NORSAR

NORSAR Scientific Report No. 2-1999/2000

Semiannual Technical Summary

1 October 1999 - 31 March 2000

Frode Ringdal (ed.)

Kjeller, May 2000

6.5 Waveform quality analysis and data conditioning for the SPITS array

Introduction

We have for a long time noticed spikes in the data stream from the SPITS array (e.g., Schweitzer, 1998). In particular, the channel *SPB5_sz* has a severe problem. As shown in the lower trace of Fig. 6.5.1, the largest spikes are seen on the raw data trace, but after filtering more spikes become visible. This is illustrated in the middle trace of Fig. 6.5.1 where the data are differentiated twice.

We will in this contribution describe a new spike detector algorithm that identifies such one-sample spikes, without being sensitive to common seismic signals. The spike detector output is shown in the top trace of Fig. 6.5.1, where we observe high detector values for the spikes, but no effect of the strong signal occurring around 02.37. The data gaps at 02.40 are also handled well by the algorithm.

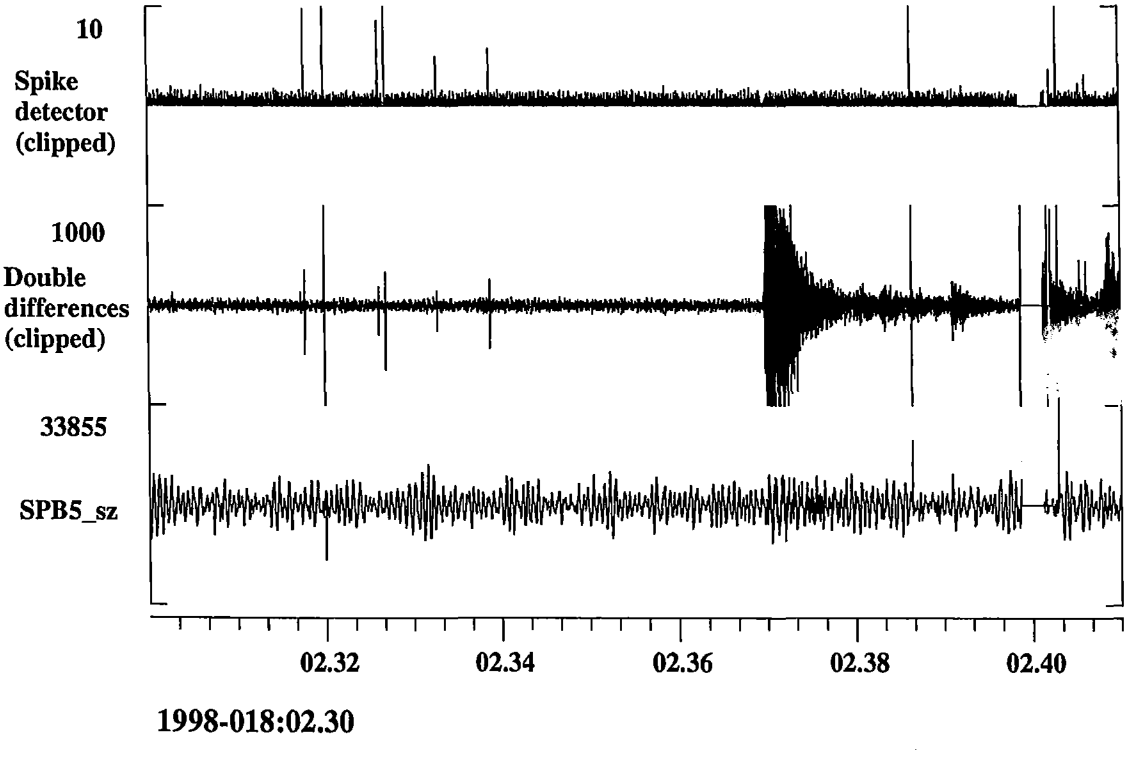

Fig. 6.5.1. *Illustration of data spikes at the SPITS array channel SPB5_sz (lower trace). In the middle trace the data are differentiated twice to accentuate the spikes, and the top trace shows the corresponding spike detector output.*

Intervals where the spike detector output exceeds a predefined threshold (typically 4) are declared as spikes, and the raw data are corrected using an interpolation scheme. The corrected data channel is shown in lower trace of Fig. 6.5.2, and the corresponding double differences and spike detector output are shown in the traces above. After correction there are no one-sample data spikes left. The spikes seen on the double differences of Fig. 6.5.2 are associated with the data gaps, for which the spike detector has little sensitivity.

