HIGH DYNAMIC RANGE PLATFORM FOR OFDMA SIGNAL COLLECTION AND PROCESSING USING JOINT DETECTION

Sergey Dickey (Wavenetix Corp., Lovettsville, Virginia, USA; sergey.dickey@wavenetix.com)

ABSTRACT

Platform for signal collection, storage, and processing based on a Windows computer. Signals are processed by an original joint detection algorithm that achieves high simultaneous dynamic range of detection of OFDM signals. The paper describes the JD algorithm and the system that includes an integrated USRP SDR, GIS-enabled database -- postgresQL/postgis -- and tool for displaying results on a map. Representative results are provided and discussed. The system is intended for use as a high-dynamic range, inexpensive signal collection and processing platform for wireless networks T&M and R&D.

1. INTRODUCTION

This article describes a signal collection and processing tool for testing OFDMA-based 4G networks. This platform can be used as a powerful, inexpensive, and versatile alternative to proprietary signal collection tools used by wireless network operators and Government agencies for wireless network test and measurements. Its advantages include the use of general-purpose SDRs, PC laptops, conveniently integrated in a single easily installed and maintained package. For facilitating signal R&D work, an interface to Matlab/Simulink environment is also provided. This feature will help researchers migrate their algorithms from Matlab to the pre-configured multithreaded C++ environment that integrates signal storage and distribution to various processing tasks, a GIS database, and a data mapping component with access to unlimited free map data.

2. SYSTEM DESCRIPTION

Fig. 1 shows the high-level system diagram. The system uses USRP N200 SDR direct-conversion radio for collecting signal I and Q samples; the radio tunes in a wide range of 400 through 4000 MHz and has the bandwidth of 40 MHz with the RF preselector card shown in the diagram. As the daughter card does not include an RF roofing filter, an external band-pass filter is used to suppress RF interference under 2.6 GHz. The radio also includes a GPS receiver for

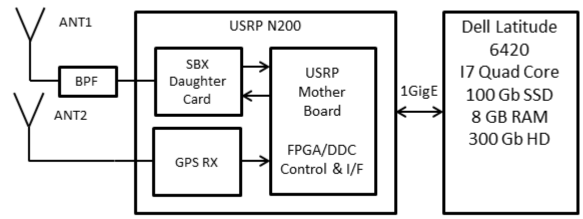

Figure 1: System diagram

providing time and frequency references, as well as geo location data.

In the current version of the tool, namely WiMAX scanning test receiver, the receiver digitizes one 10-MHz OFDMA channel at a time and provides a stream of signal complex samples at a constant rate of 25 Msps over a 1GigE interface. All further processing is done by the software running on the host laptop computer or (in the case of post processing of already acquired signals) on a server or server network.

The system works in the following two modes: (a) Acquisition mode (“ACQ”) and (b) post-processing mode (“POST”). In ACQ mode, used during signal collection, it stores signal samples continuously on the solid-state hard drive of the laptop. In addition, some limited signal processing takes place. In POST mode, the system reads signal samples from files and detects signals from various cellular base stations and measures their levels and shapes.

The two algorithms used for signal detection and measurement are the so-called “simple” mode processing and a more involved joint signal detection mode (“JD”).

Open-source PostgreSQL/postgis database is used to store all program control parameters and processed data.

The system includes a mapping subsystem based on the open-source Mapnik package.

3. SOFTWARE ARCHITECTURE

The software (codename: “Longear”) has been written in C++. Fig. 2 presents Longear’s internal architecture. Each task runs in a separate thread. Main task initializes and launches all other tasks and handles the flow of signal data in

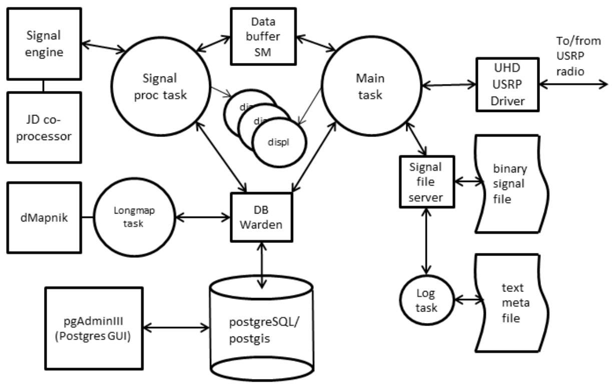

Figure 2: Software architecture

both ACQ and POST modes. Most tasks are interacting with the database, which is the center hub for configuration and signal data.

In ACQ mode, Main task handles the flow of signal data in discrete chunks called "buffers" from the USRP driver ("UHD"). Then it uses sig_proc task to synchronize with the signal and segment data into frames, each frame segment preceded by a header. Then Main task writes signal records thus generated into a formatted binary file, whereas GPS messages from the receiver, along with other system metadata (e.g., receiver gain) go into an associated text log file.

