Getting started with acoustic well log data using the dlisio Python library on the Volve Data Village dataset

Erlend Magnus Viggen[1], Erlend Hårstad[2], Jørgen Kvalsvik[2]

[1] Centre for Innovative Ultrasound Solutions and Dept. of Circulation and Medical Imaging, Norwegian University of Science and Technology, Trondheim, Norway
[2] Software Innovation Bergen, Equinor ASA, Bergen, Norway
Contact email: erlend.viggen@ntnu.no

Abstract

Three issues have long impeded academic research and teaching on well logging. First, real measured data has been hard to come by. This has now been alleviated by Equinor's 2018 release of the Volve Data Village dataset. Among its 5 TB of data, it contains 16.3 GB of various well log data, plots, and analyses. Second, no free and effective software tools to programmatically read DLIS files, one of the most common file formats for well log data today and by far the most common format in the Volve Data Village, have been available. This has now been remedied by the free and open-source Python library dlisio, first released by Equinor in 2018 and still under heavy development. Third, the data is often difficult to understand, as sufficient documentation is often not publicly available. As different tools measure, process, and store their data differently, different tools must be understood individually. This article aims to stimulate research into well logging, by showing how to use dlisio to investigate well log data from the Volve Data Village dataset. While the investigative methods used here can be adapted to other kinds of data, this article focuses on acoustic integrity logs. Specifically, we investigate data from a sonic tool (DSLT) and an ultrasonic tool (USIT), both extensively used in the dataset. In addition to identifying what the most fundamental pieces of data represent, we also show some simple examples of how this data can be reprocessed to find new results not provided in the well log file. We provide the code underlying this article in an accompanying Jupyter Notebook.

1 Introduction

Among the man-made structures in use on Earth, few are more remote and inaccessible to humans than oil and gas wells. Not only are wellbores narrow holes stretching for kilometres below the ground or the seabed, they are also extreme environments with very high temperatures and pressures. Even so, it is often necessary to perform measurements along these wellbores to determine their status, for example to investigate the geological properties of the formation after drilling or to determine if a cementing operation was carried out successfully. Such measurements are typically performed by lowering well logging tools into the well. These tools can use any of a variety of modalities, making e.g. acoustic, electromagnetic, or mechanical measurements.

ISBN 978-82-8123-020-0

Perhaps the largest share of research into well logging is carried out by the service companies that develop commercial logging tools and the oil companies that use them. There is also research activity at universities and research institutes, but independent academic research or teaching on logging based on real-world data has long been difficult for several reasons:

1. Data is not easily available. Most data is owned by oil companies, which are typically quite reluctant to release their data to researchers. While some universities and research institutes may have access to national data repositories containing well log data, such as the DISKOS database in Norway [1], this option is not available to every researcher, and such repositories do not exist in every country. Even when such repositories are accessible, the data may come with a licence that places limits on the publication of work based on it.

2. When log data is available, it is very often provided as DLIS files (short for 'Digital Log Interchange Standard'), a binary file format standardised in 1991. DLIS is a very complicated and specialised format, and no free software to read DLIS data programmatically was available until very recently. Thus, even if you had DLIS files containing log data, actually extracting the data would not have been a straightforward task, and doing so in a convenient manner would require proprietary software.

3. After the data is extracted, the descriptions provided in the file of what the different pieces of data represent is often insufficient. Additionally, publicly available documentation to help understand it is scarce. This can lead to a situation where you are not sure what your data actually represents.

Fortunately, over the past few years these problems have been at least partly solved. The first problem was alleviated by Equinor's release of the Volve Data Village dataset in 2018 [2]. This is a free and open dataset that contains a multitude of log data, which we give further details on in Section 2.1. The second problem was solved by the recent release of dlisio, a free and open-source Python library that we cover in Section 2.3. The third problem must be dealt with separately for each specific tool, as different tools measure, process, and store their data in different ways. In this article, we focus on the sonic and ultrasonic tools that are primarily used in the well integrity logs in the Volve Data Village dataset. The main goal of these logs is to determine which intervals in the well may be hydraulically isolating, by establishing where in the well cement or formation material exists in the annulus between the casing and the formation, and what the quality of these materials are.

Throughout this rest of this article, Section 2 gives an overview of the dataset (Sec. 2.1), explains the DLIS file format (Sec. 2.2), and introduces the dlisio library (Sec. 2.3). Thus, Section 2 is relevant to everyone who wants to get started with well log data from DLIS files. Section 3 demonstrates how to read, understand, and perform some reprocessing of the most important sonic (Sec. 3.1) and ultrasonic (Sec. 3.2) log data from the Volve Data Village dataset. While Section 3 is most relevant to researchers working with such data, the same general approach, which Section 4 also summarises, may be adopted by researchers that want to investigate other kinds of well log data.

The work that we present in Section 3 is also mirrored in a Jupyter Notebook that provides the underlying code [3].

ISBN 978-82-8123-020-0

Table 1: Log data file formats with 20 or more files in the Volve Data Village

Format	Description	Type	No. of files
DLIS	Digital Log Interchange Standard	Binary	607
ASC	ASCII file *(not a standard format)*	Text	345
LAS	Log ASCII Standard	Text	301
SEGY	SEG-Y Data Exchange Format	Binary	221
LTI	Log Tape Image	Binary	36
DEX	Paleo Data Exchange Format	Text	24
LIS	Log Information Standard	Binary	20

Table 2: Overview of the well integrity log files in the Volve Data Village. Intervals are specified from their log's start depth to its end depth. Section 3 focuses on the highlighted log file.

Well name	Logging date	Casing OD [in]	Sonic tool	Ultrasonic tool	File no.	Pass type	Depth interval [mMD]	Length [m]
F-9	2009-06-26	$13\frac{3}{8}$	DSLT	USIT	1	Main	845.2–150.0	695.2
					2	Repeat	845.4–149.8	695.6
F-11 B	2013-06-05	$9\frac{5}{8}$	DSLT	USIT	1	Repeat	3183.8–2704.6	479.2
					2	**Main**	**3185.0–2474.8**	**710.2**
F-12	2016-08-18	$9\frac{5}{8}$	ASLT	USIT	1	Main	2960.1–2481.7	478.4
					2	Repeat	2960.4–2836.5	123.9
F-15 C	2013-10-09	7	DSLT	USIT	1	FPM	59.4–3047.3	2987.9
					2	Main	3046.4–2729.6	316.8
					3	Repeat	3045.3–2740.6	304.7

2 Data

2.1 The Volve Data Village dataset

In June 2018, Equinor released the Volve Data Village dataset for the purposes of research and study [2]. Its current license is based on Creative Commons Attribution 4.0 International licence, but with some additions to e.g. disallow resale of the dataset. It contains 5 TB of data over almost 40 000 files containing subsurface and production data from the Volve field. Among this data, we find seismic, reservoir, drilling, production, and well log data.

The dataset's 16.3 GB of well log data, plots, and analyses covers 24 wellbores. Primarily, it contains mud logs, logging-while-drilling results, pressure test results, petrophysical composites and interpretations, production logs, integrity logs, and biostratigraphic analyses. Table 1 lists the most common data formats in the dataset, showing that DLIS is by far the most common. While the text formats can be relatively easily parsed, binary formats such as DLIS are very difficult to read without software tailored to the purpose.

In this article, we will focus on the well integrity wireline logs. Table 2 gives an overview of the relevant well log files in the dataset. It shows that the most common sonic tool is the Digitizing Sonic Log Tool (DSLT) from Schlumberger [4], which we describe further in Section 3.1. It also shows that every integrity log uses the Ultrasonic Imager

ISBN 978-82-8123-020-0

Tool (USIT) from Schlumberger [5], which we describe further in Section 3.2. In total, the main log passes cover 2201 m of well. Including the repeat passes, which typically remeasure subsections of the main log, we have 3804 m of well log data available. In addition, we have a 2988 m long fluid properties measurement (FPM) pass, a special log pass to measure the physical properties of the fluid inside the casing.

2.2 DLIS files

DLIS is a binary file format for well logs, developed by Schlumberger in the late 80s and published by the American Petroleum Institute (API) in 1991 under the name Recommended Standard 66 v1 (RP66v1). It is now the Petroleum Open Standards Consortium (POSC) [6] that has the stewardship. RP66v1 enjoys extensive use in the oil and gas industry. API released a second version, RP66v2, in 1996, but it never really caught on. Today, the vast majority of DLIS files, including those in the Volve data set, are of version 1.

Compared to another very popular file format for well logs, the Log ASCII Standard (LAS), reading and understanding DLIS is quite the challenge. In addition to sampled data being stored in a binary representation rather than textual tables, the format itself is quite flexible and open for vendor-specific and per-file semantics. This makes large scale automatic processing of DLIS files difficult. DLIS files have no index, have no positional hints, and records are of variable size, which means finding and reading arbitrary records is impossible without a prior scan. Additionally, measured data is stored row-oriented, and integers can be variable-length. These properties make DLIS files smaller and quite flexible, but it also makes fast random access of data difficult. The versatility of the format makes it quite powerful, able to capture a wide range of data types and relationships that simpler formats are not able to represent. An example of this is image logs — DLIS has first-class support for multi-dimensional samples, which in LAS is often solved by having multiple columns and a file-specific, often unspecified, schema to relate them.

Having worked with other well log formats, the transition to DLIS can be a bit daunting. The following sections gives an overview of how DLIS files are structured and what kind of information you may expect to find in them.

2.2.1 Logical files

DLIS has a built-in mechanism for segmenting a single disk file into multiple separate logical files, as Figure 1 shows. DLIS ties no semantics to this segmentation, i.e. there is no difference between a set of logs stored in a single physical file, and the same logs stored in multiple physical files. Each file consists of a set of objects, represented as key-value pairs (dictionaries), which describe the file contents and structure. While most of these are designed to store a specific type of information, such as tool description, tool calibration, axis description, or measurement metadata, objects are open for vendor-specific extensions. There are over 20 different object types specified by the standard, although only a handful see widespread use. Objects can be defined in almost any order, with any data in between.

All objects come with a file-unique identifier, which other objects use to reference, relate, or add information. For example, borehole measurements are described by a Channel type object, which contains references to a Tool type object that describe the tool, its manufacturer, parts, and parameters.

ISBN 978-82-8123-020-0

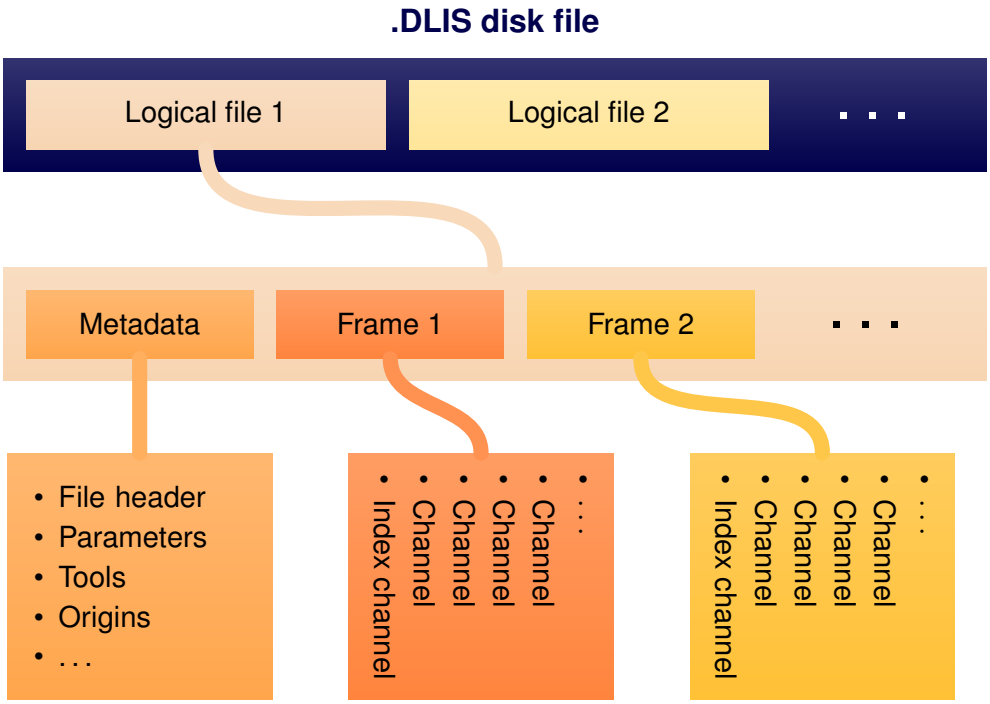

Figure 1: Simplified schematic overview of the contents of DLIS files

Table 3: A very short time-indexed Frame with 7 Channels, `TIME` being the index channel. `FRAMENO` is not a channel as such, but a numerical index of each sample in the frame.

FRAMENO	TIME	TDEP	ETIM	LMVL	UMVL	CFLA	OCD
1	16677259	852606.	0.	585	635	18	6789.05
2	16677659	852606.	0.4	585	635	18	6789.05
3	16678059	852606.	0.8	585	635	0	6789.05
4	16678459	852606.	1.2	585	635	0	6789.05
5	16678859	852606.	1.6	585	635	0	6789.05

2.2.2 Frames and channels

The lion's share of a DLIS file is typically made up of measurements taken along the wellbore. These are organised in objects called Frames, shown in Figure 1. Conceptually, a Frame can be seen as a table of data, like Table 3. The columns of this table are referred to as channels. Frames almost always have an index channel that provides the position in e.g. depth or time at which the rest of the values in the row were measured. Each Frame usually corresponds to a log run, but otherwise Frames impose little structure except grouping channels that have a common index. DLIS puts no restrictions on the number of channels per Frame, or the number of Frames in a file.

While the by far most common case is that each channel has scalar samples, i.e. a single measured numerical value per row, DLIS channels' samples can also be multi-dimensional arrays. For example, this is common for ultrasonic logs, where some channels contain a one-dimensional array of values per row, representing measurements made at different azimuthal angles.

The channels in a Frame are described by corresponding Channel objects, each with

ISBN 978-82-8123-020-0

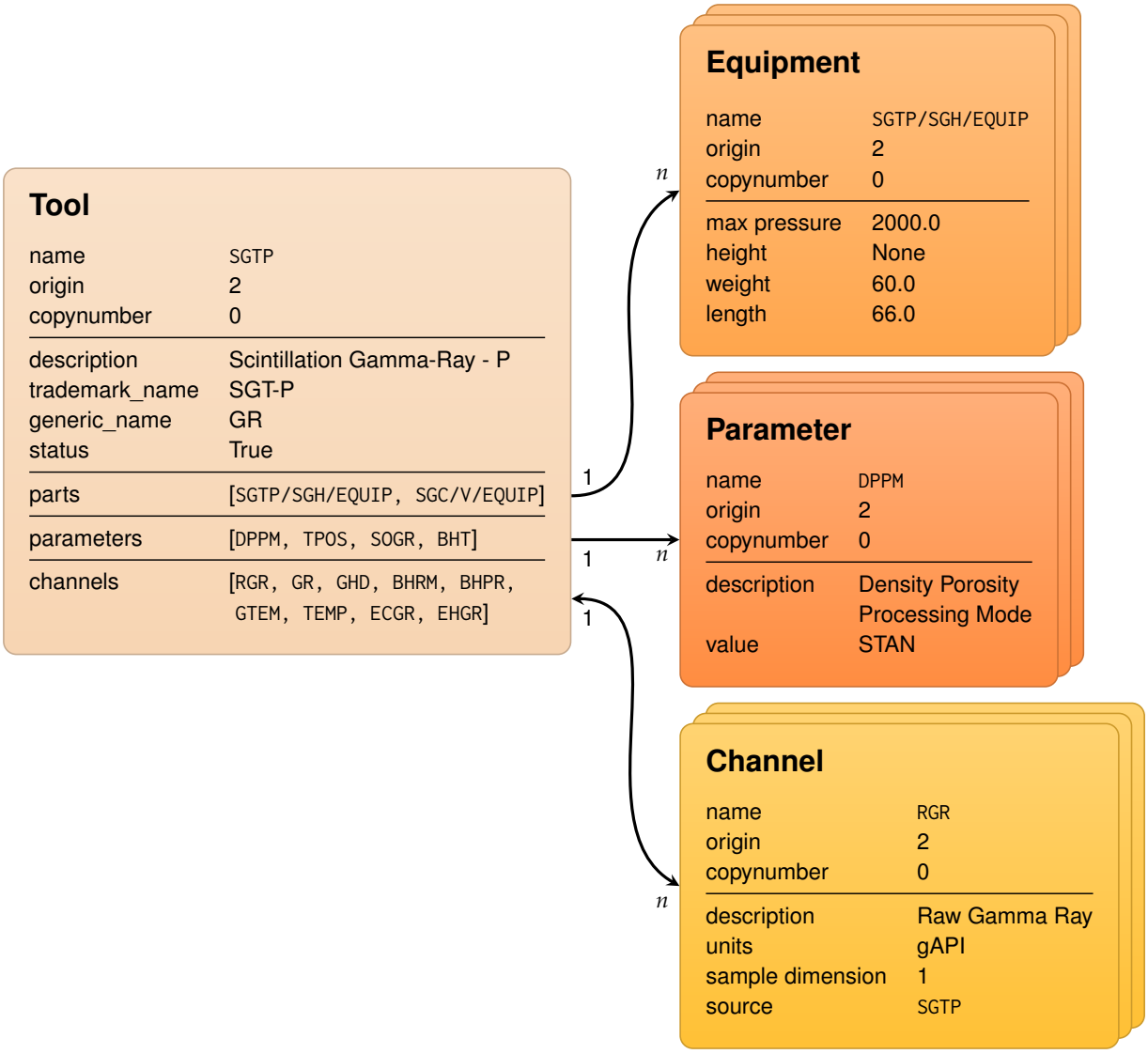

Figure 2: Schematic overview of different DLIS metadata objects and their relations

a name (often known as a 'mnemonic'[1]), a description (known as a 'long name'), unit information, the dimension (the shape and size of each sample), and various properties that vendors may choose to classify the Channel with.

