New Folder Name Proposed Format

A PROPOSED NEW FORMAT FOR THE GLASGOW 100–HOUR RUN DATA TAPES

B.F. Schutz, D.Nicholson, J.R. Shuttleworth & T.R. Barnett

Oct 91

1 Introduction

One of the key motivations for the 100–hour run was that it should inform us about the logistics of acquisition, storage and analysis of large volumes of broad–band laser interferometric gravitational wave data. In Cardiff, our experiences with the Glasgow data have given us fresh insight into how one might re–format the original data tapes such that any future analysis of the data could be made both easier and a lot faster. Furthermore, the computational tools to perform this most effectively are now at hand and this initial exercise will be valuable preparation for the design and implementation of a more general data storage format which we are planning for the European interferometer project.

This document describes our current thinking on what information should be written to the new tapes, a format for storing this information, and an assessment of the storage requirements. It has arisen from a meeting held about one month ago by the authors to pool their ideas on these issues.

Section 2 summarises our analysis of the Glasgow data, enumerating some of the conclusions we have been able to draw that are pertinent to our belief that re–formatted tapes are desirable. Section 3 recalls briefly the type of data formats that were described by Shuttleworth in his recent document. We present a summary of the information contained on the original Glasgow tapes and outline the different types of data that we believe should be present on the new tapes. Finally, Section 4 summarises the total storage requirements for the new data sets.

Note that the strategy we describe here for re–formatting the Glasgow tapes follows through almost exactly for the Max-Planck tapes. Our attention is therefore restricted only to the Glasgow tapes in this document.

2 Analysis of the Glasgow Data

John Watkins' analysis software was responsible for performing several different tasks. First — and of central importance to our proposal for a revised tape format — the main interferometer signal and various housekeeping streams were extracted from the multiplexed channels on tape. Due to the complicated bit operations that were needed to achieve this, the production of de–multiplexed data amounted to a considerable overhead in the overall analysis. We shall examine how a new tape format can resolve this unnecessary problem in the next Section.

The remainder of Watkins' analysis was performed on groups of five blocks of data in each recursion of his program, primarily because this gave roughly 2^{15} secondary error points which is a convenient number for FFT's and for the subsequent matched filtering. In fact, there is a subtlety here which shall have to bear in mind when we design our new tape formats. Each group of 32768 points is several times longer than the longest filter one would imagine applying to the data. However, due to the wrap–around problem inherent in circular correlations, it is essential to overlap the groups such that any contaminated data gets replaced. Each group was therefore backspaced by one block.

Upon de–multiplexing the various data streams, they were analysed in the follow ways:

2.1 calibration

If, within the digital byte, the calibration flag had been set, a crude interpolative scheme was used to calibrate the secondary error point data in the Fourier domain. A more explicit description of this stage in the analysis is irrelevant to what we want to discuss here so further details are omitted.

2.2 pre–whitening the noise

This is an important step in the analysis which has its origin in the matched–filtering theorem. It is, in effect, a weighting procedure applied to the noise which optimises the signal–to–noise ratio in the correlation output when the matched filters are applied. Watkins' scheme was to subdivide the 16384 complex points in each transformed group into 128 narrow 'averaging' bandwidths with 128 complex points in each bandwidth. He then asserted that the noise was white in each bandwidth and computed a noise variance, $S_h(f)$. This procedure can be viewed as a crude interpolation across the full bandwidth of the detector. It is flawed when there is structure in the noise spectrum on the same scale as the averaging bandwidth. An improved approach would be to build up the noise spectra over a longer span of time. This has the advantage of allowing one to select a much narrower averaging bandwidth and yet have many data points inside the bandwidth. The latter is necessary to guard against an

unbiased mean and standard deviation, which could result if there were a small number of points and one or two of these were particularly discrepant.

2.3 noise amplitude statistics

Once back into the time–domain, Watkins compiled noise statistics for each group. His claim is that a simple parameter, measured from the histogram of amplitude statistics, gives a good indication of the detector's performance during the span of time corresponding to one group of data, *i.e* about 1.6 seconds. We dub this diagnostic the *Gaussian parameter*. It's precise definition need not be given here, but note that we do have good evidence for a strong correlation between the Gaussian parameter and all of the housekeeping streams that monitor the performance of the detector. Since this is a key point which we will return to later, a few diagrams are appended to this document which reinforce Watkins' claim. These plot the Gaussian parameter derived for each group of secondary error point data selected from tape C4 along with the rms value for a selection of the housekeeping streams.