In POST mode, signal data are synchronized with the current position of the text log file by Log task; Main task serves signal buffers to sig_proc task using Data buffer State Machine as an atomic access interthread data handler.

The program uses a pool of several pre-allocated signal buffers, and data buffer state machine follows a simple protocol that results in no data being dropped.

Sig_proc task uses Signal engine for most of its operations. Signal engine works in several modes depending on parameters in the database configuration table. The principal modes are:

- "Simple" mode. Straightforward scanning of OFDMA signals in one or more segments. Synchronization with the signals by means of power burst detection over the signal buffer (typically 11 WiMAX frames) is fast and reliable in most cases.
- "Simple" mode with enabled signal timing search. This mode is more involved as it searches for "rogue" signals that may have their timing out of the acceptable range for TDD systems. Lack of basestation synchronization is, in our experience, a widespread occurrence in cellular networks that makes finding isolated cells a real challenge. This mode does not rely on all preambles being closely spaced in time and accordingly searches in the full acquisition frame window.
- Joint detection mode adds joint signal detection to Simple mode. Joint detector represents a separate processor class that works in tandem with Signal engine; it substantially increases the dynamic range of detection and the accuracy of (binned or averaged) signal level measurement. The downside, of course, is an appreciably longer processing time.

"Longmap" task runs autonomously taking its configuration parameters from the configuration table of the database. It continuously maps and updates signal measurement results that sig_proc put there.

Every task uses a single database gateway object, "DB warden," for thread-safe access to the database.

A separate application, pgAdminIII, provides the user or users with the full access to the database configuration and result tables. It allows the user to run queries, inspect histograms created by sig_proc, request signals to be displayed and mapped. Same capabilities are available to external tools that can connect to the database.

Three display windows present continuously updating signal envelope, correlations, and JD operation charts, as well as text messages. Graphical windows use direct2d technology from Microsoft. They run in their own threads.

4. SIGNAL PROCESSING ALGORITHMS

Most signal processing activities are performed by signal engine class that does signal search and frame synchronization. When necessary, it requests help from joint_detector class.

4.1. ACQ Mode

In this mode, sig_proc task receives buffers with signal I and Q samples from the receiver. Each time a buffer is received, the following processing steps are performed on a buffer:

- Frame synchronization. Typically uses signal power envelope to find frames, but does more processing in order to find preambles in difficult cases.
- Downsampling from 25 msps to 11.2 msps. • Buffer segmentation into frame segments with binary headers.
- Finding and applying proper scaling factors; storing scaling in the header.
- Automatic gain control procedures. Use UHD to request RX gain increase or decrease.
- Storing frame headers and corresponding frame segments in the output binary file.

4.1. POST Mode

Signal buffers come from the binary file described briefly earlier. A free-running Log task synchronizes reading positions in the binary and log files such that sig_proc task

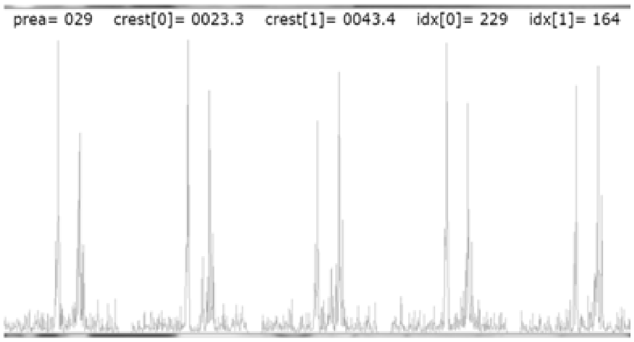

Figure 3: Signal correlation display

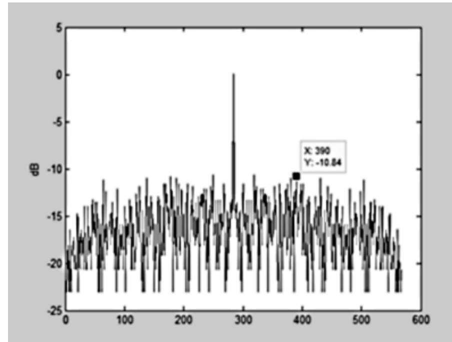

Figure 4: Autocorrelation function of P42

receives correct metadata together with signal buffers and headers.

Typically frame synchronization is performed for all of the frames in a buffer at once; using a long stretch of coherent signal for this purpose improves accuracy of this method. In some cases, different cells are not well frame-synchronized. In this case, signal engine can do a full search of a specific preamble over the full (configurable at the collection time) length of the frame in the file. The search involves full TD-FD processing of the OFDM signal while varying the FFT interval timing and finding the maximum resulting correlation.