2.2.3 Metadata

Together with the actual log data, there is often an abundance of metadata related to the acquisition of the logs. Typically only a few of the object types specified by the standard are present in a single file, as vendors are free to pick the types needed to represent their data. Additionally, vendors have the freedom to specify their own objects. However, with cryptic naming and minimal explanation, such objects can be challenging to decipher without any external explanation of the intent of these objects.

Each object type is tasked with specifying a concrete set of key-value pairs. Additionally they often rely on other objects to add more context. Take the Tool object, which describes a single tool. One of its attributes is a list of parameters, but rather than speci-

[1]The term 'mnemonic' is often used throughout the industry for the short names that identify Channels and other objects.

fying the parameter itself, it stores references to Parameter objects. In turn, the Parameter objects describe each parameter in more detail. This kind of referencing between objects, shown in Figure 2, is extensively used and is yet another example of how powerful DLIS is compared to other formats. However, an important thing to note is that these references often are one-directional. For the Tool–Parameter example, the Parameter itself has no information linking it to the Tool, as Figure 2 also shows. An important lesson from this is to be careful about looking at objects in isolation, as they are often a part of a bigger picture.

The remaining part of this section describes a few of the most commonly used metadata objects. Channel and Frames are left out as they have already been given their own section. For a complete list of all object types the reader is referred to the dlisio documentation [7].

Origin: General information about the file and the circumstances in which it was created is recorded in Origin objects. These contain information about which country, field and well the logs are from, who produced the file, who it was produced for, when and how it was produced.

Parameter: Parameter objects often provide important information about the acquisition and processing of channels, which can aid in the interpretation of the logs. These objects contains a textual description of the parameter at hand, together with the actual parameter values. Parameter objects may or may not be independent; while some are closely linked to other objects, such as tools, others define general information about the file.

Tool data: Files typically contain a lot of information related to the physical tools that were used in the acquisition of the logs. Tool objects are specifically designed for the purpose of describing a single tool. These offer general information about the tool, like its name and description. Often there are multiple names recorded, such as trademark and generic names. These might aid in finding the correct tool specification online. A physical tool is often made up by several parts. Each part is described by an Equipment object which contain serial numbers, physical dimensions and thresholds for maximum operating temperature and pressure. All Equipment that makes up the tool is referenced from the Tool object. Additionally, each tool has a record of all channels that were directly produced by it and of Parameters that relate to the tool.

2.3 The dlisio library

dlisio is a free, fast, and easy-to-use Python library for reading DLIS (RP66v1) files. It goes to great lengths to shield the end-user from format-specific details that are of no interest to him or her, such as exotic float-point formats, record segmentation, and compact dictionary representations. A lot of work goes on under the hood to provide an idiomatic Python interface that meshes well with other libraries and user expectations.

There are two main design choices that makes dlisio fast. First and foremost, it is based on the assumption that users will only care about a few bits and pieces of of a log at the time, so dlisio goes to great lengths to read as little as possible, and to user simpler data structures. In effect, this means that dlisio does a lot less work, assuming that random access in a file is fairly fast. Secondly, large parts of dlisio is written in C++, a

ISBN 978-82-8123-020-0

Table 4: Frames in the investigated log file

Frame name	Depth sampling period Δz [in]	Depth interval [mMD]	No. of channels
60B	6	3185.0–2474.8	289
20B	2	3184.9–2476.0	14
10B	1	3177.4–2479.2	7

fast system programming language. While the Python language offers many features for flexibility and ease-of-use, they come at the cost of speed. In order to provide powerful features and a pleasant user experience, dlisio integrates deeply with NumPy [8] and uses its structured arrays as an engine for Frame data.

Unlike some of its alternatives, dlisio is quite aware of the tight relationships between objects in DLIS files. The library provides a programmatic approach for for following the object references specified by the file. The initial release of dlisio was in late 2018, and the library is to this day under heavy development. The current alpha release of dlisio (0.1.16) can read almost all data in DLISv1 compliant files, and some common variations. The library is freely distributed under the free software license LGPLv3 [9]. It supports the major operating systems, and offers pre-built packages through PyPI. The online documentation [7] includes documentation of the library itself, as well as relevant parts of the DLISv1 standard.

3 Investigating acoustic well log data

In this section, we will focus on one specific log file from the Volve Data Village, namely the main pass of well F-11 B, shown highlighted in Table 2. Its filename in the dataset is `WL_RAW_PROD_AC-AIMG-CCL-GR_2013-06-05_2.DLIS`. The logged well section has a casing with an outer diameter of $\texttt{CSIZ} = 9\frac{5}{8}$ in and a nominal thickness of $\texttt{THNO} = 0.539$ in, where `CSIZ` ('Current Casing Size') and `THNO` ('Nominal Thickness of Casing') are parameters stored in the file. According to the cement log interpretation report from the dataset, the theoretical top of cement was estimated between 2670 mMD and 2980 mMD, where the latter depth accounts for cementing losses of up to $9\,\mathrm{m}^3$. ('mMD' means 'metres, measured depth', i.e. depth measured along the length of the borehole as opposed to true vertical depth.) The report also specifies that the casing is surrounded by another 14 in casing down to around 2570 mMD. Below that, the casing is surrounded by a borehole drilled with a bit size of $\texttt{BS} = 12\frac{1}{4}$ in (the 'Bit Size' parameter).

This file, like all of the other well integrity logs listed in Table 2, contains a single logical file that contains a single origin object. This means that all of the data in each disk file has been measured in the same well. All of the files' frames are indexed against borehole depth z, meaning that the differences between the frames mainly lie in their depth sampling period Δz, in addition to a small difference in the depth intervals covered by the frames. Every frame's index channel is called `TDEP` ('Tool Depth'), which provides the depth z_i of the ith depth sample of every channel in the corresponding frame as

$$z_i = \texttt{TDEP}(i). \tag{1}$$

This file has 310 channels across three frames, listed in Table 4, and 473 parameters.

Table 5: Overview of the investigated log file's tools (not every type of name is provided in the log file), along with the number of parameters and channels of each tool. Because not all of the file's parameters and channels belong to tools, this list contains fewer parameters (118 out of 473) and channels (275 out of 310) than the file.

Name	Generic name	Trademark name	Description	No. of parameters	No. of channels
ACTS-B	ACTS		Auxiliary Compression Tension Sub - B (Only external acquisition supported)	6	2
CALY	CCL	CAL-Y	Casing Anomaly Locator 3-3/8 in 31 Pin Heads	2	3
DSLT-H	SONIC	DSLT-H	Digitizing Sonic Logging Tool - H	41	17
DTC-H		DTC-H	Digital Telemetry Cartridge - Version H	1	2
LEHQT			Logging Equipment Head - QT, 3-3/8 inch 31 pin HPHT with Tension Sensor	4	0
SGTN	GR	SGT-N	Scintillation Gamma-Ray Tool	6	6
USIT	USIT-E	USIT-E	Ultrasonic Imaging	58	245

The tool metadata contains names and descriptions for every tool on the tool string used for the logging run. Table 5 shows the tools used in this file. dlisio lets us easily isolate the parameters and channels specific to every tool, and we will in the following sections look more closely at the parameters and channels of the sonic DSLT (Sec. 3.1) and ultrasonic USIT (Sec. 3.2) tools, which are given in Tables 6–9 in the Appendix.

When plotting data from this file, we restrict ourselves to the depth interval 2500–2800 mMD, as using the entire log would make the plots look overly crowded. The Jupyter Notebook accompanying this article [3] plots data from the entire log.

3.1 Sonic data

The Digitizing Sonic Logging Tool (DSLT) is a sonic cement bond logging tool from Schlumberger [4]. Such tools were first described in the literature in 1961 [10, 11], and Section 15-4.4 in [12] gives a good modern overview. In short, at various regularly spaced depths, an acoustic monopole transmitter on the tool emits an acoustic pulse that travels along the wellbore. The pulse reaches two acoustic receivers on the tool, whose distances to the transmitter are 3 ft and 5 ft.

The waveforms recorded by the receivers, which we will investigate in Secs. 3.1.1 and 3.1.2, are then analysed to draw out information about the well integrity. The first waveform component to arrive represents the fastest travel path from source to receiver. Typically, this path is where the emitted pulse travels through the fluid to the casing, where it excites an extensional wave[2] travelling along the casing that continually leaks a wavefront back into the fluid inside the casing, which then impinges on the receivers.

As the casing wave travels, it is attenuated due to energy loss into the casing fluid and the material outside the casing. When solids such as cement are bonded to the outside

[2]Specifically, the S0 leaky Lamb wave mode [13, 14].

ISBN 978-82-8123-020-0

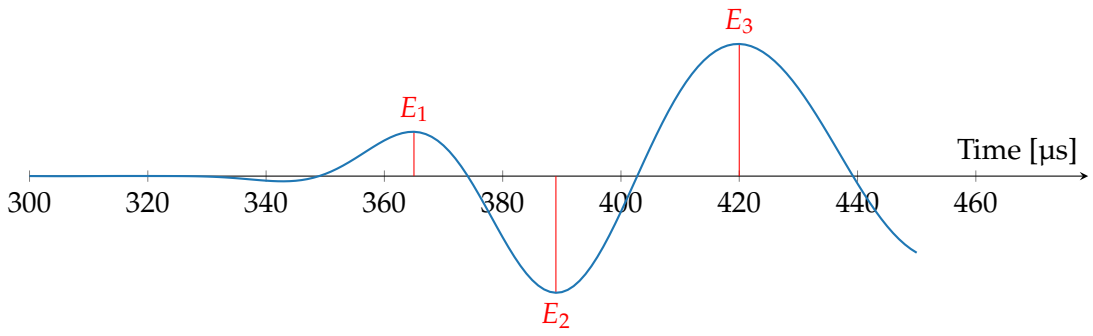

Figure 3: Example of the first waveform component in the near receiver, highlighting the peak and trough amplitudes E_1, E_2, and E_3. (This shows the same waveform as WF2 in Figure 5, but here it is upsampled 10 times by means of a Fourier method.)

of the casing, this attenuation is strong, which means that the first waveform component will then be much weaker than if only liquids such as water or mud touch the outside of the casing [13]. Thus, the amplitude of the first component gives us an idea of the material outside the casing: Very high amplitudes correspond to full coverage of fluids, while very low amplitudes correspond to full coverage of bonded solids. If bonded solids partly cover the outside of the casing, the amplitudes fall between at a point between these two extremes depending on the proportion of solid coverage [10, 12, 15]. The amplitude E_1 in mV of the first peak in the waveform measured by the near receiver, as Figure 3 shows, is known as the cement bond log (CBL) [10, 12, 13, 16]. The CBL is considered one of the most important results of the sonic tool, and we will investigate it further in Sec. 3.1.3

Tables 6 and 7 in the Appendix provide the investigated log file's parameters and channels belonging to the DSLT. Very few of these parameters and channels have a meaning that is obvious from a first look, and further documentation on them is largely not publicly available. To understand them properly, we need to investigate them more closely. This is what we will be looking into in the next few sections.

3.1.1 Sonic waveforms

Table 7 shows us that have two channels WF1 ('Waveform 1') and WF2 ('Waveform 2') in frame 20B in the file, which we see from Table 4 has a depth resolution $\Delta z = 2\,\text{in}$. Each of the two channels has a dimension of 250, i.e. contains an array of 250 values at each depth sample.[3] Thus, we may expect that these two channels contain recorded waveforms of 250 samples, measured every 2 in, for the two receivers. In other words, each channel can be seen as a two-dimensional array, e.g. $\texttt{WF1}(i, k)$, representing a sort of image where i indexes the depths z_i at which measurements were made, and k indexes samples. The waveform channels are not stored with any units, and there does not seem to be any parameters or channels[4] available that can convert the waveform values to any physical unit such as Pa or mV.

We cannot yet say for certain whether WF1 corresponds to the near receiver and WF2 to the far one, or the other way around. To find out, we must investigate what the channels

[3] This seems to be what the DWCO = 250 ('Digitizer Word Count') parameter specifies. The value of DWCO also matches the dimension of WF1 and WF2 in the other log files, even when the value is not 250.

[4] While the WF1N ('Waveform 1 Normalization Factor') and WF2N ('Waveform 2 Normalization Factor') channels seem like promising candidates, they contain nothing but the value 1 at every depth.

ISBN 978-82-8123-020-0

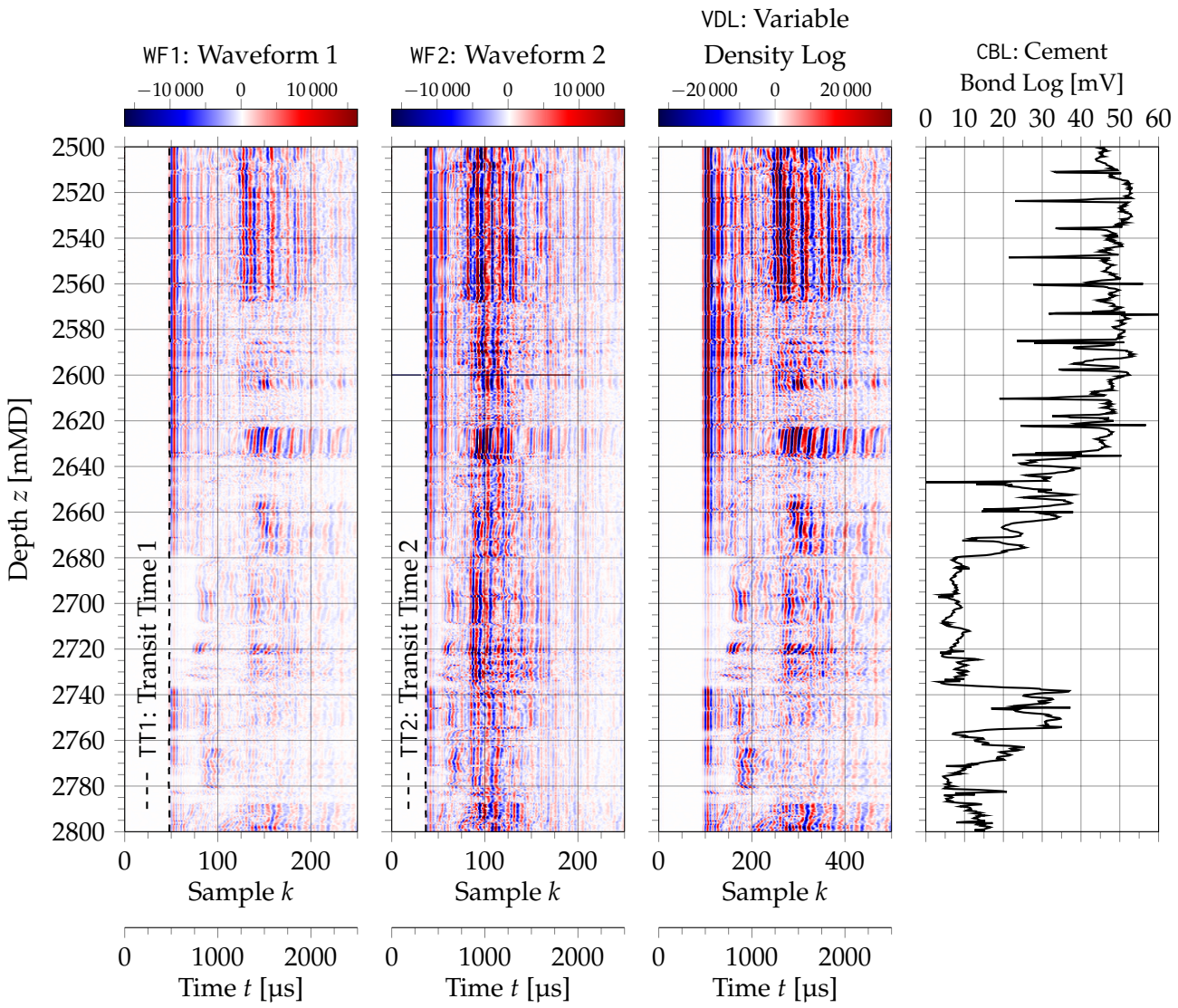

Figure 4: Various waveform and CBL related channels from the sonic tool

contain. Figure 4 shows WF1 and WF2 in a depth interval of the log. (Ignore the figure's time axes and the travel time curves for now; we will find these later.) Looking at these figures, we can make a few observations:

- The waveforms arrive earlier in WF2 than in WF1 (given that both receivers' recorded waveforms start at the same time)
- The waveforms are stronger in WF2 than in WF1 (given that both receivers are calibrated similarly)
- Some components of the waveforms travel through fast paths and arrive early, while other components travel through slower paths and arrive later. The spread between the early (fast-path) and late (slow-path) components is larger in WF1 than in WF2, indicating that WF1 has been measured further away.