2.4 housekeeping streams

The sheer volume of housekeeping data and its complexity to extract from the original multiplexed data, provided a significant barrier to writing an easy and fast program for the data analysis. After examining this housekeeping data in some detail our claim would be that the complete set of housekeeping information is not essential for a robust data analysis. In particular, the primary and secondary visibility data and the secondary feedback signal are rendered essentially redundant by the Gaussian parameter which seems to correlate remarkably well with these streams and therefore serves as a reliable indicator of whether or not the laser was locked onto a fringe.

2.5 time-series search for gravitational collapse events

The details of this are mostly irrelevant to the present document and so we omit details here.

2.6 coalescing binary search

We have already outlined the need for overlapping data sets to avoid the wrap–around problem due to circular correlations. Apart from this, the details of the cross–correlations that Watkins performed with a family of compact coalescing binary star templates need not concern us here.

3 New Tape Format

We propose to adhere to the main features of data format that Shuttleworth has described elsewhere. Data would be written into a file comprising of several *chunks*. Filemarks would separate each file and there would also be markers to delineate each chunk. It makes a lot of sense to write a number of files onto one exabyte tape rather than one big file, which is the situation with the present tapes. The reasons are the following. To an exabyte drive SCSI–interfaced to either a UNIX machine or our COMPAQ PC, block–skipping in a single file is performed by actually reading and then throwing away blocks until the desired block is reached. It is therefore an extremely slow process to extract a few blocks if they happen to lie near the end of a tape. If, on the other hand, one writes a number of files to tape and marks each one with a filemark, then the desired block can be located at a comparatively much quicker rate since the workstation–SCSI interface can perform a quick scan for filemarks. Given that an exabyte tape has a storage capacity of some 2.5 Gbytes, it seems reasonable to write 100 files of about 20 Mbytes each onto tape. This figure is a simple compromise between a file which is possibly too big to reside on a shared disk and one which is too small to perform any sort of reasonable analysis on.

3.1 contents of a chunk

Each chunk will be designated a header which identifies it as being from one of a number of standard types. The chunk then has a format that depends on its type, usually using further information provided in the header. The type definition of a chunk may allow it to contain further chunks. The types of chunks needed for the data format we are proposing here are varied, including sampled data streams, histograms, Fourier transforms, threshold–crossing event lists, statistical summaries of long data sets and so on. We now discuss more explicitly the contents of a chunk.

3.1.1 The secondary error point data

This is the main gravitational wave data from Channels 2 & 5 in the multiplexed streams. It was originally sampled at 20 kHz and occupies 12 bits. We propose to store this data on the new tapes as two byte integers. This will reduce the inefficient bit manipulations needed to convert it into a form suitable for further processing, at the expense of raising the storage allocation to 40 Kilobytes per second of data (40 Kbyps).

3.1.2 The secondary feedback data

This resides in Channel 1 of the multiplexed data and is a detector diagnostic stream that also contains some of the low–frequency gravitational wave response.

It was sampled at 10 kHz and occupies 12 bits. Although the secondary feedback signal correlates well with the Gaussian parameter, we believe that the low-frequency signal may be of sufficient interest for future types of analysis that we should not reduce its information content prior to inclusion on the new tapes. For ease of extraction, however, we propose to write the secondary feedback data to the new tapes as two byte integers which requires 20 Kbyps of storage.

3.1.3 Digital data

The digital byte contains essential information, such as minute markers and calibration flags. It is stored as the first byte of the 10 byte cycle of data on the original tapes. It will be stored unchanged on the new tapes occupying, therefore, 10 Kbyps.

3.1.4 Microphone data

This is located in Channel 0 on the original tapes. It occupies 8 bits only. It will be stored unchanged on the new tapes, for which we therefore need to allocate 10 Kbyps.

3.1.5 Seismometer data

This needs to be extracted from the multiplexed Channel 4 on the original tapes. It was sampled at $\frac{10}{6}$ kHz and occupies 12 bits. Again, to avoid unnecessary bit manipulations, we will write this information to the new tapes as two byte integers, requiring 3.33 Kbyps of storage.

