

# WIN システム用シミュレーションツールの紹介

鶴岡 弘\*

Introdcuton of Simulation Tools for WIN System

Hiroshi TSURUOKA\*

\*東京大学地震研究所地震予知情報センター

\* Earthquake Information Center, Earthquake Research Intitute, University of TOKyo

## はじめに

WINシステムによってリアルタイムに配信されている地震データをリアルタイムにモニターするツールについては、鶴岡(2003)においてその活用を含めて紹介した。本報告では、実際の地震波形の配信をシミュレートできるツールを紹介する。波形データを実際に受信する環境を構築しなくても、リアルタイム地震解析、例えば、震源決定やモーメントテンソルインバージョンのテストが簡単に可能となるので、アルゴリズムの改良やシステムの負荷テストなどが容易になる。このことはシステムの信頼性向上や WIN システムを用いる地震解析システムの普及に役立つと考えられる。

## WIN フォーマットについて

データ転送における波形フォーマットは、国内においてはWINフォーマットであり、特長はマルチチャンネルかつ異なるサンプリング周期をもつデータを1秒毎の時刻情報付き可変長の秒ブロックとして構成されている点である。データは圧縮されており、時間分割・統合が容易であるとともに、WINフォーマットを扱う様々なツール群が利用できる。データはエンディアンが異なるPCやワークステーションでも同一に扱えるよう機種に依存せずビッグエンディアンで保存されている。WINフォーマットを直接に扱うことによっても波形モニターや波形伝送シミュレーションを行うことは可能であるが、WINフォーマットを扱うAPIを用いなくてもデータ処理ができるような中間フォーマットを採用すると様々なツールやプログラムが活用できるようになる。ここでは、鶴岡(2003)で波形モニターの際に用いた shmdump のテキスト形式での出力を利用して、配信シミュレートのためのツールを作成し、そのためのデータフォーマットを提案する。

## WIN テキスト形式フォーマット

波形モニターに利用したWINテキスト形式フォーマットは、表1に示されるような形式で出力される。この形式にすることにより、WINフォーマットの波形データをテキスト形式の波形データに変換できていると考えられる。WINフォーマットの特長である時刻ラベル付きのマルチチャンネルかつ

---

\* tsuru@eri.u-tokyo.ac.jp

異なるサンプリング周期をもつデータが表現できており、時間統合が簡単であるという WIN フォーマットの特長も保持できている。また、WIN フォーマットと WIN テキスト形式フォーマットは、相互に交換できるよう wintowin という WIN テキスト形式フォーマットから WIN フォーマットに変換するツールを開発した。それぞれのフォーマットと変換ツールの関係をまとめて図1に示しておく。

### 配信地震データ記録のためのツールおよびデータフォーマット

衛星通信あるいは ADSL や光ファイバーなどの地上 IP 網を流れる WIN フォーマットのデータを受信、ディスクに保存するまでの流れを図2に示している。order 後のデータは1秒毎に波形データがまとめられるので、シミュレートするのは簡単であるが、問題は recvt 後に 共有メモリ 11 番に書き込まれるデータをシミュレートする場合である。共有メモリ 11 番には各チャネルのデータが非同期で書き込まれている。この非同期に書き込まれるデータを shmdump によりテキスト形式で出力し、その時間差を記録して保存することをすればデータ配信を記録することができる。表 2 に今回開発した shmsimu というプログラムのソースを示している。Tcl というプログラム言語のスク립トとして開発した。Tcl は時間差を msec 単位で簡単にコーディングできること、複数(UNIX, Windows および MAC)のプラットフォームで動作可能であることなどから採用した。各時刻ラベルのデータが配信される時間差を計算し保存するために、データフォーマットは表 3 のようにWINテキスト形式に DT の指示子のあとに時間差を msec で表記するデータフォーマットとした。

実際の利用例を以下に示す。

- ・ データ記録

```
% shmdump -tq 11 0252 0690 | shmsimu > shmsimu.dat
```

- ・ データ再生 (プログラムオプションに -p を指定する。)

```
% cat shmsimu.dat | shmsimu -p
```

地震解析のシミュレーションには、ディスクに保存されたデータを簡単にシミュレートできればよいと考えられる。これは、order 前のデータをそのまま利用することは、波形モニター以外ではあまりないからである。この場合には、1秒毎に波形データが出力されればよい。表 3 のフォーマットを用いることはなく、秒ブロックを一定に出力できるツールを開発すればよい。表 4 にそのプログラム winsimu のリストを示す。shmsimu に比べてプログラムが単純である。利用の実際例は以下となる。

```
% cat 05082210.0? | shmdump -tq - 0252 0690 | winsimu
```