These three observations all indicate that WF1 contains waveforms from the far receiver at 5 ft, while WF2 contains waveforms from the near receiver at 3 ft.

ISBN 978-82-8123-020-0

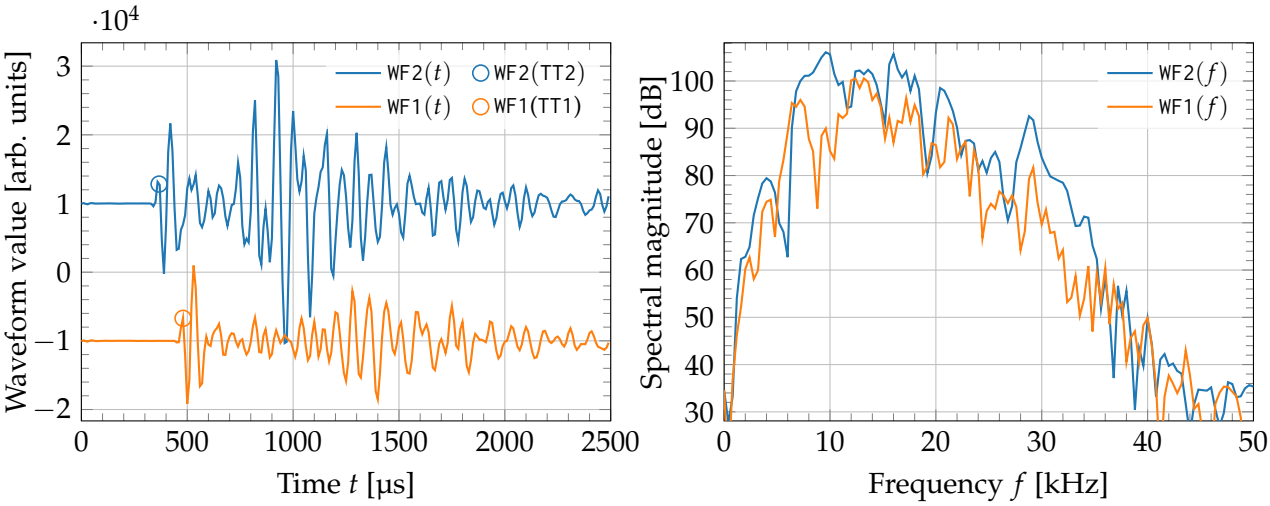

Figure 5: *Left:* Waveforms WF2 and WF1 at a depth of 2552.60 mMD, shifted by 10^4, and the first arrivals given by TT2 and TT1. *Right:* The waveforms' frequency spectra.

In the WF1 and WF2 channels, we can also see disturbances roughly every 12 m. These disturbances are caused by the casing collars used to join the individual casing joints, which are steel pipes with lengths generally around 40 ft or 12 m. Such disturbances are present in most of the channels that we will investigate in this article.

Another observation we can make from both channels are strong later arrivals in the waveforms from 2600 mMD down to almost 2570 mMD. As this corresponds to the aforementioned region where the casing is surrounded by another 14 in casing, we can expect these later arrivals to be connected with that casing. This information came from the log report and is not present in the file, showing us that we must sometimes look at information outside the file to understand the information in it.

We can now identify the different waveforms by depth and receiver. However, we do not know the time axis for the 250 samples in each waveform. In other words, we do not know what sampling rate F_s they were recorded at, or whether both receivers start recording their waveforms at the same time.

To solve this, we must look at the time-related parameters and channels in Tables 6 and 7, respectively. There, we find the two channels TT1 ('Transit Time 1') and TT2 ('Transit Time 2'), which represent the elapsed time between the transmitter firing a pulse and the first arrival of the pulse in each receiver [12]. We also find the parameter DSIN = 10 µs ('Digitizer Sample Interval'), which may represent the sampling period $T_s = 1/F_s$.

If we assume this to be the sampling period and that that both receivers start recording at time $t = 0$,[5] defined as when the transmitter fires the pulse, we find a time axis where

$$t_{\mathrm{WF}}(k) = 10k\,\mu\mathrm{s} \quad \text{for each sample} \quad k \in [0\,..\,(\mathtt{DWCO} - 1)]. \tag{2}$$

We can then compare the waveforms on this time axis against the transit time channels to verify that they match each other. Figure 4 shows that there is indeed such a match, confirming our assumptions.

To take a closer look at the actual waveforms, we extract a waveform from either receiver in the upper part of the well and show them in Figure 5, highlighting their values at the times indicated by the corresponding transit time channels. We find that the transit

[5] This may be what the DDEL = 0 µs ('Digitizing Delay') parameter specifies.

ISBN 978-82-8123-020-0

time closely matches the first visible peak in each waveform. For both waveforms, we can see that their first components (which typically arrive via the casing) are similarly shaped. The later parts of the signal, however, are not very similarly shaped, as they consist of a mix of multiple components travelling through different paths from the transmitter to the receivers.

Figure 5 also shows the frequency spectra of each waveform (after windowing with a von Hann window). These show that the waveforms are approximately band-limited and have most of their frequency content at roughly 6–22 kHz. The pulses originally emitted by the transmitters may have more high-frequency content, as the fluid inside the casing will attenuate higher frequencies more.

In summary, the `WF1` and `WF2` channels store the acoustic waveforms recorded by the receivers at 5 ft and 3 ft, respectively. These waveforms are stored at a sampling period $T_\mathrm{s} = \texttt{DSIN} = 10\,\mu\mathrm{s}$ ('Digitizer Sample Interval'), corresponding to a sampling rate of $F_\mathrm{s} = 100\,\mathrm{kHz}$. The waveforms are stored in arbitrary units, and there seems to be no straightforward way to convert them to any physical unit. The `TT1` and `TT2` channels represent the arrival times of the first waveform peaks in `WF1` and `WF2`, respectively.

3.1.2 Variable density log (VDL)

Plots like those of `WF1` and `WF2` in Figure 4, where waveforms at different depths are displayed as an image, are typically known as variable density logs (VDLs) in the literature. The commonly plotted VDLs take their waveforms from the far receiver at 5 ft, according to the literature [16]. (We have already seen that `WF1` is also taken from that same receiver.)

Indeed, Table 7 shows that there is a channel called `VDL` ('Variable Density Log') in the same frame `20B` as the `WF1` and `WF2` channels, with a dimension of 500 (i.e., 500 samples per depth). Figure 4 shows this channel and lets us compare it against the other waveforms. We see that the `VDL` channel looks very similar to the `WF1` channel, but with a different scaling. The twice as high number of samples thus points towards a twice as high sampling rate, i.e. a `VDL` sampling rate of 200 kHz, which gives us a time axis where

$$t_\mathrm{VDL}(k) = 5k\,\mu\mathrm{s} \quad \text{for each sample} \quad k \in [0\,..\,(2 \cdot \texttt{DWCO} - 1)]. \tag{3}$$

Figure 6 compares a waveform from the `VDL` channel with the `WF1` waveform at the same depth. We see that the two have the same shape, but that `VDL` is stronger. In fact, the `VDL` signal is so strong that it is saturated. In other words, its strongest peaks and troughs are clipped because they cannot exceed the range $[-32768, 32767]$ of the 16-bit integers used to store the `VDL` channel. To determine whether `VDL` is related to `WF1` through a scaling, we determine a scaling factor over every depth and common sample as

$$s_\mathrm{VDL} = \operatorname*{median}_{i,k} \left[\frac{\texttt{VDL}(i, 2k)}{\texttt{WF1}(i, k)} \right]. \tag{4}$$

Here, we use the median instead of the mean to account for the clipping of the `VDL` channel. Figure 6 shows that the scaled `WF1` waveform matches the `VDL` waveform very well, confirming that `VDL` and `WF1` are related through scaling.

In summary, the `VDL` channel contains a scaled version of the same waveforms as the `WF1` channel, but at twice the sampling rate $F_\mathrm{s} = 200\,\mathrm{kHz}$. In some of the other files in the dataset, `VDL` is in another frame than `WF1`, with a higher depth resolution. It is not clear

ISBN 978-82-8123-020-0

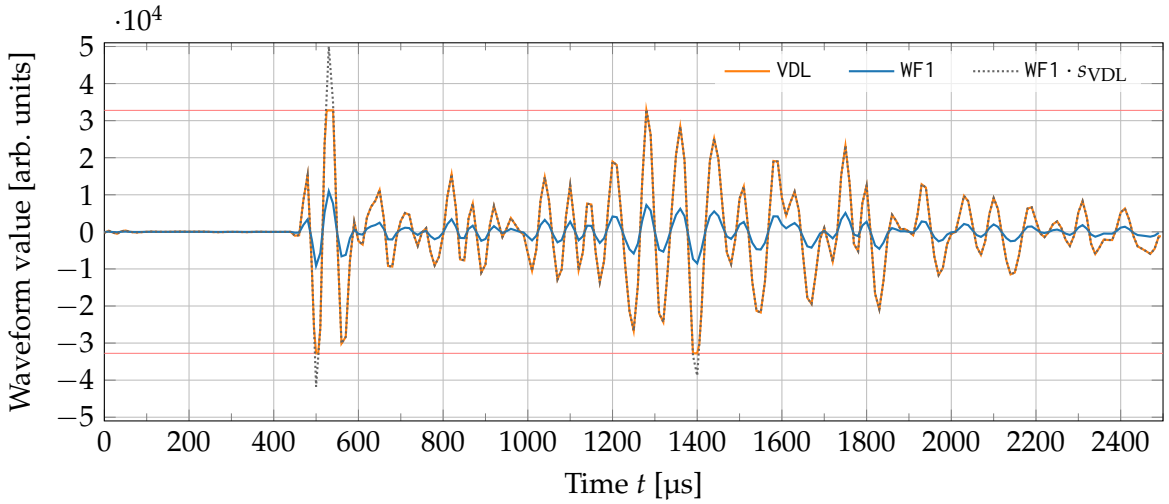

Figure 6: VDL and WF1 waveforms at a depth of 2552.60 mMD, and the WF1 waveform scaled to fit VDL. The horizontal red lines indicate the minimum (-32768) and maximum (32767) values of the 16-bit integers used to store the VDL and WF1 channels.

where these differences in time and depth resolution comes from. Either, the waveforms are originally recorded at a higher sampling rate and then downsampled to WF1 and WF2, or VDL is artificially upsampled.

3.1.3 Cement bond log (CBL)

As we explained at the start of Section 3.1, the CBL is found as the amplitude E_1 in mV of the first peak in the waveform from the near receiver [10, 12, 13, 16], as shown in Figure 3. From that figure, we can also see that CBL and WF2 are linked: Where the first component of WF2 is weak, CBL is low, and where the first component of WF2 is strong, CBL is high. The near-constantly high CBL values from the top of the log down to 2632 mMD indicates that this is a free-pipe interval, with nearly only fluids touching the outside of the casing.

We can implement our own CBL processing that first determines the first peak amplitude $E_1'(z_i)$ of every waveform in the WF2 channel. We do this through a three-step algorithm:

1. Upsample every waveform. Here, we use a Fourier method, which is very accurate for approximately band-limited signals such as our waveforms, but which treats the signal as being periodic. The method can therefore give us ringing due to the jump between the final sample and the first sample. To avoid that ringing, we apply Tukey windows before upsampling, which ensure that the waveforms start and end at the value 0.

2. Find the peak sample of each waveform in the region where we expect the peak to be.

3. Estimate the peak amplitude through a quadratic fit of the peak sample and its neighbours.

However, we find $E_1'(z_i)$ in the arbitrary units in which the waveforms were stored. To be able to compare our CBL processing with the reference CBL channel, we need to approximately transform it into an $E_1(z_i)$ value in mV. This should be a matter of finding a

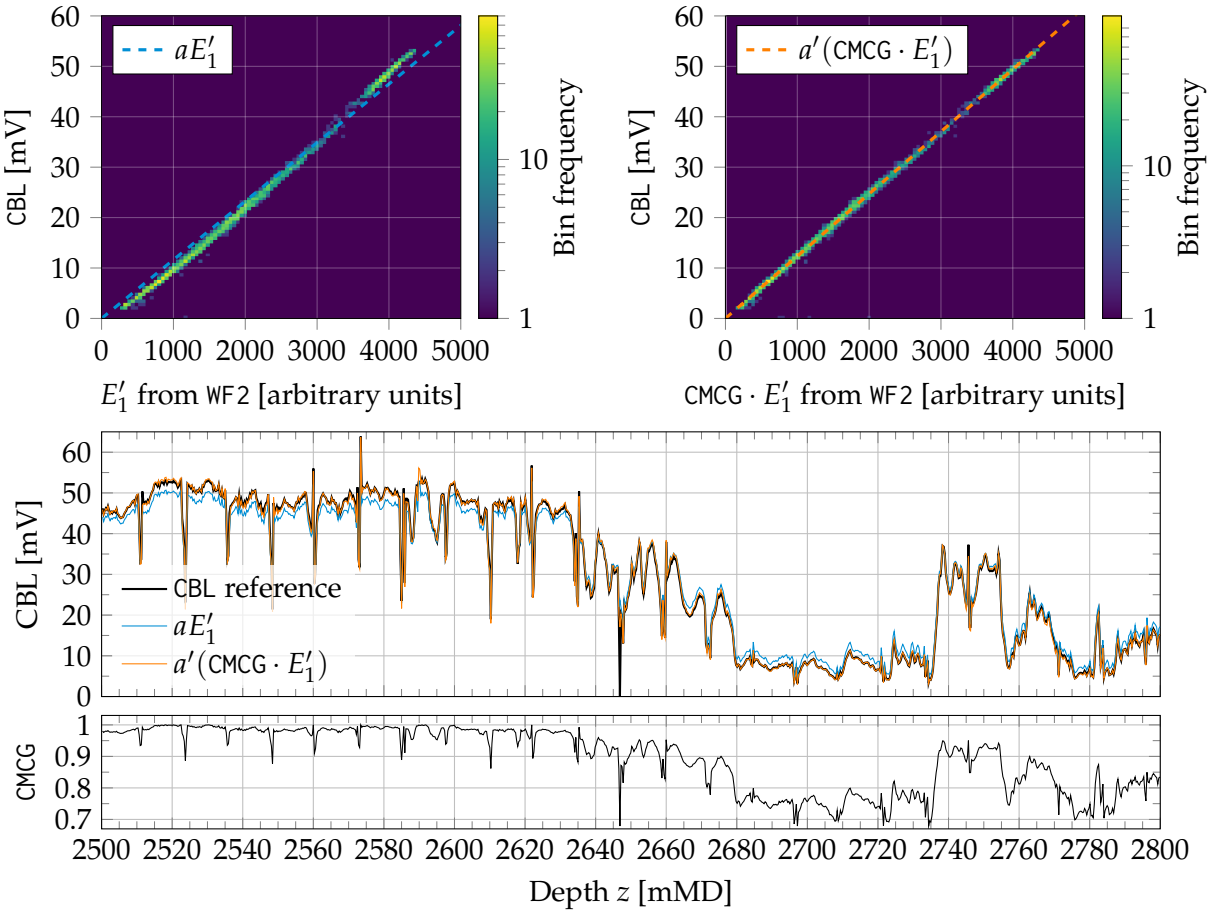

Figure 7: *Top left:* 2D histogram of the E_1' values against the CBL reference values, and the resulting scaling aE_1'. *Top right*: Same, but with E_1' values adjusted with the CMCG channel. *Middle:* Direct comparison of transformed E_1' values and the CBL channel reference. *Bottom:* The CMCG ('CBL Cement Type Compensation Gain') channel.

conversion factor a in mV/arb.units so that $aE_1'(z_i)$ is a CBL value in mV. Here, we take the simplest possible approach and find a from a linear fit, without intercept, between $E_1'(z_i)$ and $\text{CBL}(i)$ (omitting invalid negative values in E_1 and CBL when fitting).[6]

The top left plot of Figure 7 shows a slight S-shape in the relation between E_1' and CBL, which means that a simple conversion factor is not sufficient to connect the two. We see the effect of this in the middle plot of Figure 7; the scaled $E_1'(z_i)$ underpredicts high CBL values and overpredicts low ones. This implies that there is some kind of correction applied on the way from the waveform to the CBL channel.

