp/shmp.$$png");
unlink("/tmp/shmp.$$png");
```