この例は 2005年8月22日10時から10分間のWINフォーマットデータからチャンネル 0252 と 0690 を出力している例である。任意の時刻および時間長のデータをシミュレートできるように wincat という perl スクリプトも開発した。利用方法は

```
% wincat RAWDIR YR MO DY HR MI LEN
```

であり、RAWDIR ディレクトリ に保存された WIN フォーマットの分ファイルを時刻 YR/MO/DY HR:MI から LEN 分間分 cat する簡単なスクリプトである。

## シミュレーションツール利用例

ここでは、shmsimu, winsimu, wincat という WIN システム用シミュレーションツールの具体的な利用例を挙げてみる。

```
% cat shmsimu.dat | shmsim -p | wintowin 11 128 (A)
```

```
% shmdump -tq 11 0252 | shmx 0252 (B)
```

(A)において、shmsimu を用いて記録された表2のフォーマットのデータを shmsimu により再生し、wintowin で WIN フォーマットに変換、共有メモリ 11 番に書き込み、(B)において、その共有メモリ 11 番に書き込まれたデータのうち CHID:0252 を出力し、波形のモニタリングをしている。この(A)と(B)は、

```
% cat shmsimu.dat | shmsimu -p | shmx 0252
```

としても同じことである。

```
% send_raw -rt 11 172.17.192.192 7004 - /dat/log/send_raw.log (C)
```

(A)において共有メモリ 11 番に書き込まれたデータを send\_raw を利用して、ホスト(172.17.192.192)のポート7004に送信している例である。ただし、shmsimu.dat には、過去のデータが記録されているため、172.17.192.192 で recvt でデータを受信すると ログにタイムスタンプが間違っているという Warning が出力される。この(C)の例はタイムスタンプがおかしい(過去の)データを送信するテストにもなる。地震解析のシステムテスト等には、ローカルホスト内で以下のようなコマンドを実行することが多いだろう。

```
% wincat /dat/raw 2005 08 22 10 0 20 | shmdump -tq - 0252 0690 48C3 | ¥
```

```
winsimu | wintowin 12 256 (D)
```

(D)の場合はディスク上に保存された WIN データから、CHID:0252,0690,48C3 の波形データを共有メモリ 12 番に書き込む例である。共有メモリに書き込む利点は複数のプログラムからデータを共有できる(つまり、同じ波形データを利用する複数のプログラムを実行できる)ということである。

## おわりに

本報告で作成したツールを利用することにより、リアルタイムに配信される地震波形データをシミュレートし、リアルタイムで震源決定や波形インバージョンを行うシステムのテストが容易に行える環境を構築することが可能である。我々はこのツールを長周期波動場のモニタリングによるリアルタイム地震解析システムの開発に活用している。気象庁や防災科研においても数秒で震源を決定するシステムを開発しているが、同種の解析システムのシミュレーションにも利用できるだろう。今回開発したツール群は WIN のパッケージに組み込む予定なので、WIN システムのホームページからダウンロード可能である。

**謝辞:** 岩崎貴哉教授と新谷昌人助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

WIN システム <http://eoc.eri.u-tokyo.ac.jp/WIN/>.

鶴岡 弘, 2003, WIN システム用波形モニターツールの活用, 震研技報, No. 9, 14-17.

表1. WIN テキスト形式フォーマット

```
DATA FORMAT
YR MO DY HR MI SC nch
CH1 freq1 (data(i),i=1,freq1)
CH2 freq2 (data(i),i=1,freq2)
...
CHnch freqnch (data(i),i=1,freqnch)
...
出力例
02 07 25 22 13 23 3
0252 10 2 1 0 0 -1 -2 -3 -4 -5 -5
0253 10 -13 -13 -14 -15 -15 -16 -16 -17 -17 -17
0254 10 12 12 13 13 14 14 14 15 15 15
02 07 25 22 13 24 3
0252 10 -6 -7 -7 -8 -9 -10 -10 -11 -12 -13
0253 10 -17 -17 -17 -16 -16 -15 -15 -14 -13 -12
0254 10 16 17 17 18 19 20 21 22 23 25
```

