More Observability for Less Bandwidth …Where's the Trick?

José-Antonio Martínez-Heras[1]
Black Hat S.L., Córdoba, 14002, Spain

Tiago Francisco[2]
European Space Agency, Darmstadt, 64293, Germany

Alessandro Donati[3]
European Space Agency, Darmstadt, 64293, Germany

Usually, telemetry data is sampled at regular intervals. If better observability is required for certain parameters, they are sampled more often. This is not always possible due to bandwidth limitations between space and ground. A common practice to respect this limitation consists of reducing the sampling rate of other parameters so that certain parameters of interest can be monitored more closely. In this way it seems that it is not possible to increase the observability of some parameters without decreasing the observability of others. In order to provide more observability for the same or reduced bandwidth, the Fractal Resampling technique has been researched and developed at ESOC. The fractal resampling takes the original time series data samples and produces a set of fewer samples, not necessarily at regular intervals, that resembles the original time series while offering a configurable maximum error guarantee. With the proposed fractal resampling technique, it is possible to sample data at higher rates on-board and downlink only the data samples necessary to reconstruct the original signal with the desired level of accuracy with important data reductions. The fractal resampling not only enables more observability, but it also reduces bandwidth requirements. It has been applied to all housekeeping telemetry parameters of the ESA mission Rosetta; the results show that only 5.52% of the original data is needed if the 1% of the original amplitude is used as allowed maximum error.

I. Introduction

Usually, telemetry data is sampled at regular intervals. If better observability is required for certain parameters, they are sampled more often. This is not always possible due to bandwidth limitations between space and ground. A common practice to respect this limitation consists of reducing the sampling rate of other parameters so that certain parameters of interest can be monitored more closely. In this way it seems impossible to increase the observability of some parameters without decreasing the observability of others.

In order to provide more observability for the same or reduced bandwidth, the Fractal Resampling technique has been researched and developed at ESOC (European Space Operations Centre). The fractal resampling takes the original time series data samples and produces a set of fewer samples, not necessarily at regular intervals, that resembles the original time series while offering a maximum error guarantee. This means that all interesting information (e.g. peaks, short lived events) is never lost. This is a clear advantage with respect to under-sampling, which produces fewer samples by resampling data at lower regular rate at the risk of missing data peaks or other interesting information. The fractal resampling can also be considered a form of lossy compression. However, we prefer to call it "resampling" since, contrary to compression techniques, it does not need to be uncompressed.

[1] Software Engineer / Researcher, HSO-OSA, Robert Bosch Strasse 5, Darmstadt Germany.
[2] Software Engineer / Researcher, HSO-OSA, Robert Bosch Strasse 5, Darmstadt Germany.
[3] Advanced Mission Concept & Technology Manager, HSO-OSA, Robert Bosch Strasse 5, Darmstadt Germany.

The fractal resampling technique works by discarding data which is not interesting. If we are going to discard not-so-interesting data, we first have to define what we mean by interestingness. Our definition of interesting in this context is the data which is needed to take decisions. In this sense, if we need to plot a parameter in order to take an informed decision and we could take exactly the same decision using much less data, the data that is missing is not interesting.

We have developed a re-sampling technique inspired by the way 3D terrain is built in videogames. Usually, 3D terrain (e.g. mountains) are generated using fractals[1]. We realize that a time series could be considered to be a mountain silhouette (a 2D mountain). So the same principle was applied to a time series, but instead of using random values, the values from an already existing time series were used, not adding points that weren't needed. This created a similar time series, but with fewer points. In order words, we use a fractal inspired resampling technique that takes a time series (e.g. parameter behavior over a period of time) and removes the samples that are not strictly needed so that the same time series can be represented with fewer data points with the desired level of accuracy. Figure 1 shows provides with a graphical example of applying the proposed fractal resampling to the readings of a thruster temperature. According to Ref. 2 this method can be described as a top-down algorithm. There are several "top down solutions" which are used in the most varied fields. However the majority of these solutions try to recreate the same time series using as few samples as possible; our approach is different as it tries to do the resampling as fast as possible while using the minimum amount resources as possible. A possible on-board implementation has driven this decision.