Looking for possible candidates for such a correction in Tables 6 and 7, we find the CMCG ('CBL Cement Type Compensation Gain') channel, shown in the bottom plot of Figure 7. The top right plot in Figure 7 shows that multiplying $E_1'(z_i)$ with $\text{CMCG}(i)$ straightens out the aforementioned S-shape. Now, a simple conversion factor a', calculated similarly to a, connects $\text{CMCG} \cdot E_1'$ and CBL well. The middle plot of Figure 7 shows that this

[6] Even without a reference like the CBL channel, it would be possible to find an approximate conversion factor by another approach: Identify a free-pipe region of the well and find a conversion factor a that ensures that the CBL values inside the region match a predetermined reference free-pipe CBL value. (From Table 6, we see that the parameter $\text{CBRA} = 53\,\text{mV}$ ('CBL LQC Reference Amplitude in Free Pipe') provides such a reference value.)

ISBN 978-82-8123-020-0

corrected and scaled $E_1'(z_i)$ fits the CBL channel very well.

However, it is not clear how the CMCG channel is calculated, except that its shape is very similar to the CBL values, and that it is probably related to the parameter CMCF $=$ 0.679 ('CBL Cement Type Compensation Factor'). Such a correction is not described in the CBL literature cited herein. While we could surely investigate this further, we will not do so here.

In summary, it is possible to reprocess CBL at every depth z_i from the first peak amplitudes $E_1'(z_i)$ of the waveforms in the WF2 channel in the file, although there are some issues with this. The main issue is that the waveforms WF2 and WF1 are stored in arbitrary units instead of mV, and the file contains no obvious way to convert the units. Here, we have simply found a conversion factor by fitting to the values in the CBL channel, though a standalone CBL reprocessing algorithm would need to find the conversion factor in another way, for example by ensuring that free-pipe regions match a given reference free-pipe CBL value.

3.1.4 Attenuation and bond index

As we discussed at the beginning of Section 3.1, the sonic tool excites a wave travelling along the casing, whose attenuation is stronger the more the outside of the casing is covered by bonded solids. While the first peak amplitudes E_1 in the near or far receiver can tell us of the attenuation accumulated as the wave has travelled along the casing, we can also estimate this attenuation directly when we have two or more receivers. Laboratory tests where bonded cement covered part of the casing found this attenuation to be approximately linearly related to the proportion of coverage [10, 13, 15].

For amplitudes $E_{1,\text{near}}(z_i)$, $E_{1,\text{far}}(z_i)$ measured at the depth z_i in two receivers separated by a distance L_{RR} (where $L_{\text{RR}} = 2\,\text{ft}$ in our case), this attenuation can be estimated as [12, 13]

$$\alpha(z_i) = \frac{20}{L_{\text{RR}}} \log_{10}\left(\frac{E_{1,\text{near}}(z_i)}{E_{1,\text{far}}(z_i)}\right). \tag{5a}$$

If L_{RR} is measured in m, the units of α are dB/m. If only one receiver is available, at a distance L_{TR} from the transducer, the attenuation can still be approximated as [12, 15]

$$\alpha'(z_i) = \frac{20}{L_{\text{TR}}} \log_{10}\left(\frac{E_{1,\text{free}}}{E_1(z_i)}\right), \tag{5b}$$

where $E_{1,\text{free}}$ is the amplitude for free-pipe intervals in the same receiver as $E_1(z_i)$ was measured.

While Table 7 shows that the file has no attenuation channel, we can calculate it ourselves. First, we calculate $E_{1,\text{near}}(z_i)$ and $E_{1,\text{far}}(z_i)$ from the corresponding $E_1'(z_i)$ values found in Section 3.1.3 as $E_1(z_i) = a'[\text{CMCG}(i) \cdot E_1'(z_i)]$, to stay consistent with the corrections used in the file. The resulting values are plotted to the left in Figure 8. Then, we insert the values into (5a) and (5b). In (5b), we only have a free pipe reference value $E_{1,\text{free}}$ available for the near receiver, namely the parameter CBRA $= 53\,\text{mV}$ ('CBL LQC Reference Amplitude in Free Pipe'), and therefore we only calculate α' for the near receiver, with $L_{\text{TR}} = 3\,\text{ft}$.

The middle plot of Figure 8 shows the resulting attenuations. The attenuations are 0–2 dB/m in the free pipe interval, and higher where solids are bonded to the casing. Except for the free pipe interval, α is generally somewhat lower than α'. α is also more

ISBN 978-82-8123-020-0

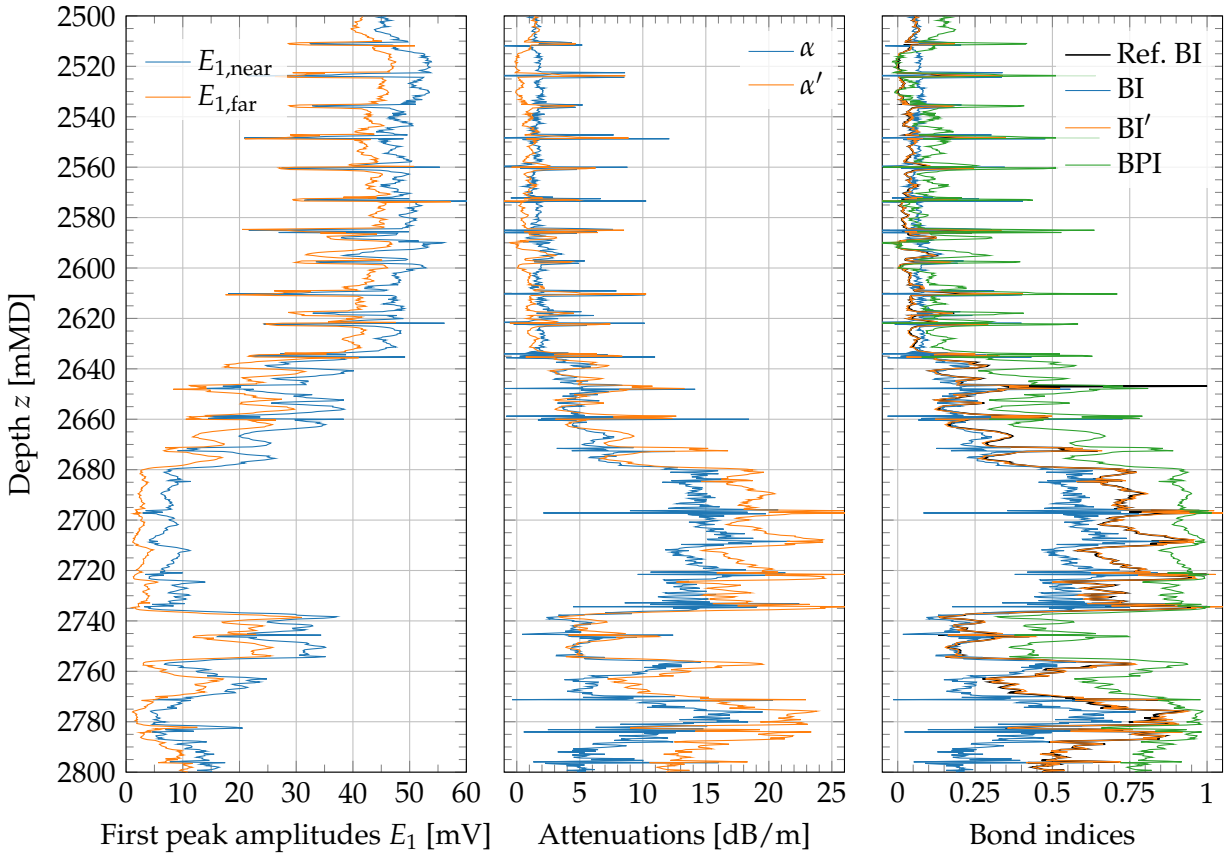

Figure 8: First peak amplitudes E_1, and attenuations and bond indices calculated from them

noisy than α'. This may be because it is affected by the noise in both $E_{1,\text{near}}$ and $E_{1,\text{far}}$, while α' is only affected by the noise in $E_{1,\text{near}}$.

Another way to represent the attenuation is the *bond index* $\text{BI}(z_i)$ [12, 13, 17], which exploits the near-linear relationship between attenuation and proportion of coverage. The bond index scales and shifts the attenuation to lie between 0 and 1 as [12]

$$\text{BI}(z_i) = \frac{\alpha(z_i) - \alpha_{\text{free}}}{\alpha_{\text{full}} - \alpha_{\text{free}}}, \tag{6a}$$

where α_{free} is the attenuation in free-pipe intervals and α_{full} is the attenuation in intervals where the casing is fully covered by bonded solids. Thus, $\text{BI}(z_i) = 0$ corresponds to no coverage and $\text{BI}(z_i) = 1$ corresponds to full coverage. The idea is that $\text{BI}(z_i)$ should correspond to the proportion of casing covered by bonded solids, so that e.g. half coverage corresponds to $\text{BI}(z_i) = 0.5$. If only one receiver is available, we can calculate an approximate bond index from (5b),

$$\text{BI}'(z_i) = \frac{\alpha'(z_i) - \alpha_{\text{free}}}{\alpha_{\text{full}} - \alpha_{\text{free}}}. \tag{6b}$$

Similar options are also possible. In the literature, the relations $\text{BI}(z_i) = \alpha(z_i)/\alpha_{\text{full}}$ and $\text{BI}'(z_i) = \alpha'(z_i)/\alpha_{\text{full}}$ are sometimes used [13, 17–19], which are related to (6a) and (6b) through the assumption $\alpha_{\text{free}} \approx 0$. Another alternative index is the *bond percentage index* $\text{BPI}(z_i)$ [12, 18, 19], which scales and shifts the CBL amplitude $E_1(z_i)$ rather than the attenuation $\alpha(z_i)$:

$$\text{BPI}(z_i) = \frac{E_{1,\text{free}} - E_1(z_i)}{E_{1,\text{free}} - E_{1,\text{full}}}. \tag{6c}$$

ISBN 978-82-8123-020-0

We can now calculate (6a), (6b), and (6c) and compare them, using the BI ('Bond Index') channel from the file as a reference. As before, we use the parameter CBRA $= 53\,\mathrm{mV}$ for $E_{1,\mathrm{free}}$, and now additionally use the parameter MSA $= 3.669\,\mathrm{mV}$ ('Minimum Sonic Amplitude') for the full-coverage amplitude $E_{1,\mathrm{full}}$. We use these to calculate approximate values of free-pipe and fully cemented attenuation using (5b), giving $\alpha_{\mathrm{free}} = 0\,\mathrm{dB/m}$ and $\alpha_{\mathrm{full}} = 25.37\,\mathrm{dB/m}$.

The rightmost plot in Figure 8 compares the various bond indices. BI and BI′ naturally have just the same shapes as α and α', respectively, as they are related by simple scaling and shifting. BI′ matches the reference BI channel in the file well, which shows that the two are calculated in the same way. The slight differences we can see between the two come from the reference being calculated from the CBL channel while we calculate BI′ from our own $E_{1,\mathrm{near}}$ found in Section 3.1.3, as well as BI having been truncated so that $\mathrm{BI}(i) \in [0, 1]$, while we did not perform such a truncation. BPI predicts higher values than BI and BI′, which is consistent with results from [12] showing that BPI overestimates the solid coverage while BI slightly underestimates it.

In summary, we can use the first peak amplitudes E_1 found from the waveform channels WF1 and WF2 to compute various attenuations and bond indices. In other words, we can compute more bond metrics than just the single bond index variant provided in the file.

3.2 Ultrasonic data

The Ultrasonic Imaging Tool (USIT) is an ultrasonic pulse-echo tool developed by Schlumberger. It was first described in the literature in the early 1990s [20, 21], and Section 15-4.5 in [12] gives a good overview of some further details.

The basic mechanism of the USIT (and similar pulse-echo tools) is that an ultrasonic transducer emits a pulse onto the casing at normal incidence. While most of the initial pulse is reflected at the fluid-casing interface, a small part of the pulse is transmitted into the casing, where it resonates. The total reflected pulse is then recorded by the transducer, and consists of the strong first reflection from the casing, known as the first-interface echo (FIE), followed by a decaying resonance.

Based on a single such pulse, we can estimate the distance from the transducer to the casing from the pulse's travel time,[7] the thickness d of the casing from the pulse's resonance frequency, and the acoustic impedance $Z = \rho c$ of the material on the far side of the casing from the pulse's resonance decay [20]. (Here, ρ and c are the material's density and pressure wave speed, respectively.)

The USIT transducer is mounted on a head that rotates[8] as the USIT moves through the well. This lets the USIT emit its pulses at different azimuthal angles. In other words, unlike the sonic tool which can only sample the well in depth z_i, the USIT can sample the well in both depth z_i and azimuthal angle φ_j. The USIT's angular resolution can be set to either $\Delta\varphi = 5°$ (giving $360°/5° = 72$ measurements per depth z_i) or $\Delta\varphi = 10°$ (giving $360°/10° = 36$ measurements per depth z_i).

[7] This assumes that we know the the sound speed of the fluid inside the casing. This is either estimated based on prior knowledge of the well or measured as part of a fluid properties measurement pass, which we mentioned earlier in Section 2.1.

[8] According to [20], it rotates at 7.5 Hz. However, the file contains a channel RSAV ('Motor Revolution Speed'), showing that this rotational speed can vary with depth. For this file, $\mathrm{mean}_i[\mathrm{RSAV}(i)] = 6.85\,\mathrm{Hz}$.

ISBN 978-82-8123-020-0

The processing algorithm to determine the casing thickness $d(z_i, \varphi_j)$ and the acoustic impedance $Z(z_i, \varphi_j)$ behind the casing is called *Traitement Très Tôt* (French for 'very early processing'), or simply T^3, and it is described in [20, 21]. In addition, the measured distances between transducer and casing can be processed to determine how the tool is eccentered, as well as the casing's inner radius $r_\mathrm{i}(z_i, \varphi_j)$. (From this we can also calculate the casing outer diameter as $r_\mathrm{o}(z_i, \varphi_j) = r_\mathrm{i}(z_i, \varphi_j) + d(z_i, \varphi_j)$. These processing algorithms are somewhat complex, and we will not look at them more closely in this article.

Tables 8 and 9 in the Appendix provide the investigated log file's parameters and channels belonging to the USIT. From the USIT parameter table, we can already find the USIT log's angular resolution through the HRES $= 5°$ ('Horizontal Resolution') and NWPD $= 72$ ('Number of Waveforms per Depth') parameters. In other words, $\Delta\varphi = 5°$, with 72 measured waveforms per depth z_i, and

$$\varphi_j = j\Delta\varphi \quad \text{for each} \quad j \in [0\,..\,(\mathtt{NWPD} - 1)]\,. \tag{7}$$

From the USIT channel table, we can see that every channel is in the frame 60B, which has a depth resolution $\Delta z = 6\,\mathrm{in}$ according to Table 4. We can also see that a number of channels have a dimension of 72, i.e. 72 values per depth. We can assume that these values belong to the different azimuthal angles φ_j.

