Comparing Logged vs. Transmitted GPS Data

1. Introduction

The main components of the tracking collar are (1) a microcontroller, (2) a GPS receiver, (3) Flash memory, and (4) a GlobalStar satellite transmitter (or modem).

Position data (latitude and longitude) are sampled at a pre-programmed rate and transmitted to a GlobalStar satellite via the satellite transmitter. All transmitted data is stored into on-board Flash memory.

It has been noted that there are differences between the transmitted and logged positions for the same sample. The following explains this difference.

2. System Description

The data out of the GPS receiver is in NMEA ASCII format.

Latitude is in the format ddmm.mmmmm, where dd is degrees and mm.mmmmm is minutes and fractional minutes to five decimal places. Longitude is in the format dddmm.mmmmm, where ddd is degrees and mm.mmmmm is minutes and fractional minutes to five decimal places. For example, latitude and longitude at our facility is 3910.57610, N and 07715.15087, W.

The first step in the processing is an ASCII-to-Floating Point conversion, and then a conversion from degrees-minutes-fractional minutes to degrees-fractional degrees. That is, the minutes and fractional minutes are divided by 60 (60 minutes per degree) and added back to the decimal degrees. This floating point number is stored into flash memory and is the logged sample.

The next step is to convert the floating point number (degrees-fractional degrees) into a format compatible with the GlobalStar transmitter message structure.

GlobalStar represents latitude and longitude as a 24-bit signed binary number; that is one sign bit and 23 data bits.

A 23-bit number can be in the range 0 to 223 – 1, or 0 to 8388607 decimal. The number 8388607 represents 90 degrees latitude and 180 degrees longitude. The scale factors are computed as follows:

For latitude, 8388607 counts represent 90 degrees, so the scale factor is 93206.7 counts per degree. For longitude, 8388607 counts represent 180 degrees, so the scale factor is 46603.4 counts per degree. The latitude and longitude are converted to counts by multiplying degrees by the scale factor (counts per degree). The resulting floating point numbers are converted to signed-binary. These two 3-byte signed binary numbers (eight bits per byte) are packed into the transmitted message.

Note that as part of the conversion from floating point to binary, the original floating point number is rounded. For example, assume the NMEA latitude is 3910.57962, N. The floating point representation in degrees-fractional degrees is 39.1763267; the result of the multiplication by the scale factor (93206.7) is 3651496.13, and the signed binary number is decimal equivalent 3651496.

At the GlobalStar site, the signed binary number is converted back to latitude/longitude by dividing by the scale factor. The problem is that the original number has been rounded and the part of the number to the right of the decimal point is lost.

The number to the right of the decimal point can be in the range .0000 to .9999. This represents an error of 0 to .0000107 (.9999/93206.7) degrees in latitude and 0 to .0000215 (.9999/46603.4) degrees in longitude.

The difference in latitude is +.0000001 to -.0000078; the range of the difference between transmitted and logged latitude is .0000088, which is close to the expected .0000107.

The difference in longitude is +.000012 to -.000008; the magnitude of the difference is .00002, which is close to the expected .0000215.

The data agrees with the firmware computations. Two things to note:

(1) We would expect the rounding error to be symmetrical; that is latitude ±.0000054 and longitude ±.0000107. This was not the case; the lack of symmetry is probably due to slight errors in the other conversion routines.

(2) The difference in location is very small. The worst case difference in longitude is .00002 degrees. At the equator one degree of longitude is 60 nautical miles, or about 364,320 feet; .0000215 degrees is equivalent to seven feet.

In summary, we looked at the code and found why there is a difference between the locations being logged and transmitted. We ran the program on the emulator and put in a break point just in front of the code where the data is logged and transmitted. We manually changed the latitude and longitude to the logged values then looked at what was transmitted.

We were able to duplicate the transmitted and logged numbers almost exactly.