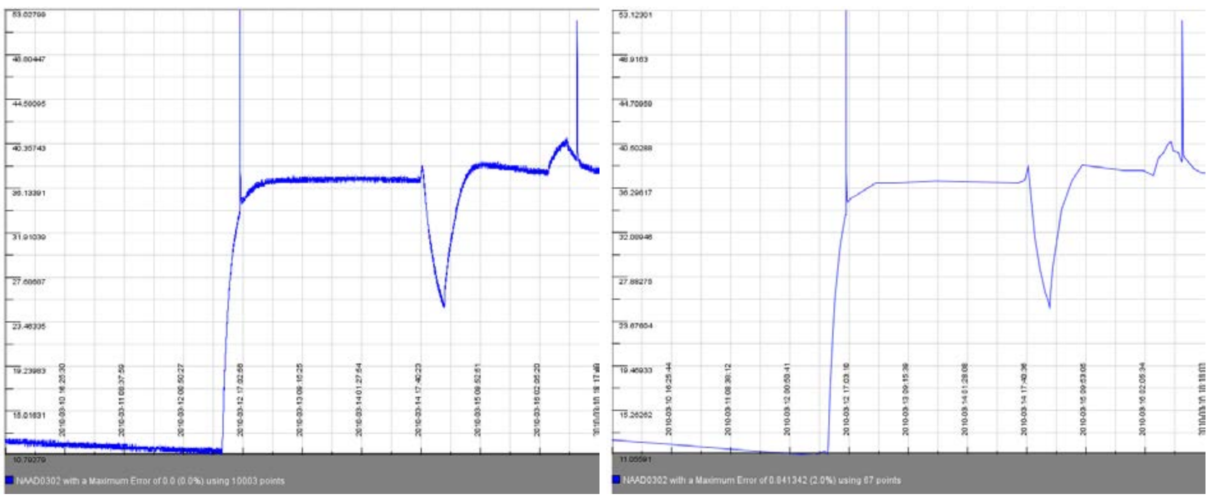

Figure 1. The effect of the fractal resampling: same operational decisions can be taken with much less data. *The two figures show the behavior of the parameter NAAD0302 (a thruster temperature) from the Rosetta ESA mission over a period of 6 days. The figure on the left consists of the original data as it comes from telemetry using 10003 data samples. The figure on the right consists of the resampled data using the proposed fractal resampling technique with a maximum error guarantee of 0.84 degrees Celsius, which results in 67 data points. Clearly, either of these two plots can be used to take operational decisions and, in this sense, the carry the same interesting information. The only different is that the fractal resampled version uses fewer data points (just 0.67% from the original).*

II. Technology

The proposed fractal resampling technique has two inputs: the original time series ([time, value] pairs) and the maximum allowed error. Its output is a new time series ([time, value] pairs) with fewer or equal number of data points as the input time series, possibly at irregular time intervals where the maximum specified error is guaranteed. This technique uses midpoint displacement in order to compress the original time series. The process is summarized in the following steps:

1. The first and last samples of the original time series are included in the list of displaced points.
2. Linear interpolation is assumed between the start and end samples determined in step 1.
3. For every point in the original time series that corresponds to the current segment, the absolute error

between its actual value and the corresponding linearly interpolated one is determined.

a. If this error is equals or above the maximum allowed error (epsilon) this segment needs to be displaced. The displacement consists of adding the middle point to the list of displaced points and applying step 2 to both the left and right side of the displacement (e.g. left = (start, displaced point), right = (displaced point, end))
b. If the error is lower than the maximum allowed error (epsilon) no displacement is needed

4. The results consists of the list of displaced points ordered by time

Figure 2 shows a graphical representation of this process.

This fractal resampling algorithm can be implemented both recursively or iteratively. The two versions have been prototyped yielding the same results in terms of compression. The iterative version is superior to the recursive one in terms of performance. We refer to the iterative implementation for the rest of the document.