3.2.1 Casing geometry channels

First, we investigate some basic channels providing casing geometry information in angle and depth:

IRBK ('Internal Radii Normalized') should represent the internal casing radius, with some kind of normalisation that we will have to investigate further.

T2BK ('Thickness of Casing minus Nominal') should represent the difference between the actual casing thickness and the nominal casing thickness given by the parameter THNO $= 0.539\,\mathrm{in}$.

ERBK ('External Radii') should represent the external casing radius.

The data arrays in each channel, e.g. T2BK(i, j), have two dimensions: i indexes depths z_i and j indexes azimuthal angles φ_j. Figure 9 shows the three channels.

From the long names of the channels, we can expect to find the casing thickness as

$$d(z_i, \varphi_j) = \mathtt{T2BK}(i, j) + \mathtt{THNO}, \tag{8a}$$

and the casing external radius as

$$r_\mathrm{o}(z_i \varphi_j) = \mathtt{ERBK}(i, j). \tag{8b}$$

However, the long name of the IRBK channel refers to some as of yet unknown normalisation. Looking up IRBK in Schlumberger's Curve Mnemonic Dictionary [22], we find that the channel has the property Differential_Radius, which means that it represents the difference between the actual radius of an object and its average radius. In Table 9, we can find a channel IRAV ('Internal Radius Averaged Value') with one value per depth, also plotted in Figure 9. We may then expect that we can find the casing inner radius as

$$r_\mathrm{i}(z_i, \varphi_j) = \mathtt{IRBK}(i, j) + \mathtt{IRAV}(i). \tag{8c}$$

ISBN 978-82-8123-020-0

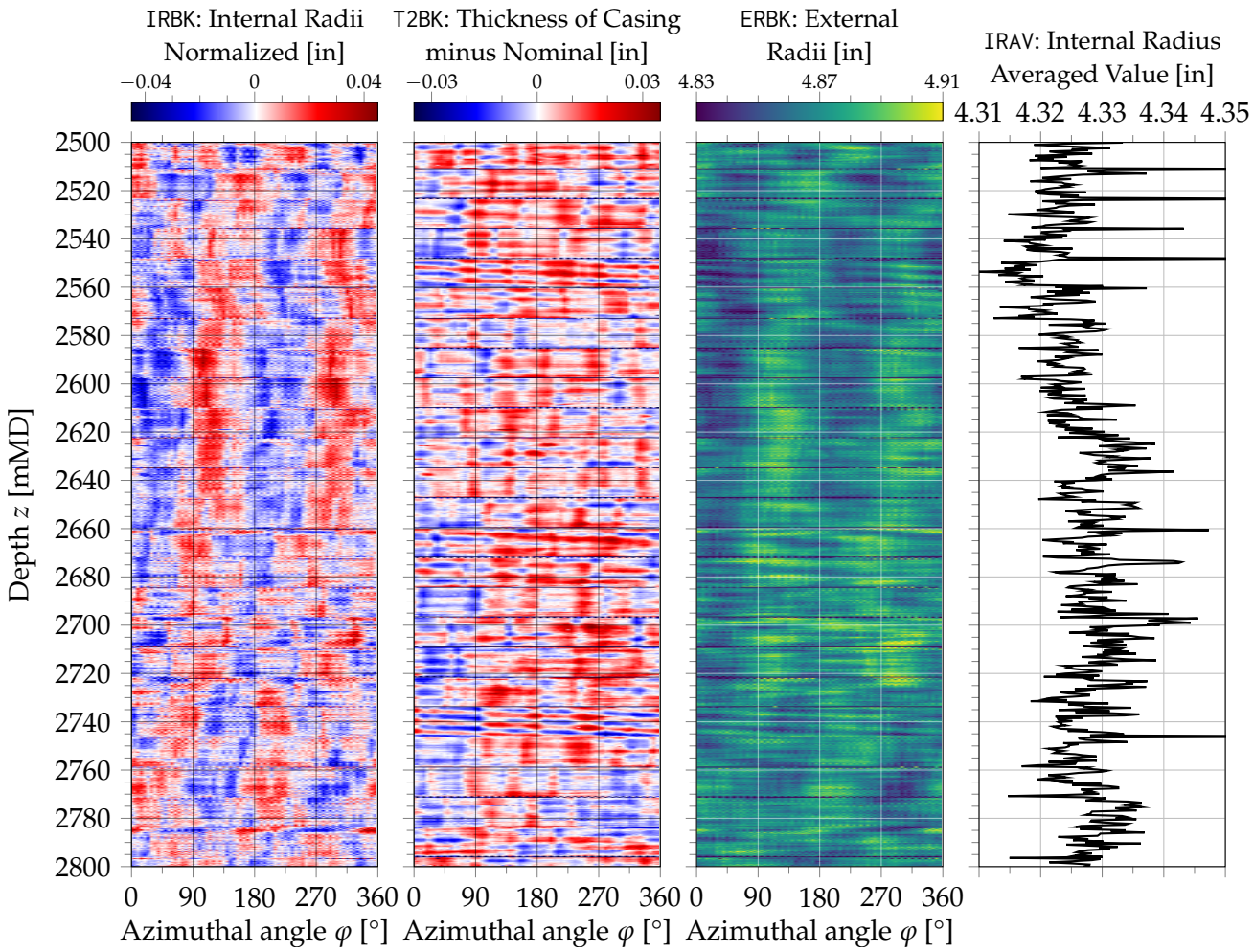

Figure 9: Various casing geometry related channels from the ultrasonic tool

To verify all of this, we can compare two methods of finding the internal radius using different channels, namely (8c) and

$$r_{\mathrm{i}}(z_i, \varphi_j) = r_{\mathrm{o}}(z_i, \varphi_j) - d(z_i, \varphi_j) = \mathtt{ERBK}(i,j) - [\mathtt{T2BK}(i,j) + \mathtt{THNO}]\,. \tag{8d}$$

In fact, a numerical comparison shows that (8c) and (8d) agree with each other to the order of machine epsilon.

In summary, we can find the casing geometry estimated by the USIT from various USIT channels and parameters as specified in (8a)–(8d).

3.2.2 Ultrasonic waveforms

Table 9 shows that the file contains channels of USIT waveforms, named `U001` ('Waveform for Azimuth 01'), `U002`, and so forth until `U072` ('Waveform for Azimuth 72').[9] Each of these has a dimension of 120, which also corresponds with the `NPPW` = 120 ('Number of Points Per Waveform') parameter. We may assume that these 120 values represent samples of a time signal, just as for the sonic waveforms. Thus, the kth waveform sample

[9] In a USIT log with an azimuthal resolution of 10°, the last channel would be `U036` ('Waveform for Azimuth 36').

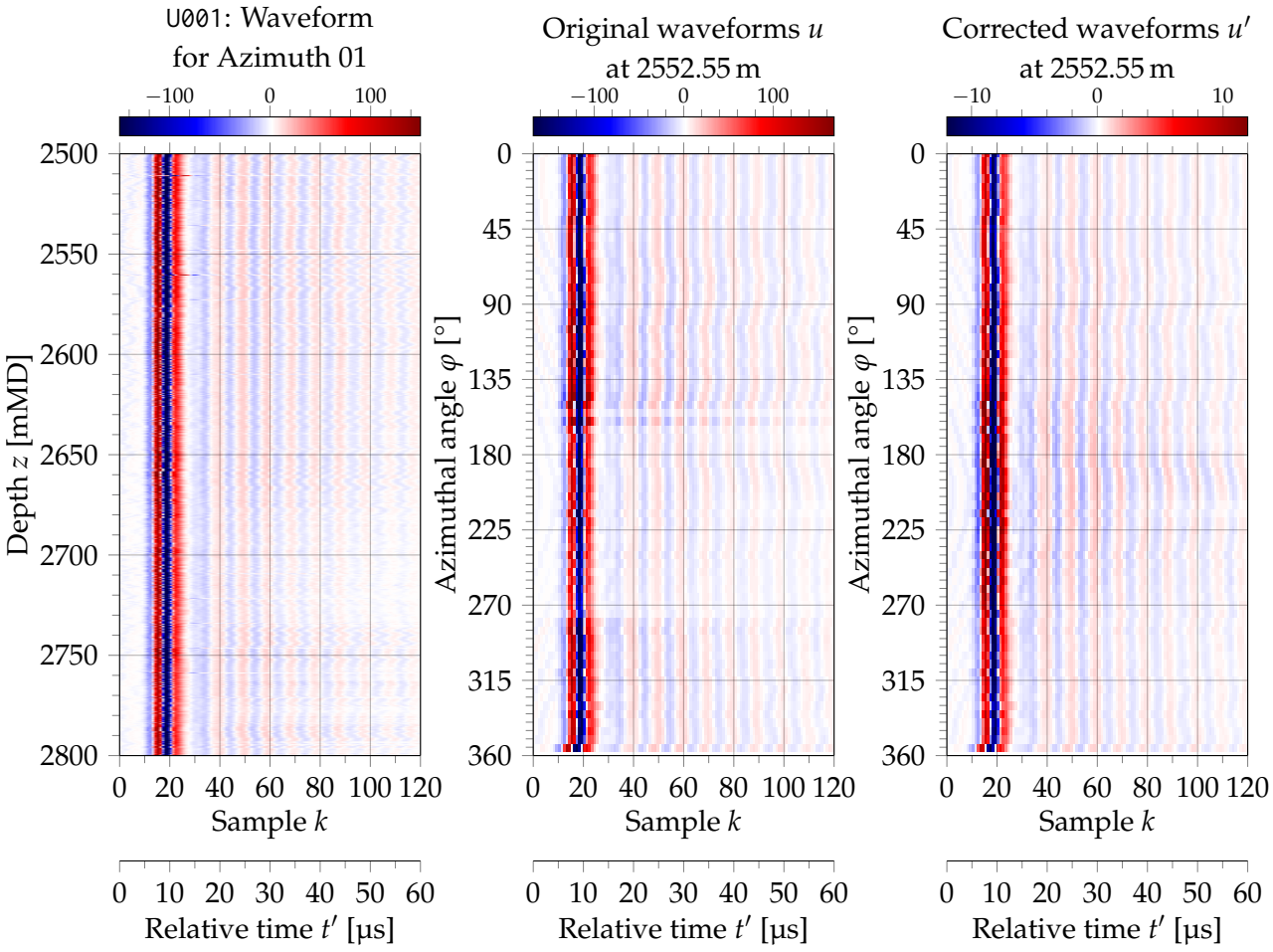

Figure 10: *Left:* The U001 channel. *Middle:* Waveforms $u(2552.55\,\text{mMD}, \varphi_j, k)$. *Right:* Waveforms $u'(2552.55\,\text{mMD}, \varphi_j, k)$, corrected according to the WAGN channel. The relative time t' in these plots represents the time since the first sample, and is covered in Section 3.2.3.

at depth z_i and angle φ_j is

$$u(z_i, \varphi_j, k) = \texttt{U###}_j(i, k). \tag{9}$$

Here, we use $\texttt{U###}_j$ to represent the ultrasonic waveform channel corresponding to the jth azimuth, e.g., $\texttt{U###}_0 = \texttt{U001}$, and $\texttt{U###}_{71} = \texttt{U072}$.

The left plot of Figure 10 shows $\texttt{U001} = u(z_i, 0, k)$ in a depth interval of the log. (Ignore the figure's time axes for now; we cover them in Section 3.2.3.) We can see that the waveform resonance decays more quickly in the depth range 2680–2735 mMD, where the CBL curve in Figure 4 indicates solids behind the casing. Faster decay with solids behind the casing is also what we would expect from the theory of these pulse-echo measurements [20].

The middle plot shows the waveforms at every azimuthal angle at a depth $z_i = 2552.55$ mMD. In the range 165–275°, we see a band of waveforms that are generally weaker than the others. We see the same effect for slightly different angular ranges when plotting the waveforms at other depths.

This difference in waveform strengh could indicate that waveforms were scaled differently before being stored. Looking through Table 9 for channels that may specify the scaling of each waveform, we find two likely candidates, namely UPGA ('USIC Programmable Gain Amplitude of Waves') and WAGN ('Waveform Applied Gain'). Both have

ISBN 978-82-8123-020-0

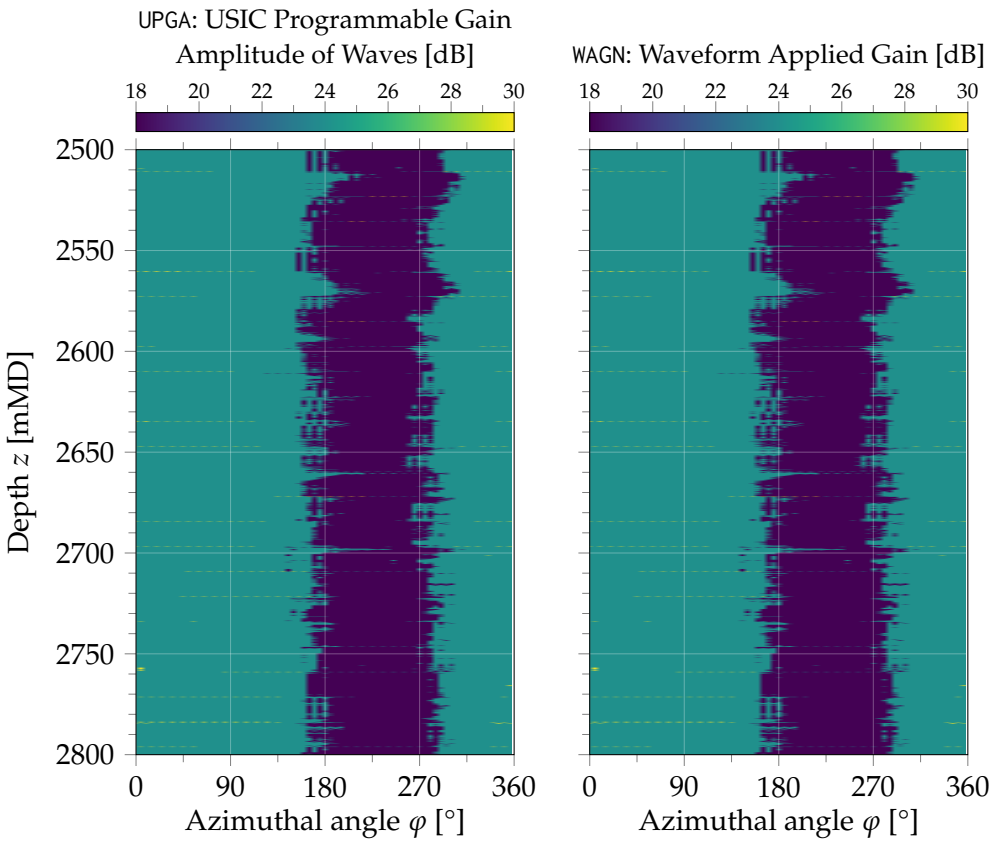

Figure 11: Two gain-related channels from the ultrasonic tool

units of dB, and a dimension of 72, i.e. one value per waveform at each depth. Figure 11 shows that the two channels look identical, and that both have bands of low gain throughout the third fourth. In fact, these two channels are numerically identical to each other in every log file in the Volve Data Village dataset except for the logs from well F-12.

We can try using these channels to re-equalise the amplitudes of the waveforms. We choose to use `WAGN`, as 'waveform applied gain' sounds exactly like what we are trying to compensate for. We convert the decibel gain into a scaling factor and calculate the corrected waveforms as

$$u'(z_i, \varphi_j, k) = u(z_i, \varphi_j, k)\ 10^{-\texttt{WAGN}(i,j)/20}. \tag{10}$$

The right plot in Figure 10 shows the corrected waveforms. We can see that this re-equalisation has removed the band of weaker waveforms, so that we no longer have any sudden transitions in the waveform amplitude. Testing this re-equalisation on the logs from well F-12 using both `WAGN` and `UPGA` shows that `WAGN` gives a better re-equalisation.

In summary, the USIT has 72 (if $\Delta\varphi = 5°$) or 36 (if $\Delta\varphi = 10°$) channels `U001`, `U002`, . . . that store the ultrasonic waveforms as (9) shows. Different amounts of gain may have been applied to the different waveforms before they were stored, and we may correct for this using the `WAGN` channel as (10) shows.