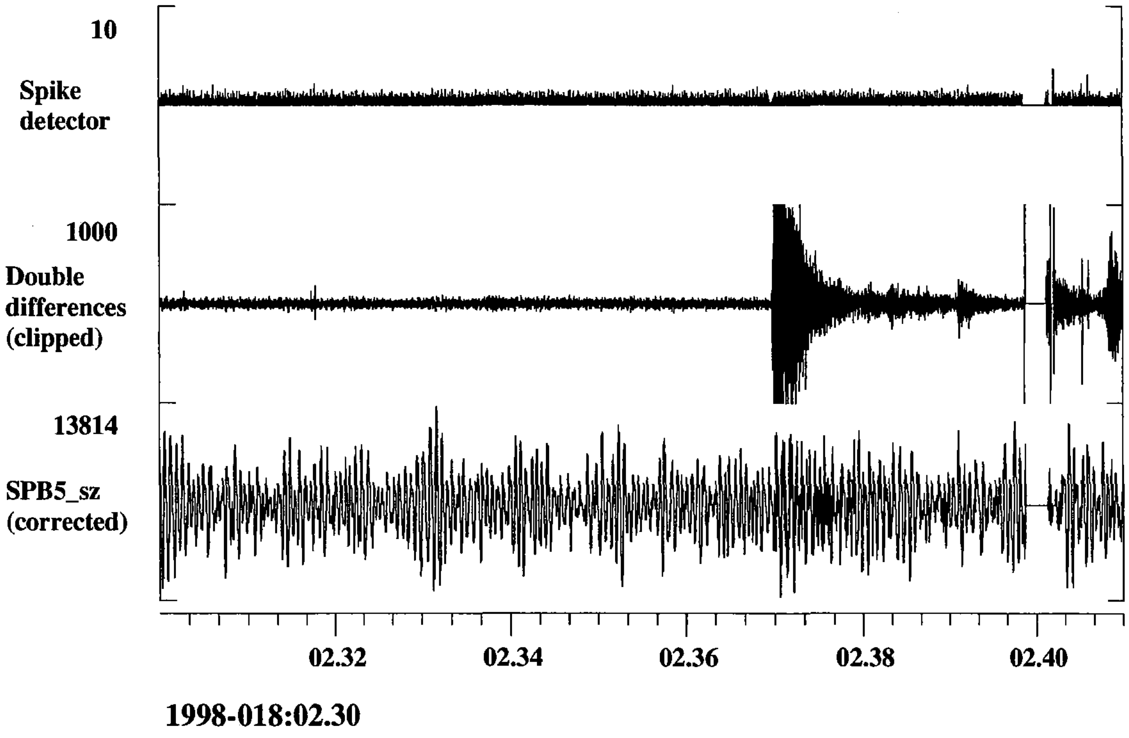

Fig. 6.5.2. *The lower trace shows the spike corrected version of channel SPB5_sz, with the original seen in the lower part of Fig. 6.5.1. The middle and upper traces are the corresponding high-pass filtered data and the spike detector output.*

The spike detector algorithm

In order to find a detector algorithm that is sensitive to spikes but insensitive to strong seismic signals, we have to utilize the features that separate these populations. Our assumption is that the energy associated with a spike is usually restricted to one (or a very few) samples, whereas seismic signals in almost all cases have energy with a duration of several samples. The developed spike detector procedure utilizing this difference is illustrated in Fig. 6.5.3, and we will in the following describe the different steps:

Step 1: *High-pass filter the data using double differences*

In order to increase the signal-to-noise ratio (SNR) of the spikes, a differential filter is applied twice to the data. This acts as a high-pass filter, and we see from trace no. 2 at the bottom of Fig. 6.5.3 that the SNR is increased and that the original one-sample spike is now transformed into three samples.