After symbol timing has been established, the signal engine iterates over WiMAX frames and preambles. An FFT converts a preamble symbol (and interference and noise) to frequency domain where each subcarrier symbol is multiplied by the corresponding tabulated preamble symbol with the given index. After an inverse FFT, the correlation "trace" is used for signal detection. An example of a resulting correlation form is given in Fig. 4.

Each detected signal's results are written into a database table; in addition, a histogram for signal level difference distribution between the strongest and weakest signal received in the same frame is incrementally built over the course of processing and stored in the database.

5. JOINT SIGNAL DETECTION AND ITERATIVE INTERFERENCE CANCELLATION

Longear provides two versions of joint detection of signals from multiple cells. A good survey of relevant work can be found in [1]. One could possibly divide the work in this area into a more theoretical, "algebraic," approach, and more "heuristic," practicality-driven one. We call the former "algebraic" because it achieves its goal by solving a system of equations. The complexity of that approach, which grows exponentially with the number of signals and their lengths, gave rise to the "heuristic" approach, which is iterative interference cancellation.

5.1. Co-channel interference mitigation in RF test receivers

Apparently the first attempt to develop joint detection for just two different signals was made circa 1998, in respect to GSM signals [2].

The problem of increasing the dynamic range of GSM detection for the purpose of signal identification was more successfully addressed by DTI/PCTEL "Clarify" system [3] and, independently, by R&S GSM test receivers [4]. Both approaches used fixed, or known variable [5] patterns to identify signals by their "signatures" and timing. They exploited the processing gain provided by the patterns used (around 150 symbols) to improve recognition dynamic range.

The proliferation of CDMA-based technologies in the '00s diminished the importance of joint detection for signal identification, as those networks used tremendously long channelization codes that made identification an easy task – 30 dB and more were the numbers for the lowest signal-to-total power ratios specified. However, starting with the advent of HSPA, the size of spreading factors has been decreasing. Instead of 32K or 38K – long "short" codes common in CDMA and WCDMA, Rel '99, now all that was left was, for example, the 284-long preamble code in the 10-MHz WiMAX channel. Its processing gain is

$$G_{proc} = 10 \cdot \log_{10}(284) = 24.5 \text{ dB}.$$

Assuming that SINR of about 12 dB is needed to achieve reliable detection, the minimum SINR available in this case for signal identification is:

$$S_{min} = 12 - 24.5 = -12.5 \text{ dB}.$$

This is close to the practical dynamic range of a typical WiMAX scanner in the field, and this is not sufficient for propagation model calibration and other network optimization-related tasks.

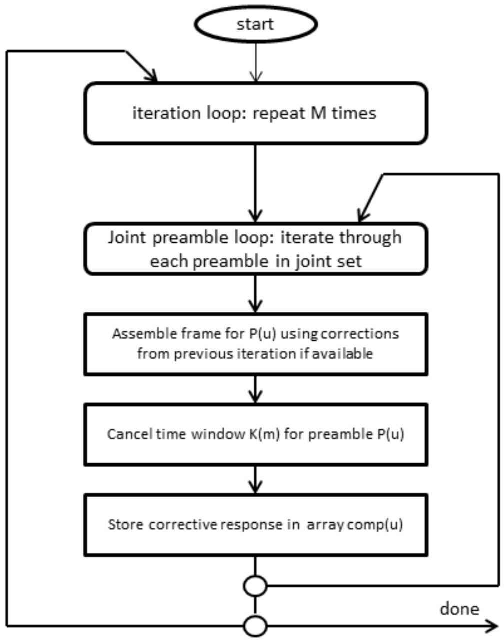

Figure 7: PIC algorithm flowchart

5.2. Reference pattern cross-correlation as dynamic range-limiting factor

We start with the autocorrelation function for one of the preamble codes (others will have similar autocorrelation functions). Figure 4 is the plot of the modulus of the autocorrelation function for preamble with index 42 (“P42”). From the chart, the autocorrelation “noise floor” lies at around -11 dB from the peak. Although autocorrelations are less relevant than cross-correlations to the problem of mutual interference, one should expect similar properties for cross-correlations.

It is helpful to estimate the dynamic range of the simple processor. Figure 5 shows a simulated, in Matlab, example of two preambles in the received signal (12 dB level difference). On the right, the weaker preamble is still detectable, but the margin (peak to interference floor) is merely 5 dB, which is the minimum needed for reliable identification, based on experience.