3.2.3 Time axis of ultrasonic waveforms

Even though we now have all of the ultrasonic waveform data in the file, we are in a similar situation as we were in for the sonic waveforms: We do not know what the time

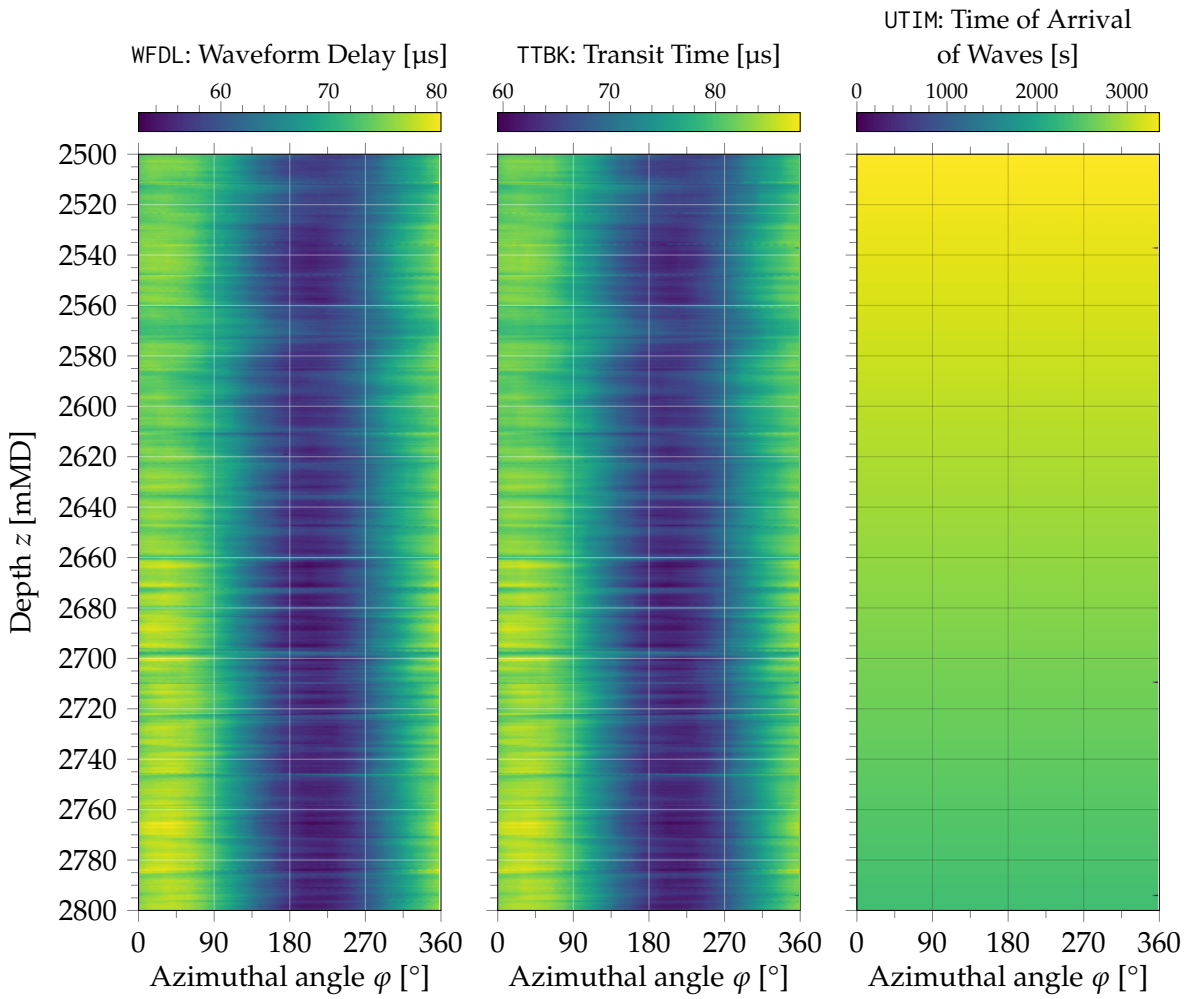

Figure 12: Various waveform time related channels from the ultrasonic tool. (As we only show an extract from the upper part of the log, the values of UTIM are somewhat unrepresentative. At the bottom of the log, $z = 3185$ mMD, UTIM starts at 846 s and increases with decreasing depth.)

axis of each waveform is.

The USFR ('Ultrasonic Sampling Frequency') parameter from Table 8 would seem like a promising first step, but its values throughout the log files in Table 2 (500 kHz, 666 kHz, and even 0 kHz) are much too low when we know from the literature that the ultrasonic pulses can carry significant energy from 200–700 kHz [12, 20]. Instead, we can look more closely at Table 9, where we find three depth-by-azimuth channels with time units, namely WFDL ('Waveform Delay'), TTBK ('Transit Time'), and UTIM ('Time of Arrival of Waves'). Figure 12 shows these channels in the subinterval 2500–2800 mMD.

First, let us look closer at the simplest-looking channel, UTIM. It increases from a low value at the bottom of the well (where the logging tool starts) to a high value at the top of the well (where the tool ends). As such, it might represent the times, relative to some initial reference time, at which each waveform was measured. If that is the case, the first and last values of UTIM tell us that it took the USIT 2534.9 s to go from the bottom to the top of the measured interval. We can compare this to other time measures in the log file. While there is no time index channel in the file, the CS ('Cable Speed') channel gives us the speed at which the tools are moving through the well. The length of the depth interval of frame 60B and the mean cable speed $\mathrm{mean}_i[\mathtt{CS}(i)] = 0.2805\,\mathrm{m/s}$ tell us that it

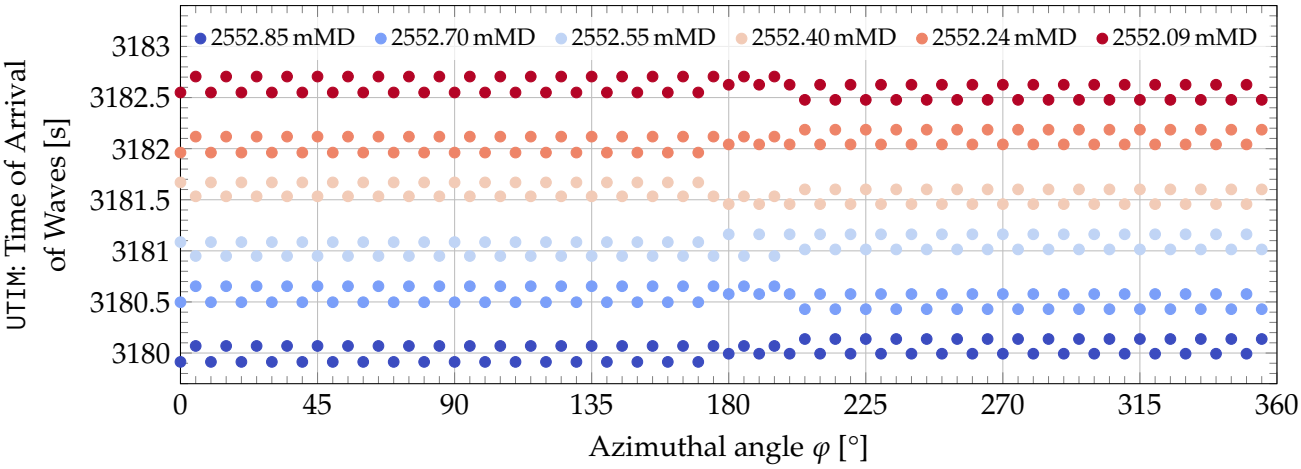

Figure 13: Values of the UTIM channel against azimuthal angle at a selection of depths z_i denoted by colours

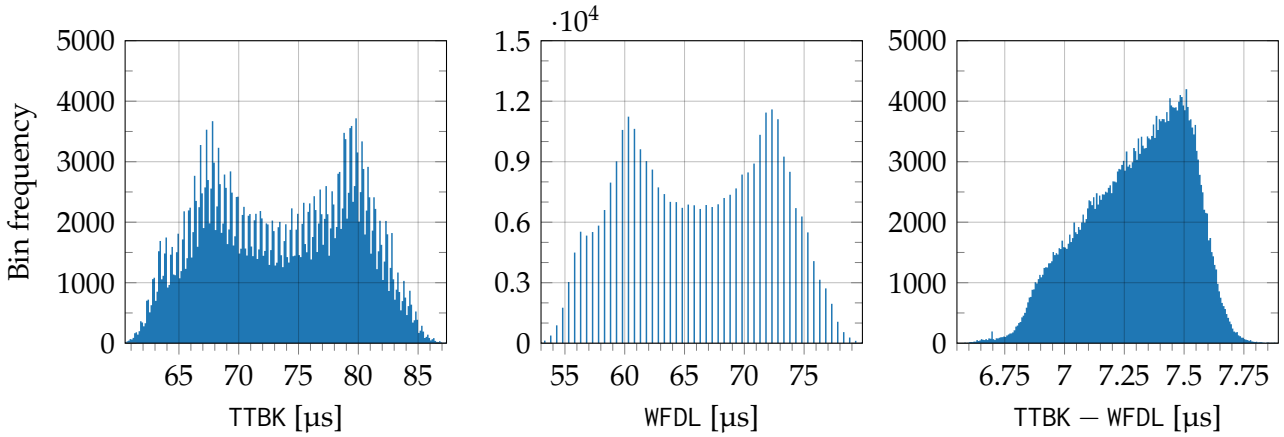

Figure 14: Histograms of the values from the TTBK ('Transit Time') and WFDL ('Waveform Delay') channels, and of the sample-by-sample difference of these values

took the USIT 2531.9 s to go from bottom to top. These two estimates match to 3 s, which supports the interpretation of UTIM as the time at which each waveform was measured.

While UTIM does not bring us any closer to finding the waveforms' time axes, we can still take a closer look at when the different measurements at different depths were made. Figure 13 indicates that the tool measures at every second azimuth over one rotation, and then fills in the remaining azimuths in a later rotation.

Next, let us look at TTBK ('Transit Time'). From its description, it probably corresponds to the 'travel time' defined in [20] as the time between the emission and the reception of the pulse's envelope peak.[10] Thus, it should represent the time the pulse takes from the transducer to the casing and back. This implies that the differences between high and low TTBK values at the same depth come from the tool being eccentered: The travel time is shortest when the transducer is pointing in the direction of eccentering, and longest when it is pointing against that direction. This also implies that the waveforms in the U001, U002, . . . channels have been shifted to align their peaks.

As the WFDL ('Waveform Delay') channel has the same overall shape as TTBK, but with

[10]While [20] describes a fast approximate method to find this peak, [21] describes a more accurate approach that we will follow here.

ISBN 978-82-8123-020-0

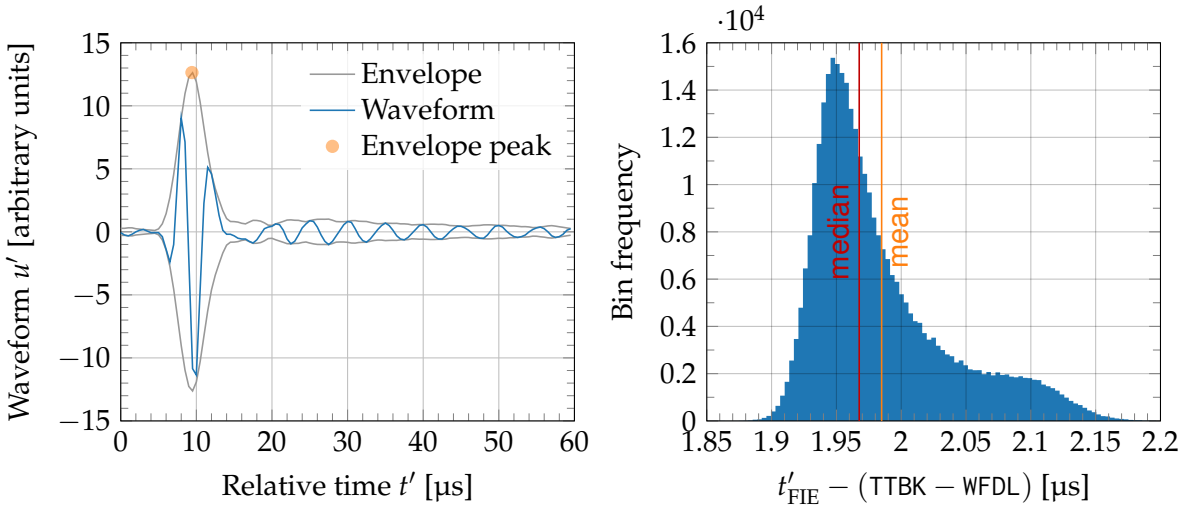

Figure 15: *Left:* The waveform $u'(2552.55\,\mathrm{mMD}, 0°, k)$, its envelope, and its peak at $t'_{\mathrm{FIE}}(2552.55\,\mathrm{mMD}, 0°)$. *Right:* Histogram of the difference $t'_{\mathrm{FIE}}(z_i, \varphi_j) - [\mathtt{TTBK}(i,j) - \mathtt{WFDL}(i,j)]$ over all i and j.

overall lower values, then perhaps it represents the time of the first waveform sample? We can investigate this through the histograms in Figure 14. They show that the difference between TTBK and WFDL is almost always 6.75–7.75 µs.

However, the most interesting result in Figure 14 is the histogram of WFDL values. It indicates that the values of WFDL are quantised, with a constant step separating them. Further investigation reveals this step to be 0.5 µs. If the waveforms in the file are all 120 samples long and aligned to each other by their peaks, this implies that longer waveforms were originally recorded, and that the waveforms in the file are shorter time windows of these original waveforms. If that is the case, and these time windows start and end exactly at a sample so that no interpolation is necessary, then the step between WFDL values should represent the sampling period $T_\mathrm{s} = 0.5\,\mu\mathrm{s}$ of the waveforms. This gives us a sampling rate $F_\mathrm{s} = 1/T_\mathrm{s} = 2\,\mathrm{MHz}$. We can now find a relative time axis, giving the time since the first sample, as

$$t'_{\mathrm{US}}(k) = 0.5k\,\mu\mathrm{s} \quad \text{for each sample} \quad k \in [0\,..\,(\mathtt{NPPW} - 1)]. \tag{11}$$

Figure 10 shows this relative time axis.

We have thus assumed that WFDL represents the time of the first sample in each waveform, that TTBK represents the arrival time of the waveform envelope peak, and that the waveforms are sampled at $F_\mathrm{s} = 2\,\mathrm{MHz}$. We can test these assumptions against each other, by comparing, over all waveforms, the time difference TTBK − WFDL with the waveform's time difference between the first sample and the pulse peak arrival. (The latter time difference is equivalent to the relative time t'_{US} at which the pulse peak arrives.) We find the pulse arrival time via the peak of the waveform envelope, which is found as the magnitude of the analytic waveform signal found through the Hilbert transform. We then estimate the location of the envelope peak more exactly through a quadratic polynomial fit.

Figure 15 shows the relative peak arrival time t'_{FIE} for one waveform, and a histogram of the difference $t'_{\mathrm{FIE}}(z_i, \varphi_j) - [\mathtt{TTBK}(i,j) - \mathtt{WFDL}(i,j)]$ over all i and j. The histogram shows differences close to 2 µs. Unless our assumptions about the meanings of the WFDL and/or

ISBN 978-82-8123-020-0

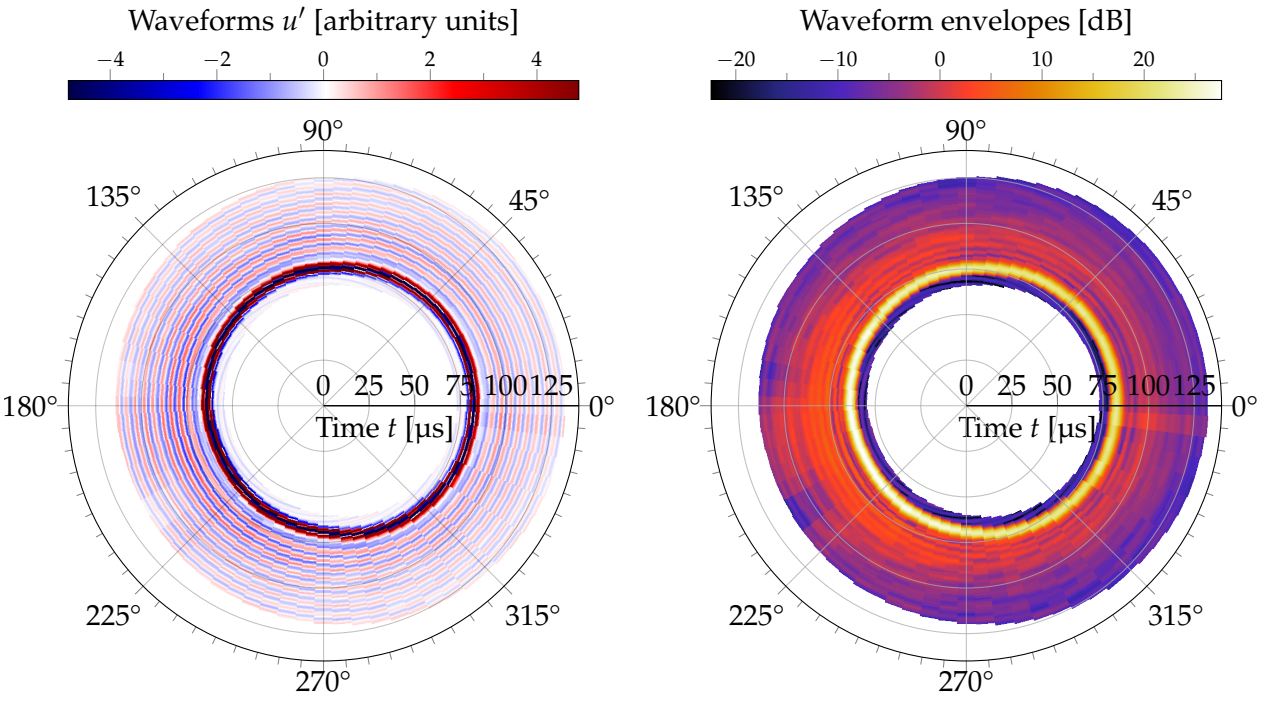

Figure 16: *Left:* Polar plot of waveforms $u'(2552.55\,\text{mMD}, \varphi_j, k)$ against time given by (12). *Right:* Envelopes of the waveforms to the left, in decibels.