We would like to highlight an important performance improvement that was implemented in step 3. As soon as the measured absolute error of any of the data points under examination is above the maximum allowed error it is know that a displacement should occur. Therefore, the rest of the samples do not need to be checked. This early stopping mechanism translates in an important performance improvement as the number of linear interpolation performed is significantly reduced. This makes the proposed resampling technique superior to other top-down alternatives[2], even if it produces suboptimal results[4]. The benefit of being able to run this resampling technique on-board the spacecraft is elaborated in section III.

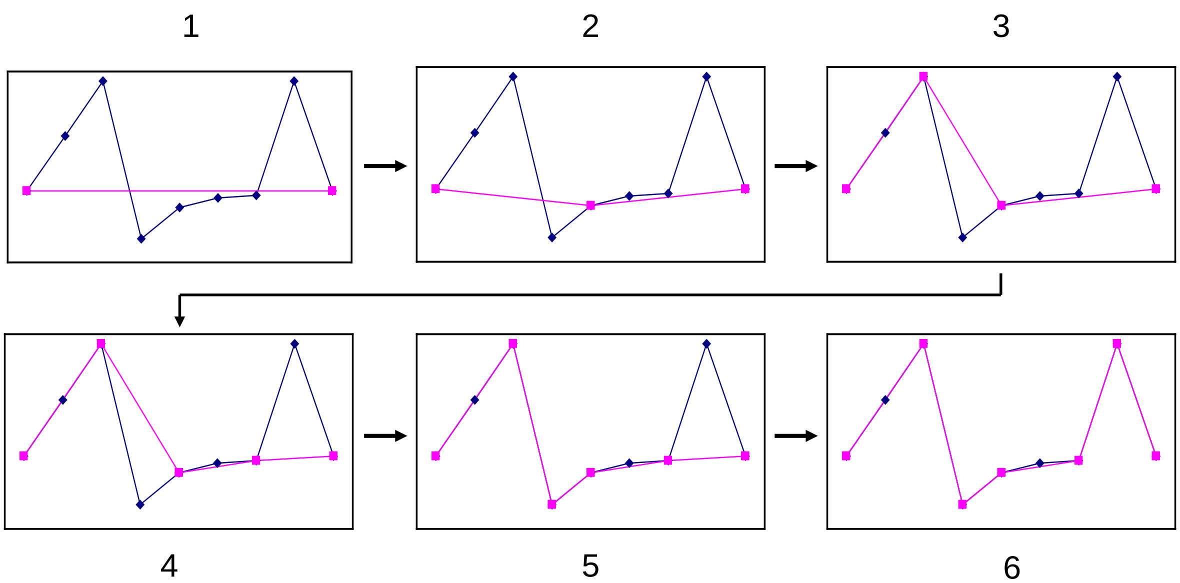

Figure 2. Fractal Inspired Resampling Algorithm. *Graphical representation of the proposed fractal resampling technique. The original time series is blue and the pink time series represents fractal resampled time series. At every step, the middle point of every segment is added to the resampled time series if the maximum error guarantee cannot be satisfied.*

III. Better Observability by means of high sampling rate on-board before fractal resampling

In order to capture short-lived events parameters readings should be sampled on-board at higher rates. Then. The fractal resampling technique could be applied before send it to ground. This would allow obtaining better observability on-ground with much less samples. A side benefit is that the bandwidth requirements are further reduced. It might appear that to follow this approach, a bigger storage is needed on-board to keep the high-rate

[4] Other resampling techniques find the data point that maximizes the error and include this point in the resampled version. The advantage of this approach is that the final resampled time series has fewer data points and it possibly resembles a bit better the original time series. The big disadvantage is that every data point needs to be checked.

sampled data. However, we could also think about applying the fractal resampling on parameter not at transmission level but at SSMM (solid state mass memory) storage level. This way the amount of memory needed, as well as the bandwidth requirements, are reduced.