Step 2: *Calculate an LTA-like reference level for the spike detector*

The spike detector developed is a running procedure where we test each data sample against a reference level. From experiments we have found the maximum amplitude $max(abs(ddiff<t_w>))$ within a time window following each test sample to be very useful. The trace $ddiffmax(t) = max(abs(ddiff<t_w>))$ is number 3 from the bottom of Fig. 6.5.3.

Step 3: *Calculate the spike detector output*

As illustrated in trace nos. 2 and 3 from top of Fig. 6.5.3, the spike detector output for a given time t is calculated as $abs(ddiff(t)) / ddiffmax(t+\tau)$, where τ corresponds to the time shift of the reference level. To ensure that two- or three-sample spikes also fall outside the reference time window, the window start time is shifted about 0.2 seconds relative to t.

Step 4: *Define spike detections*

From testing of several data intervals we have found that a trigger threshold of 4 is a good initial value for defining spikes. For each spike detection interval, the time of the spike is taken to be the maximum.

Step 5: *Correct spikes using spline interpolation*

The spike in the raw data is replaced by a spline interpolated value of the data surrounding the initial spike, as shown in the top trace of Fig. 6.5.3.

Step 6: *Iterate the procedure*

The procedure can be applied iteratively to the spike corrected data. If spikes are still detected, we assume that the data contain other types of data problems, and we have the option to replace a longer time interval with spline interpolated data.