TTBK channels are incorrect, this means that WFDL and TTBK use different reference times $t = 0$. This would mean that the values from one needs to be offset to be directly comparable to the values from the other.

In fact, Table 8 shows that there is a parameter USTO $= -2\,\mu\text{s}$ ('USIT Time Offset') that matches the time offset that Figure 15 implies. Thus, we may use USTO to offset either WFDL or TTBK so that their reference times match. As we have assumed TTBK's reference $t = 0$ to be time of emission of the pulse peak and WFDL does not have an obvious reference, we choose to offset WFDL as WFDL + USTO. After this offset of 2 µs, Figure 15 shows that we would get differences much smaller than the sampling period of 0.5 µs. While the source of these differences is uncertain, they might simply stem from the USIT's processing having a different method to estimate the time of the envelope peak than the method used here.

To sum this up, we can find a time axis for any USIT waveform $u(z_i, \varphi_j, k)$ as

$$t_{\text{US}}(z_i, \varphi_j, k) = \texttt{WFDL}(i, j) + \texttt{USTO} + 0.5k\,\mu\text{s} \quad \text{for each sample} \quad k \in [0\,..\,(\texttt{NPPW} - 1)]\,. \tag{12}$$

If the assumptions made here are correct, then $t_{\text{US}} = 0$ represents the time at which the pulse peak was emitted.

Now that we have time axes for every waveform, we can plot the waveforms against time. Figure 16 shows the waveforms at a single depth, and their envelopes, against angle and time. From it, we can see how the waveforms arrive later in certain directions because the tool is eccentered relative to the casing.

In summary, the waveforms were stored at a sampling rate of $F_{\text{s}} = 2\,\text{MHz}$ and realigned according to their FIE pulse arrival before being stored in the waveform channels. This means that we need to calculate separate time axes $t_{\text{US}}(z_i, \varphi_j, k)$ for every waveform, as (12) specifies.

ISBN 978-82-8123-020-0

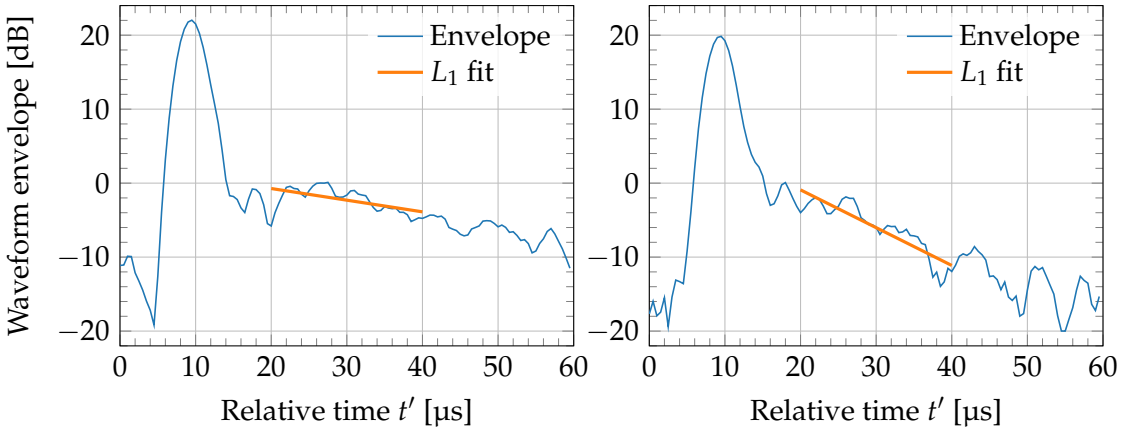

Figure 17: *Left:* Envelope in dB for $u'(2552.55\,\text{mMD}, 0°, k)$ (fluid behind casing). *Right:* Envelope in dB for $u'(2700.07\,\text{mMD}, 0°, k)$ (solid behind casing). Both plots also show fits giving the envelope decay rate L_1.

3.2.4 An example of ultrasonic waveform reprocessing

The ultrasonic waveforms provided in the log files may be reprocessed to provide information about e.g. the material behind the casing and the casing thickness. We will show a very simple example of such reprocessing, to give an idea of what is possible.

As we mentioned above, the tail of the waveforms come from the pulse's decaying resonance inside the casing. From the literature [20], we know that this decay should be at least approximately exponential, and that the decay is faster with higher-impedance materials behind the casing. If the envelope is decaying approximately exponentially, the decay will be approximately linear on a logarithmic scale, as Figure 17 shows.

The impedance behind the casing that the `AIBK` ('Acoustic Impedance') channel provides has been found through the aforementioned T^3 algorithm, which is somewhat complex. Instead of that algorithm, we take a look at the simpler alternative approach of [23], where information about the material behind the casing is found through a linear fit of the decay in the logarithmic envelope. This fit gives us a decay rate L_1 in dB/µs, which we can then compare against the impedance in the `AIBK` channel.

We calculate L_1 for every waveform in the investigated file, based on a fit of the envelope tail in $t'_{\text{US}} \in$ 20–40 µs, as Figure 17 shows.[11] Figure 18 compares the result L_1 against `AIBK`. From it, we can see that there is a good overall visual match between the two; areas of low and high impedance largely correspond to each other. However, galaxy patterns are more visible in L_1, especially around 180° in the 2500–2570 mMD interval. Such galaxy patterns indicate echoes from a third interface beyond the annulus [12, 20], which in this interval would be the surrounding 14 in casing mentioned at the start of Section 3. The T^3 analysis may be less susceptible to these galaxy patterns because it uses time windowing to emphasise the early part of the resonance tail, before third interface echoes typically arrive. (In [23], L_1 was applied in lab measurements without nearby third interfaces, so that such echoes were not an issue.)

Additionally, L_1 seems to be more affected by eccentering: Around $\varphi = 0°$, where

[11] This choice of time range was made by weighing three factors: First, the range should be as wide as possible to ensure a good fit. Second, the range should not start before the first interface echo is over and done interfering with the resonance tail, as Figure 15 shows. Third, the range should not end too late, as it may then cover more third interface echoes or even noise if the envelope decays very quickly.

ISBN 978-82-8123-020-0

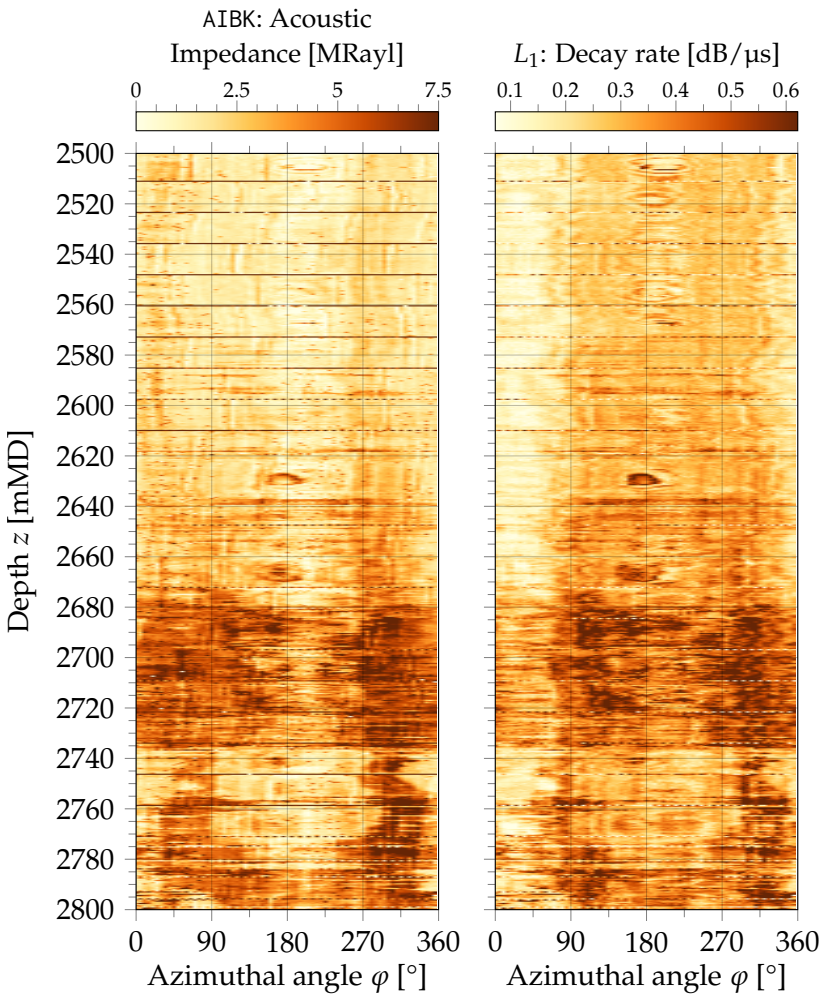

Figure 18: Comparison of two methods to estimate what's behind the casing. *Left:* The T^3 algorithm from [20, 21]. *Right:* The decay rate L_1 from [23].

Figure 16 shows that the tool is further away from the casing, L_1 is generally lower than around $\varphi = 180°$. The AIBK channel's acoustic impedances Z, on the other hand, are more uniform around the upper part of the log. That is what we would expect with fluids behind the casing, which the low impedances and decay rates tell us that we almost certainly have at these depths. Thus, the L_1 method seems to be more susceptible to the effects of eccentering.

While AIBK and L_1 represent different quantities (impedance Z in MRayl and the decay rate in dB/µs, respectively), we can expect these quantities to be correlated. Figure 19 shows the joint distribution of the two quantities for this file. Furthermore, we quantify the correlation by two measures: Pearson's $r = 0.457$ quantifies the linear correlation between the quantities, while Spearman's $\rho = 0.634$ quantifies their rank correlation. Together, the two measures show that there is a clear positive correlation between the quantities that is not quite linear, although their relationship is far from one-to-one.

This comparison between Z and L_1 is just a simple example of what we can do with the waveforms in the file. We could also use other, more complex processing methods to analyse these waveforms and compare with the processing results provided in the file. However, one challenge with this approach is that we do not have access to a ground truth that we can use to objectively evaluate the quality of processing methods. All

ISBN 978-82-8123-020-0

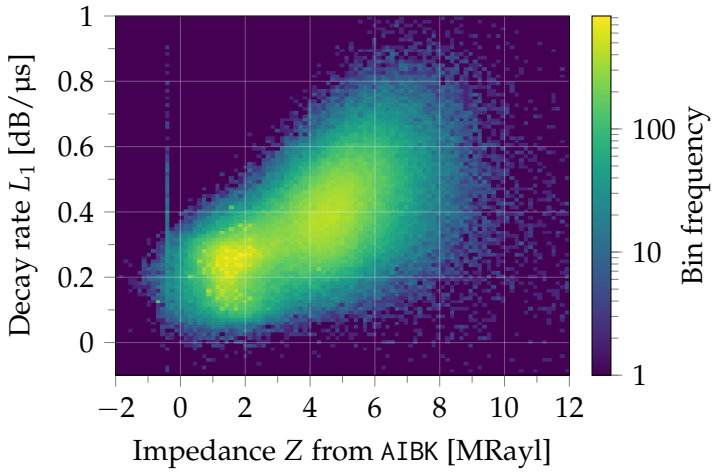

Figure 19: 2D histogram showing the joint distribution of the impedances Z given in the AIBK channel and the computed decay rates L_1.

we have is the measured waveforms and one processing method's estimates of what the waveforms imply. Therefore, we can only make relatively qualitative evaluations of methods' quality with field data like this.

4 Conclusion

This article, and its accompanying Jupyter Notebook [3], shows how we can use the dlisio Python library [7, 24] to programmatically access, explore, investigate, and process data from DLIS (RP66v1) files. dlisio makes these tasks about as straightforward as they can be, considering the complexities of the file format. The library makes the data in the DLIS file directly accessible, with no loss of numerical precision or any need to dump the file contents to another format.[12] It is also fast, free, open-sourced under the LGPLv3 license [9], and easily available.

While getting hold of well log files to work with has previously been a problem, the Volve Data Village dataset [2] makes a large number of well log files available, including 607 DLIS files. With data and a library to read it available, the remaining challenge is to understand what the different pieces of data represent well enough to be able to use them correctly. As each tool measures, processes, and stores its data in its own way, this challenge must be tackled individually for every tool.

In this article, we have investigated a sonic tool (DSLT) and an ultrasonic tool (USIT), which are extensively used in the well integrity logs in the Volve Data Village dataset, using one log file in particular as an example. We have identified a number of fundamental and important channels and parameters, investigated and compared them to confirm our understanding of what they represent, and demonstrated reprocessing of some of the data: For the sonic tool, we showed how waveforms can be reprocessed to compute CBL and bond index curves matching the ones provided in the file, as well as computing attenuations and bond indices not provided in the file. As an example of waveform reprocessing for the ultrasonic tool, we used a simple algorithm to identify areas of low

[12] Converting data to another format such as HDF5 may still be a good idea for purposes that require fast random access to data, especially if it is spread across a number of DLIS disk files, as the structure of DLIS files makes this difficult. Such purposes include machine learning on well log data.

ISBN 978-82-8123-020-0

and high impedance behind the casing, which gave a decent correspondence with the estimated behind-casing impedance stored in the file.

While these investigations were carried out for two specific acoustic tools, the general approach used here can be applied to investigate other tools when there is not sufficient documentation available on how their data is stored:

1. Read the available scientific and technical literature on the tool to gain an understanding of the physical and signal processing principles that it uses.
2. Use dlisio to isolate the parameters and channels belonging to the tool.
3. Based on the understanding of the tool gained from the literature, identify which parameters and channels are the most fundamental and/or important, and formulate hypotheses on what they represent.
    - Log plots of data from the tool (which the Volve Data Village dataset contains for the integrity logs) can help here, as they typically focus on the most important channels.
    - When looking for a channel or parameter providing a particular piece of data, it can be helpful to restrict the search to channels or parameters with appropriate units or dimensions. For example, a search for a particular channel related to time could be restricted to channels with time units (e.g. s, ms, µs), and a search for a channel with, say, N azimuthal measurements per depth could be restricted to channels with dimension N.
4. Test your hypotheses of what the channels represent by closely investigating their content, comparing them against other related channels and parameters, and/or using channels to recompute results stored in other channels. Such comparisons and recomputations can reveal incorrect hypotheses. New and better hypotheses then need to be formulated and tested until the tool data is sufficiently understood.

We, the authors of this article, hope that it and its companion Jupyter Notebook will stimulate research into well logging. We hope that they give other researchers a head start on their work by demonstrating the data and software tools that are available, and how to use them to get to work on the data that the researchers are interested in.

Acknowledgements

The authors thank Equinor AS, the former Volve license partners ExxonMobil Exploration and Production Norway AS and Bayerngas (now Spirit Energy) for permission to use the Volve dataset. They also thank Bjarne Rosvoll Bøklepp for his helpful comments on an early draft of the article.