In order to provide a graphical example, we have compared the fractal-inspired resampling resulting data with the original data and the original data sampled at half its original rate. The parameter chosen for this experiment is NPWD1604 (relatively difficult to compress) from the Rosetta ESA mission. As can be observed in Figure 3, the fractal-inspired resampling has proven superior to down-sampling as it successfully represents the original time series with about the same number of samples with much higher fidelity. The fractal-inspired resampling avoids the problem of missing key events when using fewer samples.

One may ask if this is true for all parameters or only for a small set. For this reason, a systematic characterization has been done for all Rosetta house-keeping telemetry parameters. This characterization covers 1 month of data and used 1% of the range of each parameter in this month as the maximum allowed error. Optimizations could be made to allow more error in less sensitive parameters but, for the sake of simplicity, the 1% was used for all. The compression in terms of resampled samples compared to original samples is 5.52% (original samples: 142841116, resampled samples: 7883797). This means that even if some parameters compress worse than others, overall, it is worthwhile to perform optimal resampling. Figure 4 shows that most parameters benefit enormously from the proposed resampling method. Notable exceptions are gyroscope readings, some currents and some voltages.

In order to use the proposed resampling technique, the time series, or part of it, should be available in a buffer. This means that data cannot be sent as soon as it is available since a certain number of data points are needed for the fractal resampling to be effective. Ideally, the buffer should be big enough to allow for good compression and also small enough to allow for on-board implementation and near real-time data transfers.

We have analyse the effect of the buffer size, in terms of numbers of samples that the buffer contains, for some typical telemetry parameters. For every parameter, it was computed the compression rate for buffers of sizes 5, 10, 15, 20, 25, 30, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100 samples for 4 different maximum guaranteed errors: 0.5%, 1%, 2%, 5%.

The level of compression varies depending on the parameter behaviour and on the accepted error. However, in all cases, the effect of the buffer size is the same. It is characterized by a negative exponential pattern that levels as the buffer size increases. It is remarkable that most of the compression gains are achieved with a relatively small buffer (e.g. 30 to 50 samples) as can be seen in Figure 5. This small buffer size has several advantages:

- It needs very limited on-board memory.
- Data can be sent very fast, i.e. if the buffer size is 30 samples we only need to wait the time needed to generate 30 samples before we can transmit them.
- Data can be store resampled on-board, releasing plenty of data storage resources.

IV. Other applications

- *Lossy Compression*: Finding the optimal sampling of a time series guaranteeing a maximum error results most of the times in an important reduction in the number of samples. Therefore in this sense, the ‘fractal resampling of time series’ can be used as lossy compression technique. The advantage as compared with other lossy techniques is that the maximum error can be guaranteed. Another advantage is that it does not need to be uncompressed. The output data is in the same format as the input data.
- *Storage and bandwidth reduction*: as fewer samples are required for the same information (within a given error) the requirements of both on-board and on-ground storage are reduced. In addition the bandwidth requirements of the space – ground and ground – ground channels are reduced.
- *Enabler for other applications*: some applications cannot work in a performing way if too many samples are in use. Typical examples include plotting and data mining. By having much less samples, this kind of analysis becomes feasible. The proposed fractal resampling technique is used in a new MUST[3,4] client to improve the performance of plotting in internet browsers[5].
- *Noise removal*: a side effect of using the fractal resampling is the removal of time series noise (e.g. signal noise). This happens if the noise is smaller than the maximum allowed error.