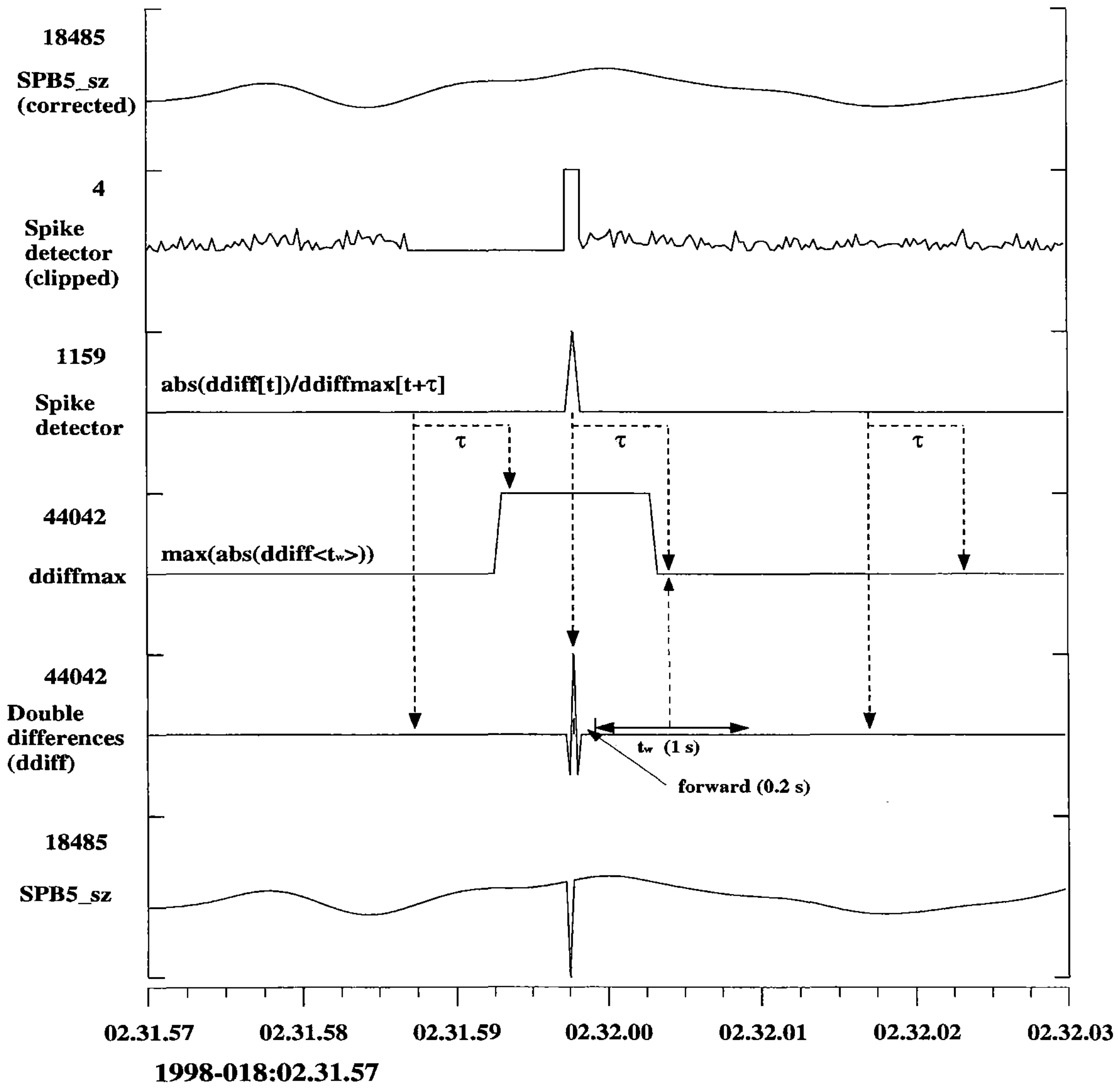

Fig. 6.5.3. *Figure illustrating the spike detector algorithm. See text for details. The numbers to the left of each trace give the amplitude maximum.*
The lower trace shows the raw data for SPITS channel SPB5_sz, including a clear spike.
Trace no. 2 from the bottom shows the data after applying a differential filter twice.
The reference level for the spike detector is shown in trace no. 3 from the bottom. This is calculated as $\underline{max(abs(ddiff<t_w>))}$ *where the maximum is taken within a window* $<t_w>$ *of typically 1 second.*
Trace no. 3 from the top shows the spike detector output calculated as $\underline{spidet(t) = abs(ddiff(t)) / ddiffmax(t+\tau)}$. *The arrows indicate the time shifts used in the numerator and denominator of the equation. In trace no. 2 from the top the spike detector output is plotted with a maximum amplitude of 4, which is the detection threshold currently used. The top trace shows the channel SPB5_sz after correction, using a spline interpolated value of the data surrounding the initial spike*

The example shown in Fig. 6.5.3 illustrates that the method works well for simple spikes. In Fig. 6.5.4 the method is applied to a high SNR P_n phase from an event located at a distance of approximately 250 km, and we can see that the spike detector has little sensitivity to such signals.

