Proposal for a Binary Receiver Independent Exchange Format

Kees de Jong and Hans van der Marel

Delft University of Technology, Department of Geodesy
Thijsseweg 11, 2629 JA Delft, The Netherlands
e-mail: K.deJong@geo.tudelft.nl

Summary

The introduction of the Receiver Independent Exchange Format (RINEX) in 1989 was an important milestone. In its present form, using Hatanaka and standard Unix compression tools, it is quite an efficient format to store GPS data on a daily, or even hourly basis.

However, both existing and new applications of GPS have a tendency to go to real-time operation and use higher sample rates. In order to support these applications there is a need for a binary receiver independent exchange format. This format should meet the following requirements: receiver independent, small and compact data format (must be able to handle 1 second data), no predefined boundaries, upward and downward compatible and flexible. In this paper we propose a format, NCBI (New Compressed Binary) that meets these requirements.

Test software, which is freely available from DUT, has been developed to generate NCBI files from RINEX and vice versa. This software runs on both big- and little-endian systems.

1. Introduction

The compressed binary (CBI) format was developed in 1993-1995 for the Dutch Active GPS Reference System (AGRS.NL). Experiences over the past five years revealed some limitations of this format, such as the relatively limited resolution of code observations. In this document a new receiver-independent compressed binary (NCBI) format is proposed for GPS and GLONASS P1, P2 and C/A code and carrier observations. In addition, L1 and L2 Doppler observations and P1, P2 and C/A SNR's are supported.

The building blocks of the format are messages. Each message starts with a one-byte identifier. This identifier contains information on the type of message, including the satellite system used and the receiver tracking scenario. The following messages are currently supported:

- station, receiver and antenna data
- broadcast ephemerides data
- events
- meteorological data
- observation data

The basic idea behind the format is that it forms a continuous stream of data. Whether the data is transmitted continuously, in bursts, or stored in files makes no difference. A new file can be started almost anywhere in the data stream. All records are self-supporting.

2. Observation messages

For an observation epoch, three types of messages are required:

1. Start of epoch (contains time of observation)
2. Observations
3. End of epoch (to indicate the end of an observation epoch)

The start of epoch contains the time of observations. The time of observation t is the same for all tracked satellites. It can be stored as number of seconds since 6 Jan 1980 00:00:00 and fraction of a second. Assuming a resolution of 1 ms and a precision of ± 0.5 ms, the algorithm for the compression of time is given by ($\lfloor . \rfloor$ denotes rounding down):

$$t_{\sec,NCBI} = \lfloor t \rfloor \text{ s}$$
$$t_{frac,NCBI} = \left[\left(t - \lfloor t \rfloor\right) \cdot 10^3 \right] \text{ ms}$$

The seconds are stored in 32 bits and the fractional part in 10 bits. The compressed time $t_{\sec,NCBI}$ is valid until the beginning of the 22nd century.

It is assumed that at each observation epoch, a receiver outputs one or more of the following observations (p denotes code [m], ϕ carrier [cycles]):

p_1 , ϕ_1 , ϕ_2 , p_2 , $p_{C/A}$, $\phi_{C/A}$

in one of the following combinations: { p_1 }, { p_1 , ϕ_1 }, { p_1 , ϕ_1 , ϕ_2 }, { p_1 , ϕ_1 , ϕ_2 , p_2 } (non-cross correlating and cross correlating), { p_1 , ϕ_1 , ϕ_2 , p_2 , $p_{C/A}$ } or { p_1 , ϕ_1 , ϕ_2 , p_2 , $p_{C/A}$, $\phi_{C/A}$ }. Doppler and SNR observations are assumed to be available in addition to the code and carrier observations. In order to avoid too many different NCBI messages, only the following combinations of code and carrier, Doppler and SNR observations are assumed to occur

Code and carrier	**SNR**	**Doppler**
p_1	None or SNR_1	None
p_1 , ϕ_1	None or SNR_1	None or D_1
p_1 , ϕ_1 , ϕ_2	None or SNR_1	None or D_1
p_1 , ϕ_1 , ϕ_2 , p_2	None or (SNR_1 , SNR_2)	None , D_1 or (D_1 , D_2)
p_1 , ϕ_1 , ϕ_2 , p_2 , $p_{C/A}$	None or (SNR_1 , SNR_2 , $SNR_{C/A}$)	None , D_1 or (D_1 , D_2)
p_1 , ϕ_1 , ϕ_2 , p_2 , $p_{C/A}$, $\phi_{C/A}$	None or (SNR_1 , SNR_2 , $SNR_{C/A}$)	None , D_1 or (D_1 , D_2)

The size of each message is indicated below

Start of epoch	**Size (bits)**
Identifier	8
$t_{\sec,NCBI}$	32
$t_{frac,NCBI}$	10
Total	50
Total (bytes)	7