表2. プログラム:shmsimu

```
#!/usr/bin/tclsh
set fpin stdin
set fpout stdout
set fp $fpin
set cat 0
set speed 1
for { set j 0 } { $j < $argc } { incr j } {
    set tmpargv [lindex $argv $j]
    set optflag [string range $tmpargv 0 1]
    if { $optflag == "-p" } {
        set cat 1
    } elseif { $optflag == "-s" } {
        set speed [string range $tmpargv 2 [string length $tmpargv]]
    }
}
if { $cat == 0 } {
    set tm0 [clock clicks -milliseconds]
}
while {[eof $fp]} {
```

```

if { $cat == 0 } {
    set tm [clock clicks -milliseconds]
    set dtm [expr $tm - $tm0]
    puts "DT $dtm"
} else {
    gets $fp line
    scan $line "DT %d" dtm
    set ms [expr $dtm/$speed]
    after $ms
}
gets $fp line
scan $line "%d %d %d %d %d %d %d" yr mo dy hr mi sc nch
puts $line
for { set i 0 } { $i < $nch } { incr i } {
    gets $fp tmpdata
    puts $tmpdata
}
flush $fpout
if { $cat == 0 } {
    set tm0 $tm
}
}
close $fp

```

表 3. シミュレーション用データフォーマット

```

05 10 02 19 12 28 1
0690 10 38 27 13 -1 -8 -14 -20 -30 -36 -45
DT 428
05 10 02 19 12 29 1
0252 10 360 329 329 314 302 301 298 313 317 323
DT 550
05 10 02 19 12 30 1
0252 10 317 322 321 320 307 299 301 290 291 290
DT 330
05 10 02 19 12 29 1
0690 10 -57 -64 -70 -80 -86 -96 -100 -104 -95 -83
DT 771

```

表4. プログラム : winsimu

```

#!/usr/bin/tclsh
set fpin stdin
set fpout stdout
set fp $fpin
set cat 1

```

```

set speed 1
for { set j 0 } { $j < $argc } { incr j } {
    set tmpargv [lindex $argv $j]
    set optflag [string range $tmpargv 0 1]
    if { $optflag == "-s" } {
        set speed [string range $tmpargv 2 [string length $tmpargv]]
        set ms [expr 1000/$speed]
    }
}
while {![eof $fp]} {
    gets $fp line
    scan $line "%d %d %d %d %d %d %d" yr mo dy hr mi sc nch
    puts $line
    for { set i 0 } { $i < $nch } { incr i } {
        gets $fp tmpdata
        puts $tmpdata
    }
    flush $fpout
    after $ms
}
close $fp

```

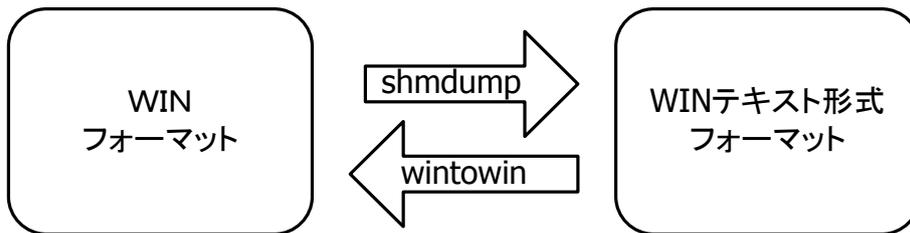


図1. WIN フォーマットと WIN テキスト形式フォーマットおよび変換ツール

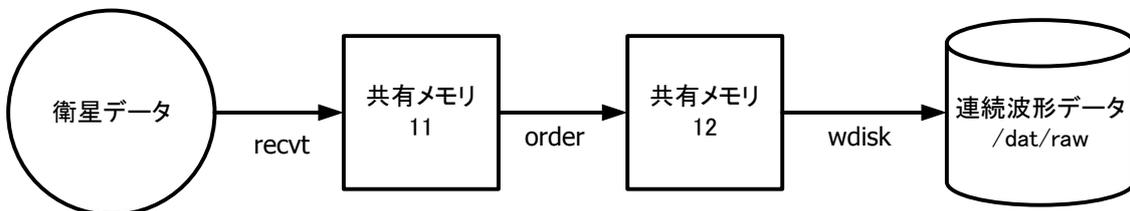


図2. 地震波形データがディスクに保存されるまでのフロー