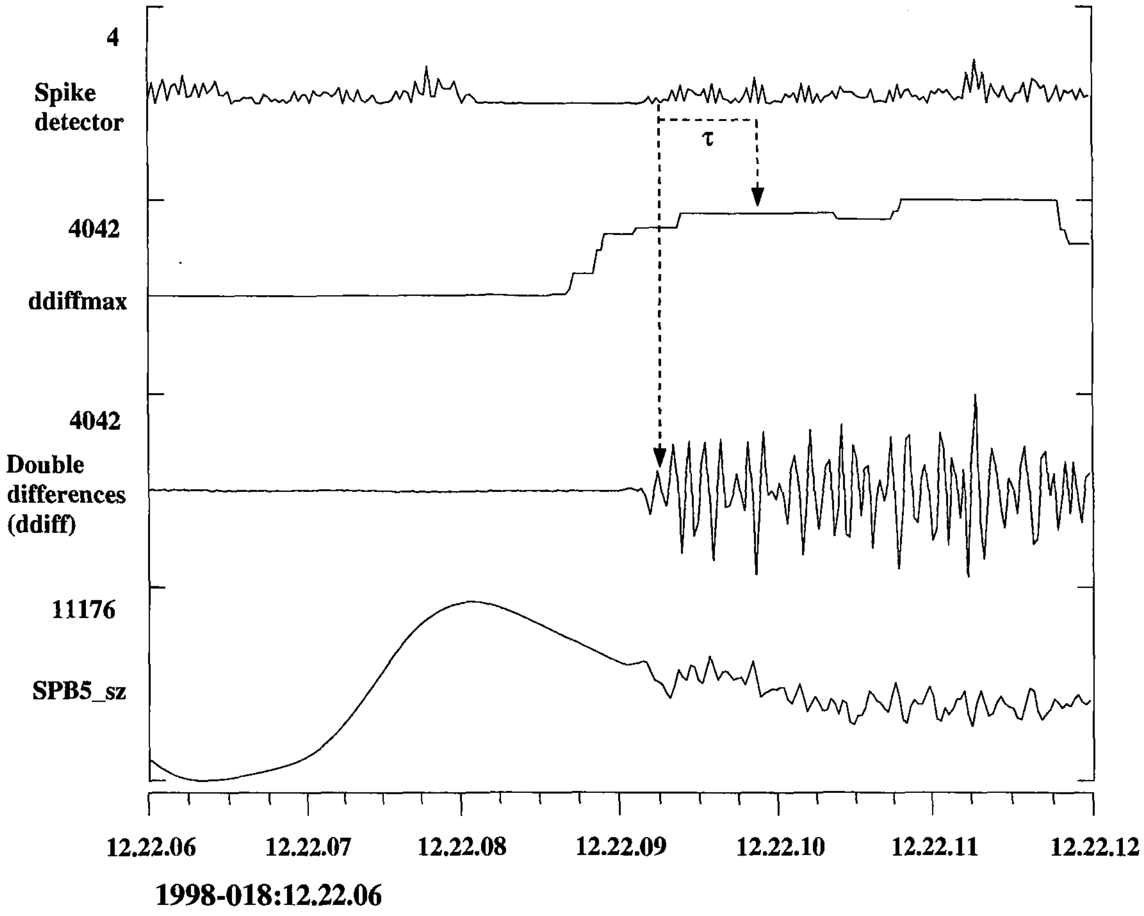

Fig. 6.5.4. *Figure illustrating the spike detector algorithm applied to a P_n phase from an event located approximately 250 km from the SPITS array. See caption of Fig. 6.5.3 for an explanation of the figure.*

For signals from very local events, having impulsive characteristics and of short duration, the spike detector has some sensitivity as illustrated in Fig. 6.5.5. However, a detection threshold of 4 in almost all cases prevents such signals being declared as spikes. In addition, we have introduced a test where amplitudes on both sides of the declared spike are compared. A large difference between these amplitudes strongly indicates that we have a signal, and the spike detection is consequently rejected. With this additional test, the spike detector has a very low false alarm rate.