End of epoch	**Size (bits)**
Identifier	8
Total	8
Total (bytes)	1

Doppler	**Size (bits)**	
$D_{1,NCBI}$	24	24
$D_{2,NCBI}$		12
Total	24	36
Total (bytes)	3	5

Code and carrier	**Size (bits)**					
Identifier	8	8	8	8	8	8
Satellite identifier	6	6	6	6	6	6
$p_{1,NCBI}(t)$	34	34	34	34	34	34
$\phi_{1,NCBI}(t)$		21	21	21	21	21
$\phi_{2,NCBI}(t)$			21	21	21	21
$p_{2,NCBI}(t)$				17	17	17
$p_{C/A,NCBI}(t)$					17	17
$\phi_{C/A,NCBI}(t)$						10
Total	48	69	90	107	126	134
Total (bytes)	6	9	12	14	16	17

SNR	**Size (bits)**		
$SNR_{1,NCBI}$	8	8	8
$SNR_{2,NCBI}$		8	8
$SNR_{C/A,NCBI}$			8
Total	8	16	24
Total (bytes)	1	2	3

The satellite identifier was chosen as a six-bit number in the range 0-63. To avoid needless complexities in compression and decompression, blocks of code and carrier, Doppler and SNR observations are aligned at byte-boundaries.

3. Compression of observations

The observations, Doppler and SNR are stored in a few bytes only. For the observations the following assumptions are made:

1. Required resolution: 1 mm in code, 1 mcycle in carrier.
2. The code observation p_1 is always in the interval $\left[17\cdot 10^6, 28\cdot 10^6\right]$ m.
3. The difference $p_2 - p_1$ is in the interval $\left[-15, 115\right]$ m.
4. The difference $p_{C/A} - p_1$ is in the interval $\left[-32, 32\right]$ m.

Theoretically, the difference between p_1 and p_2 should be positive, since the L2 ionospheric delay is larger than that of L1. However, due to receiver and/or satellite hardware delays, this difference may be negative as well.
Denoting the compressed quantities by the subscript NCBI, and rounding to the nearest integer by [.], the compression algorithm is given as:

$$p_{1,NCBI}(t) = \left[\left(p_1(t) - 17\cdot 10^6\right)\cdot 10^3\right] \text{ mm}$$

$$\phi_{1,NCBI}(t) = \left[\left\{\left(\phi_1(t) - \frac{p_{1,NCBI}(t)\cdot 10^{-3} + 17\cdot 10^6}{\lambda_1}\right) - \left[\phi_1(t_0) - \frac{p_1(t_0)}{\lambda_1}\right]\right\}\cdot 10^3\right] \text{ mcycles}$$

$$\phi_{2,NCBI}(t) = \left[\left\{\left(\phi_2(t) - \frac{p_{1,NCBI}(t)\cdot 10^{-3} + 17\cdot 10^6}{\lambda_2}\right) - \left[\phi_2(t_0) - \frac{p_1(t_0)}{\lambda_2}\right]\right\}\cdot 10^3\right] \text{ mcycles}$$

$$p_{2,NCBI}(t) = \left[\left\{p_2(t) - \left(p_{1,NCBI}(t)\cdot 10^{-3} + 17\cdot 10^6\right) - 50\right\}\cdot 10^3\right] \text{ mm}$$

$$p_{C/A,NCBI}(t) = \left[\left\{p_{C/A}(t) - \left(p_{1,NCBI}(t)\cdot 10^{-3} + 17\cdot 10^6\right)\right\}\cdot 10^3\right] \text{ mm}$$

$$\phi_{C/A,NCBI}(t) = \left[\left\{\phi_{C/A}(t) - \phi_{1,NCBI}(t)\cdot 10^{-3} - \left[\phi_{C/A}(t_0) - \left(\phi_1(t_0) - \frac{p_1(t_0)}{\lambda_1}\right)\right]\right\}\cdot 10^3\right] \text{ mcycles}$$

An offset of 50 m is subtracted from the difference between p_1 and p_2 resulting in a quantity which is in a symmetric interval around zero, i.e., in the interval $\left[-65\cdot 10^3, 65\cdot 10^3\right]$ mm.
Note that the compressed carrier observations contain an additional integer bias term to avoid overflows. This bias will be lumped with the original carrier ambiguity and is therefore not required when decompressing the observations. Based on the above assumptions, the bit requirements for the NCBI code observations are 34 bits (no sign bit) for $p_{1,NCBI}(t)$, and 17 bits (MSB is sign bit) for $p_{2,NCBI}(t)$ and $p_{C/A,NCBI}(t)$ each. The time varying effects in $\phi_{1,NCBI}(t)$ and $\phi_{2,NCBI}(t)$ are due to the opposite sign of the ionospheric effects in code and carrier measurements. The bit requirements were therefore set to 21 bits (MSB is sign bit) for each. Since $\phi_{C/A,NCBI}(t)$ is the difference between $\phi_{C/A}$ and ϕ_1, all time-dependent effects cancel and the difference can always be brought into the range ± 0.5 cycles. The bit requirements for $\phi_{C/A,NCBI}(t)$ becomes 10 bits (MSB is sign bit).