Appendix

The investigated log file, namely file 2 from well F-11 B, contains many parameters and channels. Tables 6 and 7 provide the sonic tool's parameters and channels, respectively, while Tables 8 and 9 provide the ultrasonic tool's parameters and channels, respectively.

Table 6: DSLT parameters in the investigated log file

Name	Long name	Value	Units
AMSG	Auxiliary Minimum Sliding Gate	180.0	µs
BILI	Bond Index Level for Zone Isolation	0.8	
CBLG	CBL Gate Width	45.0	µs
CBRA	CBL LQC Reference Amplitude in Free Pipe	53.0	mV
CMCF	CBL Cement Type Compensation Factor	0.679	
DDEL	Digitizing Delay	0.0	µs
DETE	Delta-T Detection	E1	
DSIN	Digitizer Sample Interval	10.0	µs
DTCM	Delta-T Computation Mode	FULL	
DTFS	DSLT Telemetry Frame Size	536	
DWCO	Digitizer Word Count	250	
ENABLED	Equipment or Computation Acquisition Status	Yes	
FATT	Acoustic Attenuation due to Fluid	0.0	dB/m
FCF	CBL Fluid Compensation Factor	1.0	
GOBO	Good Bond	6.259	mV
ITTS	Integrated Transit Time Source	DT	
MAHTR	Manual High Threshold Reference for first arrival detection	120	
MATT	Maximum Attenuation	23.393	dB/m
MAX_TOOL_SPEED	Maximum service speed allowed for, or attained by, a logging tool.	1371.6	m/h
MCI	Minimum Cemented Interval for Isolation	4.515	m
MNHTR	Minimum High Threshold Reference for first arrival detection	100	
MODE	Sonic Firing Mode (e.g. DDBHC = Depth-Derived BHC, STRA = Single Transmitter, SREV = Single Receiver	CBL	
MSA	Minimum Sonic Amplitude	3.669	mV
NMSG	Near Minimum Sliding Gate	344	µs
NMXG	Near Maximum Sliding Gate	1026	µs
NUMP	Number of Detection Passes	2	
RATE	Firing Rate	R15	
SDTH	Switch Down Threshold	20000	
SFAF	Sonic Formation Attenuation Factor	10.663	dB/m
SGAD	Sliding Gate Status	OFF	
SGAI	Selectable Acquisition Gain	1X	
SGCL	Sliding Gate Closing Delta-T	130	µs/ft
SGCW	Sliding Gate Closing Width	25	µs
SGDT	Sliding Gate Delta-T	57	µs/ft
SGW	Sliding Gate Width	110	µs
SLEV	Signal Level for AGC	5000.0	mV
SPSO	Sonic Porosity Source	DT	
SUTH	Switch Up Threshold	1000	
VDLG	VDL Manual Gain	5.0	
WMOD	Waveform Firing Mode	FULL	
ZCMT	Acoustic Impedance of Cement	5.6	MRayl

Table 7: DSLT channels in the investigated log file

Name	Long name	Units	Dimension	Frame
BI	Bond Index		1	60B
CBL	CBL Amplitude	mV	1	60B
CBLF	CBL Amplitude (Fluid Compensated)	mV	1	60B
CBSL	CBL Amplitude (Sliding Gate)	mV	1	60B
CMCG	CBL Cement Type Compensation Gain		1	60B
TT	Transit Time for CBL	µs	1	60B

ISBN 978-82-8123-020-0

Table 7: (continued)

Name	Long name	Units	Dimension	Frame
TT1	Transit Time 1	µs	1	20B
TT2	Transit Time 2	µs	1	20B
TT2S	Transit Time 2 Secondary	µs	1	20B
TTSL	Transit Time (Sliding Gate)	µs	1	60B
VDL	Variable Density Log		500	20B
WDE1	Waveform Delay 1	µs	1	20B
WDE2	Waveform Delay 2	µs	1	20B
WF1	Waveform 1		250	20B
WF1N	Waveform 1 Normalization Factor		1	20B
WF2	Waveform 2		250	20B
WF2N	Waveform 2 Normalization Factor		1	20B

Table 8: USIT parameters in the investigated log file (abridged)

Name	Long name	Value	Units
AFVU	Automatic Fluid Velocity Update	Off	
AGMN	Minimum Gain of Cartridge	-12	dB
AGMX	Maximum Gain of Cartridge	40	dB
BERJ	Bad Echo Rejection	ON	
CMTY	Cement Type	Regular Cement	
C_ALGO	Collar detection algorithm	CCLU	
C_BLANK	Ignore on each side of each collar, inches	12	
C_DLEN	Collar detection length, inches	24	
C_JNO	Joint number offset	0	
C_MFILT	Apply 3-point median filter to statistics?	No	
C_MPL	Minimum pipe length, inches	24	
C_NUP	Number joints from bottom?	Yes	
C_TPC	Collar detection threshold percentage	50	
C_WIND	Collar detection window length, inches	600	
DOT	Diameter of Transducer Sensor	4.874	in
EMXV	EMEX Voltage	90	V
HRES	Horizontal Resolution	5 deg	
MAX_TOOL_SPEED	Maximum service speed allowed for, or attained by, a logging tool.	2057.4	m/h
NPPW	Number of Points Per Waveform	120	
NWPD	Number of Waveforms per Depth	72	
OPLEV	USIT Remove Flagged Data Level	OPT2	
RCOD	Reference Calibrator Outer Diameter	7.0	in
RCSO	Reference Calibrator Standoff	1.37795	in
RCTH	Reference Calibrator Thickness	0.2952	in
SDNV	Number of Vertical Samples used for Micro-debonding Computation	5	
SDTHOR	Acoustic Impedance STD Horizontal Threshold for Micro-debonding	0.5	
SDTVER	Acoustic Impedance STD Vertical Threshold for Micro-debonding	0.3	
SUBT	USIT Sub type	10INC	
TCUB	T^3 Processing Level	VXLP	
THDH	Maximum Search Thickness (percentage of nominal)	130.0	%
THDL	Minimum Search Thickness (percentage of nominal)	70.0	%
THDP	Thickness Detection Policy	Fundamental	
TVD	True Vertical Depth	0.0	
U-USIT_DDT5	USIC Downhole Decimation for T5 only	0_NONE	

ISBN 978-82-8123-020-0

Table 8: (continued)

Name	Long name	Value	Units
ULOG	Logging Objective	MEASURE	
UMAO	USIT Measurement Angular Offset	18.0	deg
UMFR	Modulation Frequency	333333	Hz
UPAT	Emission Pattern	300K	
USFR	Ultrasonic Sampling Frequency	500000	Hz
USTO	USIT Time Offset	-2.0	µs
USUB	USIT Sub Identifier	10INC	
UWKM	Working Mode	D606005L	
VCAS	Ultrasonic Transversal Velocity in Casing	51.4	µs/ft
VRES	Vertical Resolution	6.0INCH	
WINB	Window Begin Time	53.795	µs
WINE	Window End Time	120.393	µs
WLEN	T^3 Processing Length	32.327	µs
ZCAS	Acoustic Impedance of Casing	46.25	MRayl
ZINI	Initial Estimate of Cement Impedance	-1.0	MRayl
ZTCM	Acoustic Impedance Threshold for Cement	2.6	MRayl
ZTGS	Acoustic Impedance Threshold for Gas	0.3	MRayl

Table 9: USIT channels in the investigated log file (abridged). We use ... to represent abridged channels in long series of similar channels.

Name	Long name	Units	Dimension	Frame
AGMA	Maximum Allowed Electronic Gain	dB	1	60B
AGMI	Minimum Allowed Electronic Gain	dB	1	60B
AIAV	Acoustic Impedance Average	Mrayl	1	60B
AIBK	Acoustic Impedance	Mrayl	72	60B
AIMN	Acoustic Impedance Minimum	Mrayl	1	60B
AIMX	Acoustic Impedance Maximum	Mrayl	1	60B
AI_MICRO_DEBONDING_IMAGE	Acoustic Impedance With Micro-debonding Image	Mrayl	72	60B
AWAV	Amplitude of Wave Average	dB	1	60B
AWBK	Amplitude of Wave	dB	72	60B
AWMN	Amplitude of Wave Minimum	dB	1	60B
AWMX	Amplitude of Wave Maximum	dB	1	60B
AZEC	Azimuth of Eccentering	deg	1	60B
BPRE	Burst Pressure	psi	1	60B
CCLU	Casing Collar Locator Ultrasonic	in	1	60B
CEMR	Ratio of Cement Measurements to Total		1	60B
CFVL	Memorized Fluid Acoustic Slowness	µs/ft	1	60B
CZMD	Acoustic Impedance of Mud	Mrayl	1	60B
ECCE	Amplitude of Eccentering	in	1	60B
ERAV	External Radii Average	in	1	60B
ERBK	External Radii	in	72	60B
ERMN	External Radii Minimum	in	1	60B
ERMX	External Radii Maximum	in	1	60B
ERNO	Nominal Casing External Radius	in	1	60B
GASR	Ratio of Gas Measurements to Total		1	60B
GNMN	Waveform Gain Minimum	dB	1	60B
GNMX	Waveform Gain Maximum	dB	1	60B
HRTT	Histogram of Raw Transit Time Index	µs	180	60B
IDQC	Internal Diameter Quality Check	in	1	60B
IRAV	Internal Radius Averaged Value	in	1	60B
IRBK	Internal Radii Normalized	in	72	60B

ISBN 978-82-8123-020-0

Table 9: (continued)

Name	Long name	Units	Dimension	Frame
IRMN	Internal Radius Minimum Value	in	1	60B
IRMX	Internal Radius Maximum Value	in	1	60B
IRNO	Nominal Casing Internal Radius	in	1	60B
MDR	Micro-debonding Ratio		1	60B
MLOSS	Metal Loss	%	1	60B
RB	Relative Bearing	deg	1	60B
RB_USIT	USIT RB	deg	1	60B
RSAV	Motor Revolution Speed	c/s	1	60B
T1BK	Internal Radius Metal Loss	in	72	60B
T2AV	Thickness Average of Casing minus Nominal	in	1	60B
T2BK	Thickness of Casing minus Nominal	in	72	60B
THAV	Thickness Average Value	in	1	60B
THBK	Casing Thickness Normalized	in	72	60B
THMN	Thickness Minimum Value	in	1	60B
THMX	Thickness Maximum Value	in	1	60B
THNO	Nominal Casing Thickness	in	1	60B
TTAV	Transit time average	µs	1	60B
TTBK	Transit Time	µs	72	60B
TTMN	Transit Time Minimum Value	µs	1	60B
TTMX	Transit Time Maximum Value	µs	1	60B
U-ED1	External Diameter in USI mode	in	1	60B
. . .	…	…	…	. . .
U-ED36	External Diameter in USI mode	in	1	60B
U-ID1	Internal Diameter in USI mode	in	1	60B
. . .	…	…	…	. . .
U-ID36	Internal Diameter in USI mode	in	1	60B
U001	Waveform for Azimuth 01		120	60B
. . .	…		…	. . .
U072	Waveform for Azimuth 72		120	60B
UCAZ	Ultrasonic Azimuth	deg	1	60B
UDEP	USIT River Depth Index	m	1	60B
UFLG	USIT Processing Flags		72	60B
UFRT	Firing Rate	Hz	1	60B
UPGA	USIC Programmable Gain Amplitude of Waves	dB	72	60B
USBI	USIT Bond Index		1	60B
USGI	USIT Gas Index		1	60B
UTDL	USIC Tachometer Delay	µs	1	60B
UTIM	Time of Arrival of Waves	ms	72	60B
WAGN	Waveform Applied Gain	dB	72	60B
WDMN	Waveform Delay Minimum	µs	1	60B
WDMX	Waveform Delay Maximum	µs	1	60B
WFDL	Waveform Delay	µs	72	60B
WPKA	Waveform Peak Amplitude		72	60B

References

[1] Norwegian Petroleum Directorate, “DISKOS – The Norwegian national data repository for petroleum data.” [Online]. Available: https://www.diskos.no (Accessed 2020-04-04).

[2] Equinor, “Volve Data Village dataset,” 2018, released under a license based on CC

BY 4.0. [Online]. Available: https://data.equinor.com/ (Accessed 2020-04-04).

[3] E. M. Viggen, E. Hårstad, and J. Kvalsvik, "Getting started with acoustic well log data using dlisio," 2020. [Online]. Available: https://github.com/equinor/dlisio-notebooks/blob/master/acoustic.ipynb

[4] Schlumberger, "Cement bond logging tools product sheet." [Online]. Available: https://www.slb.com/-/media/files/production/product-sheet/cement-bond-logging-tools-ps.ashx (Accessed 2020-04-04).

[5] Schlumberger, "USI ultrasonic imager tool product sheet." [Online]. Available: https://www.slb.com/-/media/files/production/product-sheet/usi.ashx (Accessed 2020-04-04).

[6] "Petroleum Open Standards Consortium." [Online]. Available: https://www.energistics.org/ (Accessed 2020-04-04).

[7] "dlisio documentation." [Online]. Available: https://dlisio.readthedocs.io (Accessed 2020-04-04).

[8] T. E. Oliphant, *A guide to NumPy*. Trelgol Publishing USA, 2006.

[9] "GNU Lesser General Public License (LGPL) version 3," Free Software Foundation, Jun. 2007. [Online]. Available: http://www.gnu.org/licenses/lgpl.html

[10] M. Grosmangin, F. P. Kokesh, and P. Majani, "A sonic method for analyzing the quality of cementation of borehole casings," *Journal of Petroleum Technology*, vol. 13, no. 2, pp. 165–171, 1961.

[11] W. L. Anderson and T. Walker, "Research predicts improved cement bond evaluations with acoustic logs," *Journal of Petroleum Technology*, vol. 13, no. 11, pp. 1093–1097, 1961.

[12] M. Allouche, D. Guillot, A. J. Hayman, R. J. Butsch, and C. W. Morris, "Cement job evaluation," in *Well Cementing*, 2nd ed., E. B. Nelson and D. Guillot, Eds. Schlumberger, 2006, ch. 15.

[13] G. H. Pardue and R. L. Morris, "Cement bond log—A study of cement and casing variables," *Journal of Petroleum Technology*, vol. 15, no. 5, pp. 545–555, May 1963.

[14] H. Wang, G. Tao, and X. Shang, "Understanding acoustic methods for cement bond logging," *The Journal of the Acoustical Society of America*, vol. 139, no. 5, pp. 2407–2416, 2016.

[15] J. Jutten and P. Parcevaux, "Relationship between cement bond log output and borehole geometrical parameters," in *SPE/IADC Drilling Conference*. Society of Petroleum Engineers, 1987.

[16] M. Bellabarba, H. Bulte-Loyer, B. Froelich, S. Le Roy-Delage, R. van Kuijk, S. Zeroug, D. Guillot, N. Moroni, S. Pastor, and A. Zanchi, "Ensuring zonal isolation beyond the life of the well," *Oilfield Review*, vol. 20, no. 1, pp. 18–31, 2008.

ISBN 978-82-8123-020-0

[17] H. D. Brown, V. E. Grijalva, and L. L. Raymer, "New developments in sonic wave train display and analysis in cased holes," *The Log Analyst*, vol. 12, no. 1, pp. 27–40, Jan. 1971.

[18] H. Gai and C. F. Lockyear, "An acoustic method to evaluate cement quality in an oil well," in *International Conference on Acoustic Sensing and Imaging*. Institution of Electrical Engineers, Mar. 1993, pp. 120–125.

[19] H. Gai and C. F. Lockyear, "Cement bond logs - A new analysis to improve reliability," *SPE Advanced Technology Series*, vol. 2, no. 1, pp. 34–42, Mar. 1994.

[20] A. Hayman, R. Hutin, and P. Wright, "High-resolution cementation and corrosion imaging by ultrasound," in *SPWLA 32nd Annual Logging Symposium*, 1991, p. 25.

[21] P. Wright, "Method and apparatus for the acoustic investigation of a casing cemented in a borehole," US Patent 5,216,638, Jun., 1993.

[22] Schlumberger, "Curve mnemonic dictionary." [Online]. Available: https://www.apps.slb.com/cmd/index.aspx (Accessed 2020-04-04).

[23] T. Sirevaag, T. F. Johansen, I. Larsen, and R. M. Holt, "Laboratory setup for improved logging behind casing," in *SPWLA 58th Annual Logging Symposium*. London, UK: Society of Petrophysicists and Well Log Analysts, 2018, p. 16.

[24] "dlisio source code repository." [Online]. Available: https://github.com/equinor/dlisio (Accessed 2020-04-04).

ISBN 978-82-8123-020-0