3.1.6 Primary and secondary visibility data

We argued earlier that the Gaussian parameter was a good diagnostic of the behaviour of the detector. It effectively duplicates the rôle of these data and we therefore propose that for the new tapes their informational content be reduced to simple threshold indicators. This would entail keeping a record of the running mean and standard deviation of the signals together with a record of the value of the signal when it crosses some preset threshold and the precise location in the stream at which this threshold–crossing occurs. Although we have no exact numbers at hand for the storage allocation required for this summary information, it is clearly going to be minimal.

3.1.7 Processed data

Any type of data analysis will require FFT's at some stage and since the repetitive application of this algorithm on groups of data during our analysis was one of the main bottlenecks, we propose to record FFT's for each group of 32768 points. Although this will significantly speed up any future analysis of the data

it does mean that the storage requirements are increased considerably. There are two reasons for this. The first of these, which we have discussed earlier, concerns the need for backspacing of each group in order to compensate for the wrap–around problem that arises during correlation with the filters. Secondly, the FFT data must be stored as four byte real numbers. The upshot of this is that the FFT data require 100 Kbyps of storage. It may be of benefit to store the raw data prior to its FFT such that the two sets of data are contiguous on the new tapes. This would add an extra 10 Kbyps of storage due to backspacing.

The procedure that we described earlier for pre–whitening the noise spectrum argued for building up Fourier spectra over a certain time span and then averaging these in order to construct an estimate for the spectral density of noise, $S_h(f)$. In view of this, one might like to record an averaged power spectrum onto the new tapes, based on all groups of data that "pass" the Gaussian parameter test. This extra and potentially very useful information would require negligible storage.

Once the data has been transformed back into the time domain, there are further useful quantities that can be measured and it would seem to us of advantage to output these onto the new tapes, especially since their storage requirements are minimal. Firstly, a histogram of amplitude statistics would be valuable. This provides a quick look diagnostic of the performance of the detector from which one can assess some statistical attributes for the detector noise. One of these is the Gaussian parameter and this deserves to be recorded in its own right for the reasons that we have already described in an earlier Section. In addition to this parameter, a running mean and standard deviation of the gravitational wave data, both before and after the noise has been pre-whitened, should also be recorded.