The compression algorithm for the doppler observations is given as:

$$D_{1,NCBI} = \left[D_1\cdot 10^3\right] \text{ mHz}$$
$$D_{2,NCBI} = \left[\left\{D_2 - \left(f_2/f_1\right)\cdot D_{1,NCBI}\cdot 10^{-3}\right\}\cdot 10^3\right] \text{ mHz}$$

Under the assumptions

1. Required resolution: 1 mHz

2. The doppler observation D_1 is always in the interval $\left[-5.5\cdot10^3, 5.5\cdot10^3\right]$ Hz
3. The difference $D_2-\left(f_2/f_1\right)\cdot D_1$ is in the interval $\left[-2, 2\right]$ Hz.

the bit requirements for the NCBI Doppler observables are 24 bits (MSB is sign bit) for $D_{1,NCBI}$ and 12 bits (MSB is sign bit) for $D_{2,NCBI}$.

The SNR of observations is expressed in dbHz. The format for the SNR observations is based on the following resolutions:

1. $\max(9,SNR)\le SNR\le 21$: resolution 0.3 dBHz
2. $21< SNR\le 51$: resolution 0.2 dBHz
3. $51\le SNR\le \min(70.5,SNR)$: resolution 0.3 dBHz

With this quantization scheme, it is possible to store the SNR values in a single byte. The compression algorithm for all three SNR observations is given as:

if $SNR\le 21$ then
$$SNR_{NCBI}=\left[\left(\max(9,SNR)-9\right)/0.3\right]$$
else if $SNR\le 51$ then
$$SNR_{NCBI}=\left[\left(SNR-21\right)/0.2\right]+40$$
else
$$SNR_{NCBI}=\left[\left(\min(70.5,SNR)-51\right)/0.3\right]+190$$

Based on the above assumptions, the bit requirements for each SNR observable is 8 bits (no sign bit).

The terms $\left[\phi_1(t_0)-p_1(t_0)/\lambda_1\right]$, $\left[\phi_2(t_0)-p_1(t_0)/\lambda_2\right]$, $\left[\phi_{C/A}(t_0)-\left(\phi_1(t_0)-p_1(t_0)/\lambda_1\right)\right]$ may have to be re-computed due to data gaps, cycle slips, severe multipath, etc. The criteria for recomputing these terms are:

1. Change in receiver observation scenario (e.g., from cross correlating to non-cross correlating); all three terms are recomputed.
2. The interval between two consecutive observation epochs is too large; all three terms are recomputed.
3. The difference between $0.5\left\{\lambda_1\phi_1(t_k)-p_1(t_k)\right\}-0.5\left\{\lambda_1\phi_1(t_{k-1})-p_1(t_{k-1})\right\}$, i.e., the change in ionospheric effect between two consecutive epochs, is too large (this may be due to a cycle slip in ϕ_1); $\left[\phi_1(t_0)-p_1(t_0)/\lambda_1\right]$ is recomputed.
4. The difference between $\left(f_1^2/f_2^2+1\right)\left\{\lambda_2\phi_2(t_k)-p_1(t_k)\right\}-\left(f_1^2/f_2^2+1\right)\left\{\lambda_2\phi_2(t_{k-1})-p_1(t_{k-1})\right\}$ is too large (this may be due to a cycle slip in ϕ_2); $\left[\phi_2(t_0)-p_1(t_0)/\lambda_2\right]$ is recomputed.
5. The difference between $0.5\left\{\lambda_1\phi_{C/A}(t_k)-p_1(t_k)\right\}-0.5\left\{\lambda_1\phi_{C/A}(t_{k-1})-p_1(t_{k-1})\right\}$ is too large (this may be due to a cycle slip in $\phi_{C/A}$); $\left[\phi_{C/A}(t_0)-\left(\phi_1(t_0)-p_1(t_0)/\lambda_1\right)\right]$ is recomputed.

4. Conclusion

It should be kept in mind that binary dated may be stored in different ways. Big Endian (BE) systems place the most significant byte at the lowest memory address, whereas Little Endian (LE) systems place the least significant byte at the lowest memory address. The order of the bits within a byte is the same. Systems equipped with a Motorola processor are examples of BE systems, those with Intel processors of LE systems.

A small program to test the above compression algorithms, for code and carrier observations only, is available. This program is written in standard Ansi C and performs the compression algorithm on Rinex observation files. Another program, also in Ansi C, can be used to perform the decompression. Both programs were tested on BE and LE systems. It is possible to decompress files, compressed on a BE system, on an LE system and vice-versa: the software determines the way numbers are stored by itself.
The criteria to test if a re-initialisation has to be performed are based on the time difference between two consecutive observation epochs to a satellite and the change in ionospheric effect, computed using code and carrier, at consecutive epochs. These criteria may need further tuning.