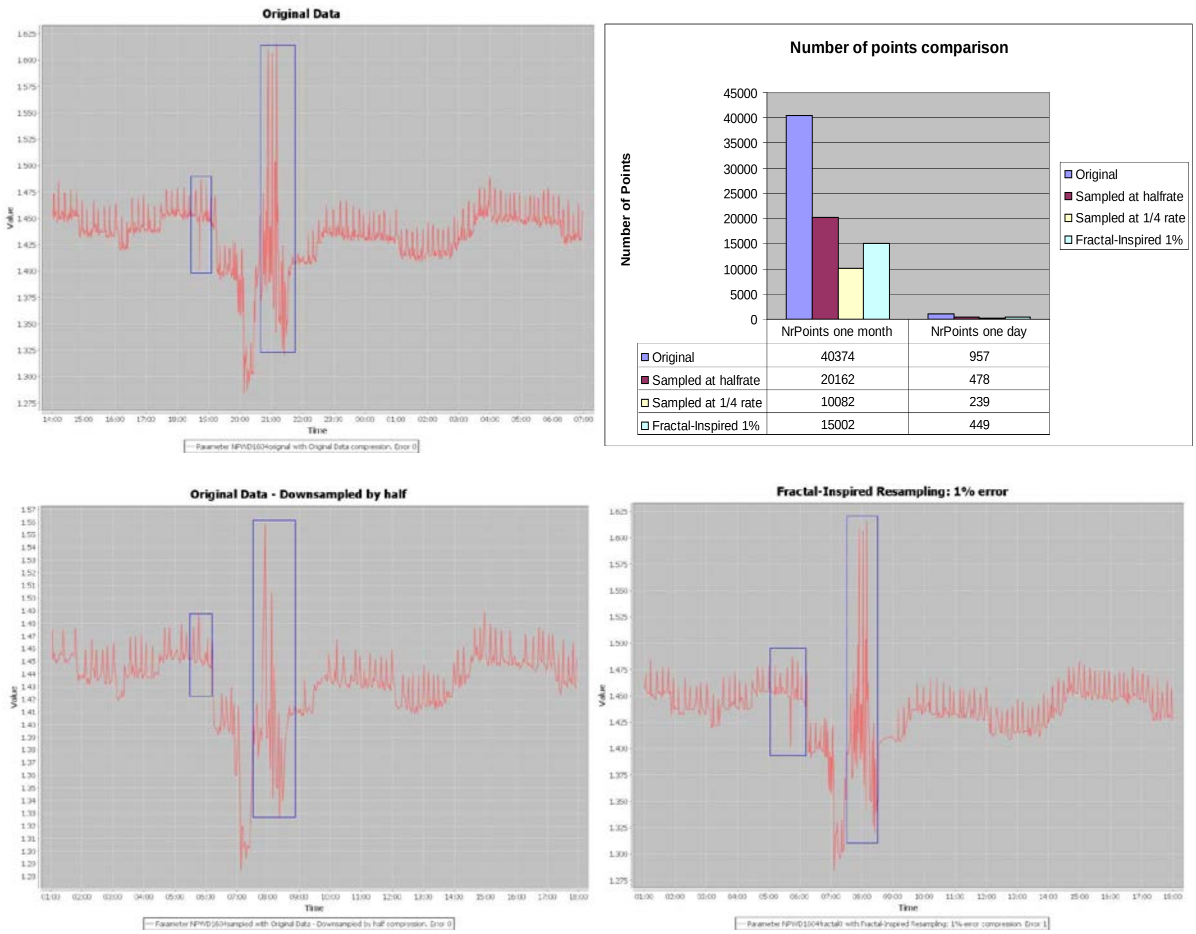

Figure 3. Rosetta Parameter NPDW1604 (relatively difficult to compress). *The top-left figure contains the original data. The bottom-left refers to the data down-sampled by half. The bottom-right refers to data resampled by using the proposed fractal resampling technique. In all plots, blue boxes have been used to highlight interesting peaks that are lost in the version down-sampled by half. These peaks are preserved in the fractal resampled version even if the number of resulting samples is equivalent. The figure top-right contains the details.*