4 Total Storage Requirements

We can now collate the figures that we have estimated in the last Section for the storage requirements of the various types of data that we would like to write onto the new tapes. Rounding up to allow for the small items that cannot be assigned a length yet, and including the extra 10 Kbyps mentioned in Section 3.1.6, the total storage requirement is about 200 Kbyps. Dividing this by 20 Mb gives 100 seconds of data which would be allocated to a single file. Conversely, the original data tapes were recorded at 10 bytes per sampling cycle of 10 KHz. These figures yield a total of 100 Kbyps. The new data will therefore require twice as much storage, *i.e* about 60 8mm exabyte tapes instead of 30. We believe that this increase in storage is more than offset by the ease and speed with which any further analysis of the Glasgow data will be able to proceed.

```
GlasgowFileWrapper  :=
                GW__FILE                8 character string, not nul terminated
                <size>                  4-byte integer
                [Description]
                [GlasgowDateChunk]
                [GlasgowSpanWrapper]
                [PowerSpectrumSampleSet]


GlasgowDateChunk    :=
                GW__DATE                8 character string, not nul terminated
                <size>                  4-byte integer
                <year>                  4-byte integer
                <month>                 4-byte integer
                <day>                   4-byte integer
                <hour>                  4-byte integer
                <minute>                4-byte integer
                <second>                8-byte real
                [Description]


GlasgowSpanWrapper  :=
                GW__SPAN                8 character string, not nul terminated
                <size>                  4-byte integer
                [GlasgowBlockWrapper]..  (repeated as often as necessary)


GlasgowBlockWrapper :=
                GW_BLOCK                8 character string, not nul terminated
                <size>                  4-byte integer
                [SecondaryFeedbackSampleSet]
                [DigitalByteSampleSet]
                [MicrophoneSampleSet]
                [SeismicSampleSet]
                [PrimaryErrorPointSampleSet]
                [PrimaryVisibilityStatisticSet]
                [SecondaryVisibilityStatisticSampleSet]
                [WirePushStatisticSet]
                [SecondaryErrorPointSampleSet]
                [FourierTransformSampleSet]
                [AmplitudeStatisticSampleSet]


SecondaryFeedbackSampleSet :=
                FEEDBACK                8 character string, not nul terminated
                <size>                  4-byte integer
                <frequency>             4-byte real
                <elements>              4-byte integer
                <array>                 2-byte array [elements]
                [Description]           (optional)


DigitalByteSampleSet :=
                DIGITAL_                8 character string, not nul terminated
                <size>                  4-byte integer
                <frequency>             4-byte real
                <elements>              4-byte integer
                <array>                 1-byte array [elements]
                [Description]           (optional)


MicrophoneSampleSet :=
                MICROPHN                8 character string, not nul terminated
                <size>                  4-byte integer
                <frequency>             4-byte real
                <elements>              4-byte integer
                <array>                 2-byte array [elements]
                [Description]           (optional)


SeismicSampleSet :=
                SEISMIC_                8 character string, not nul terminated
                <size>                  4-byte integer
                <frequency>             4-byte real
                <elements>              4-byte integer
                <array>                 2-byte array [elements]
                [Description]           (optional)


PrimaryErrorPointSampleSet :=
                PRIERROR                8 character string, not nul terminated
                <size>                  4-byte integer
                <running mean>          4-byte real
                <running rms>           4-byte real
                <group mean>            4-byte real
                <group rms>             4-byte real
                <frequency>             4-byte real
                <elements>              4-byte integer
                <array>                 2-byte array [elements]
                [Description]           (optional)


PrimaryVisibilityStatisticSet :=
                PRI_STAT                8 character string, not nul terminated
                <size>                  4-byte integer
                <running mean>          4-byte real
                <running rms>           4-byte real
                <group mean>            4-byte real
                <group rms>             4-byte real
                <maximum value>         2-byte integer
                <minimum value>         2-byte integer
                [Description]           (optional)


SecondaryVisibilityStatisticSet :=
                SEC_STAT                8 character string, not nul terminated
                <size>                  4-byte integer
                <running mean>          4-byte real
                <running rms>           4-byte real
                <group mean>            4-byte real
                <group rms>             4-byte real
                <maximum value>         2-byte integer
                <minimum value>         2-byte integer
                [Description]           (optional)


WirePushStatisticSet :=
                WIREPUSH                8 character string, not nul terminated
                <size>                  4-byte integer
                <running mean>          4-byte real
                <running rms>           4-byte real
                <group mean>            4-byte real
                <group rms>             4-byte real
                <maximum value>         2-byte integer
                <minimum value>         2-byte integer
                [Description]           (optional)
```

```
SecondaryErrorPointSampleSet :=
                SECERROR                8 character string, not nul terminated
                <size>                  4-byte integer
                <frequency>             4-byte real
                <elements>              4-byte integer
                <array>                 2-byte array [elements]
                [Description]           (optional)


FourierTransformSampleSet :=
                FOURTRAN                8 character string, not nul terminated
                <size>                  4-byte integer
                <frequency>             4-byte real
                <elements>              4-byte integer
                <array>                 4-byte real array [elements]
                [Description]           (optional)


AmplitudeStatisticSampleSet :=
                AMP_STAT                8 character string, not nul terminated
                <size>                  4-byte integer
                <running mean>          4-byte real
                <running rms>           4-byte real
                <group mean>            4-byte real
                <group rms>             4-byte real
                <Gaussian parameter>    4-byte real
                <frequency>             4-byte real
                <elements>              4-byte integer
                <array>                 4-byte real array [elements]
                [Description]           (optional)


PowerSpectrumSampleSet :=
                PWR_SPEC                8 character string, not nul terminated
                <size>                  4-byte integer
                <transform count>       2-byte integer
                <frequency>             4-byte real
                <elements>              4-byte integer
                <array>                 4-byte real array [elements]
                [Description]           (optional)


Description :=
                DESCRIPT                8 character string, not nul terminated
                <size>                  4-byte integer
                <text>                  N-character string, not nul terminated
```