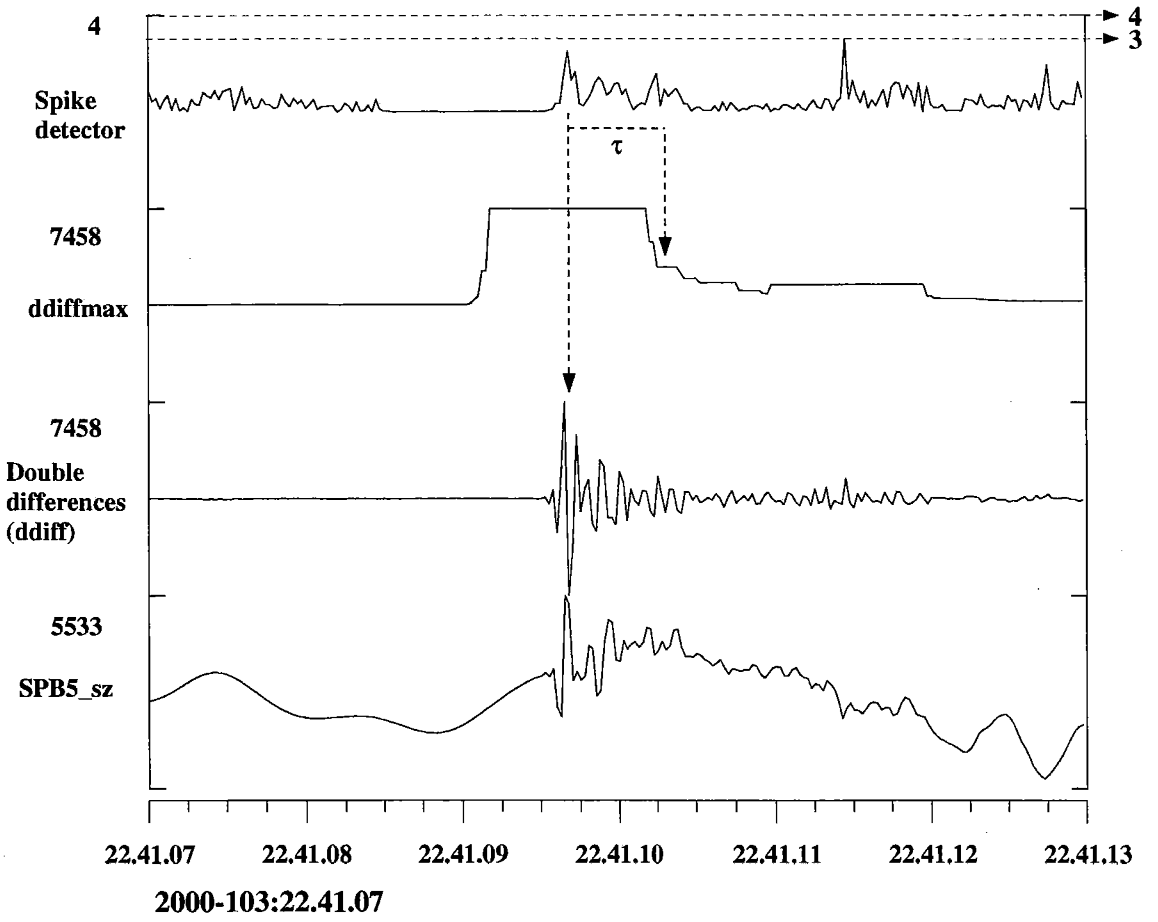

Fig. 6.5.5. *Figure illustrating the spike detector algorithm applied to an event located a few kilometers from the SPITS array. See caption of Fig. 6.5.3 for an explanation of the figure.*

Discussion

The spike detector and the corresponding data correction procedure provides an efficient method to remove well separated one-sample spikes from seismic data. In cases with more complicated data errors, like very frequent spikes possibly in combination with data gaps, the procedures also have some merit, in particular when applied iteratively. Even for such complicated cases, exemplified in the right most part of Figs. 6.5.1 and 6.5 2, the data quality is significantly improved.

Concerning the cause of the spikes in the SPITS data channel *SPB5_sz*, we are investigating a problem within the data acquisition system. However, our experience is that such data problems frequently occur on other types of systems, and that the developed procedure is of general use.

The procedure has been operationally tested at NORSAR, and has shown to significantly improve the processing quality of the SPITS array. However, a systematic study quantifying the performance of the method remains to be done. This is important for determining the optimum trigger threshold and the corresponding false alarm rate.

The algorithm is available in NORSARs processing system EP through the following two commands:

corrspike <n>

Run the spike detector on n data channels and correct the spikes. In addition the processing parameters can be adjusted using additional arguments to the command.

detspike

Run the spike detector on the top data channel on the data stack and correct the spikes. In addition, the high-pass filtered data (double differences), the detector reference levels, and spike detector output are put on top of the data stack, as shown in Fig. 6.5.5.

References

Schweitzer, J. (1998). Tuning the automatic data processing for the Spitsbergen array (SPITS). In: NORSAR Semiannual Tech. Summ. 1 April - 30 September 1998, NORSAR Sci. Rep. 1-98/99, Kjeller, Norway, 110-125.

T. Kværna