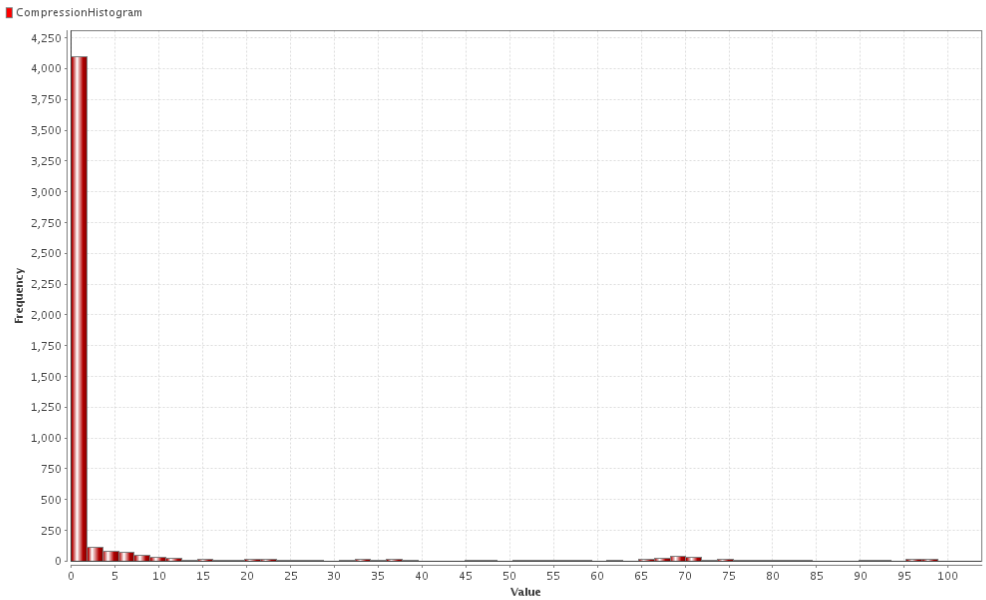

Figure 4. Compression Histogram: all Rosetta house-keeping parameters in an histogram view. *The horizontal axis represents the compression measured as the percentage of resampled samples with respect original samples when 1% of the original amplitude is used as maximum guaranteed error. The vertical axis represent the frequency (e.g. number of parameters with a given compression rate). It is clear from the graph that most of the parameters benefit greatly from the proposed optimal resampling technique.*

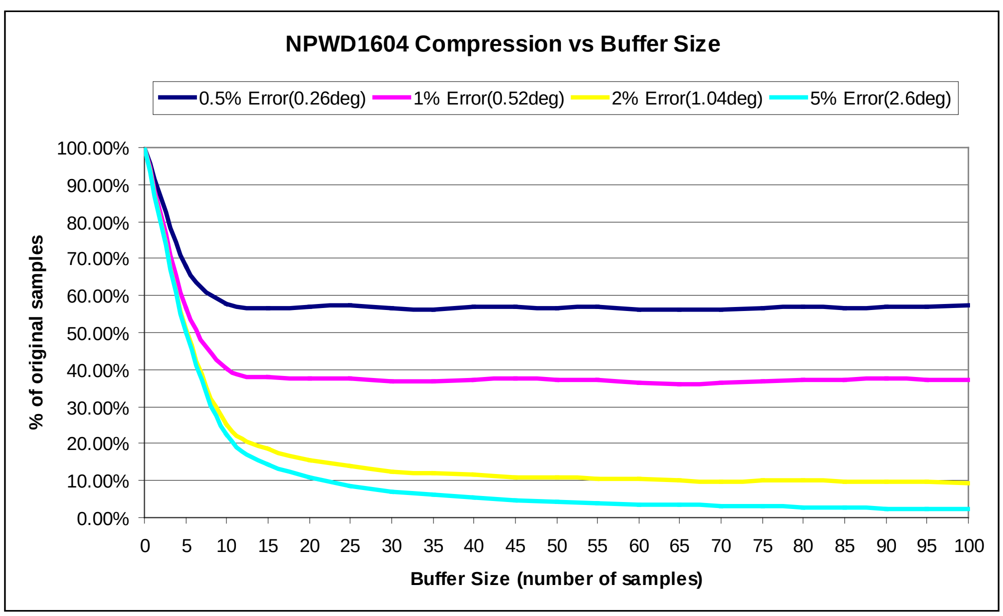

Figure 5. Compression versus Buffer Size. *The fractal inspired resampling needs to consider several data points at once to be effective. This can be practically done by using a buffer. This plot explores graphically how small this buffer can be without impacting the resampling performance. It can be observed that using a buffer of [30, 50] data points is the best idea since adding more data points to the buffer does not yield better compression performance. The same asymptotical pattern has been observed for other telemetry parameters.*

V. Compliance with CCSDS

The CCSDS house-keeping telemetry packets assume that the parameters traveling inside the packets are sampled at regular interval. All parameters in a packet share the time, which is specified in the packet header[5]. If the fractal resampling is used, the classic house-keeping telemetry packets cannot be built because, most probably, there will not be a value for every parameter at the same time. There are two possible solutions to overcome this limitation: use file-based systems or encapsulate [time, value] pairs in CCSDS packets.

A. Use file-based systems

A possibility is to transition from packet-based system to file-based system. In file-based system, information is stored in files rather than in packets. We propose to use a different file for every house-keeping parameter. This file will contain samples resulting from applying the fractal resampling technique to the parameter sampled on-board. This file can be then downlinked at the next communication possibility. Having different files for different parameters avoids the problem of having a sample at specific time for one parameter and not for another. This way, different time series are treated independently.

B. Data encapsulation in CCSDS packets

The resampled time series could be encapsulated in CCSDS packets. This solution is also used among the science community in order to transport science data in CCSDS packets. In this case, we propose to encapsulate the resampled samples in a CCSDS packet. The data structure could be the following: [CCSDS packet headers, data = [parameterID, [time1, value1], [time2, value2], [time3, value3] … [timeN, valueN]]].

VI. Conclusion

A method for finding the optimal sampling for time series inspired by fractals has been proposed. The main benefits are the reduction of the number of samples guaranteeing a given maximum error. This approach is superior to usual down-sample in the sense that the lossy compressed version resembles better the original behavior, usually with far fewer samples. Typical applications include better observability with reduced bandwidth requirements and lossy compression. While other similar solutions might exist, the proposed approach opens the possibility for an on-board implementation.

This technique might not be appropriate for all parameters (e.g. when hard real-time requirements exists). However, most parameters will benefit from the proposed fractal resampling technique. In practical terms, some modification would be needed to accommodate the fact that samples are not provided at regular intervals. In this sense, file-based systems will be more suitable. Another alternative would be to keep using packets and having parameters in pairs [time, value].

The accepted error must be defined for every parameter. We propose to use 1% of the range value of the each parameter as default. Then, engineers can change it for any parameter if not appropriate.

The described fractal-inspired resampling also allows fixing the number of resulting points. This results in a new time series with the given number of samples that best resembles the original one. In this case, the technique can estimate the resulting maximum error that would be made if an additional sample would be available.

While the focus of this work has been house-keeping telemetry observability, the proposed technique is generic and can be applied to any time series.

References

[1]Generating Random Fractal Terrain (http://gameprogrammer.com/fractal.html) – retrieved on 24th April 2012.

[2]Keogh, E., Chu, S., Hart, D. & Pazzani, M. (2001). An Online Algorithm for Segmenting Time Series. In *Proceedings of IEEE International Conference on Data Mining*. pp 289-296

[3]Martínez-Heras, J.; Baumgartner, A.; Donati, A.; “MUST: Mission Utility and Support Tools”, Proceedings DASIA 2005 conference, Edinburgh, UK.

[4]Baumgartner, A.; Martínez-Heras, J.; Donati, A.; Quintana, M.; “MUST – A Platform for Introducing Innovative Technologies in Operations”, Proceedings of the ISAIRAS 2005 conference, in Munich, Germany.

[5]Oliveira, H.; Lais, A.; Francisco, T.; Donati, A.; "Enabling visualization of large telemetry datasets"; Proceedings SpaceOps 2012 Conference, Stockholm, Sweden, June 11-15 (to be published).

[5] There is an exception though, the super-commutated parameters. The so called super-commutated parameters are parameters whose values appear several times within the same packet at different locations. Each location is to be interpreted as having a specified (defined in the mission database) delta time with respect the packet time stamp.