5.3. Parallel iterative interference cancellation

First release of the joint detection (“JD”) feature adopted the approach called “parallel iterative interference cancellation,” or “PIC” [6]. Figure 7 shows schematically the main steps of the algorithm. The inner loop, “preamble loop,” iterates over every “joint preamble,” that is a preamble index given in the list of preamble signals to be cancelled. For each of the joint preambles, a frame with all corrections for all joint

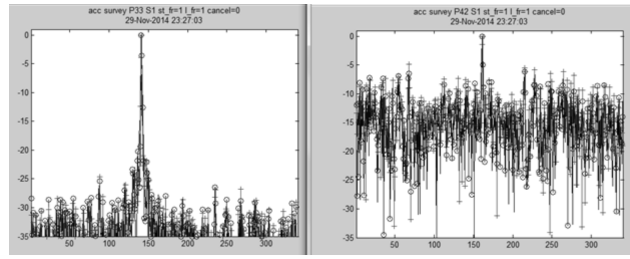

Figure 5: P33@0 dB, P42@-12 dB

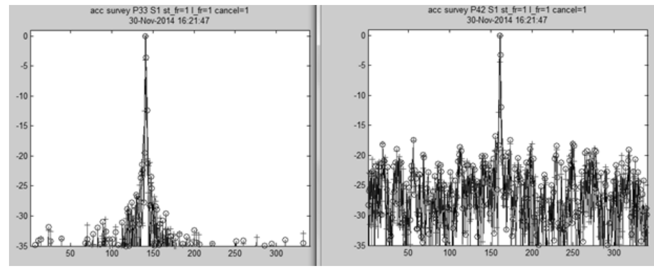

Figure 6: P33@0 dB, P42@-12 dB; PIC

preambles, except the correction for the preamble that is to be processed in this iteration of the preamble loop, is assembled.

The next step in this loop is the “cancellation” procedure.” Inside this block, the algorithm goes over all time-shift “chips” and iteratively finds for each such a corrective response, using a weighted and shifted copy of this preamble’s time-domain waveform, that minimizes the total error calculated over the cancellation window. Each of the time-shift partial corrective responses is added to the array that stores the full correction determined for the preamble.

When the preamble loop finishes, the process is repeated as the next algorithm’s iteration until the last one. The number of iterations is fixed and is set as a parameter.

5.4. Serial iterative interference cancellation

Parallel algorithm has been implemented and tested; it exhibited acceptable performance levels in the lab and in the field. However, it was discovered that its efficacy could be improved with a few rather commonsensical rearrangements. A similar and independently developed algorithm for CDMA signals is reported in [7].

The main idea of the parallel algorithm was to clean the whole time-shift response window in each iteration, using all available information obtained during preceding iterations. However, not all of the responses, calculated during last iteration, were correct, some obtained using dirty data. Because of that, the “corrections” actually might have the opposite effect, contributing “noise” energy to the signal and provoking more erroneous contributions in subsequent iterations.

The algorithm has been redesigned, resulting in the “serial iterative interference cancellation,” or “SIC.” The following list highlights the differences between the two versions of the algorithm.

- Instead of going through a fixed sequence of steps, the modified algorithm adapts to the instantaneous conditions observed at the beginning of an iteration.
- Instead of trying to clean the whole response window, the new algorithm minimizes the cleaning area, ideally cancelling a single (the strongest) time-shift response during each iteration.
- Since the rigid sequence has been eliminated, the iteration itself has been "atomized" to contain only a single-point cancellation for a single joint preamble. Accordingly, the number of iterations has increased, but each iteration is smaller and does not do useless and/or harmful work; as a result, convergence has improved.
- The frame assembly operation has also been streamlined: now all of the corrections from previous iterations are applied in all cases, even when the frame will be used to clean a joint preamble. Previously full cancellation at each step started from the original version for the selected preamble whereas all other joint preambles had been cleaned in that frame.

Table 1 lists iteration parameters for the parallel and serial versions of the algorithm. As can be seen from the table, the serial algorithm is substantially faster.

Table 1. Iteration parameters for PIC and SIC

Parameter	PIC	SIC	Comment
Number of iterations	4	12 - 64	Depends on the number of joint signals or how many of them are present in the air (2 – 6); also depends on CIR length. Set by user.
Number of preambles treated in each iteration	2 – 6	1	PIC: all preambles in the set even if not detected
Cancel window size, chips	16 , 16, 32, 34	5	PIC: different window size depending on iteration number
Total number of elementary (chip) cancellations per joint preamble	98	6 - 16	

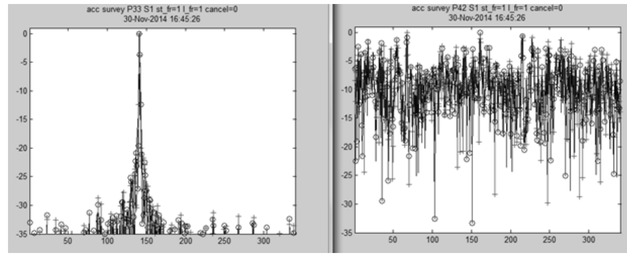

Figure 8: P42@-20 dB; simple processor

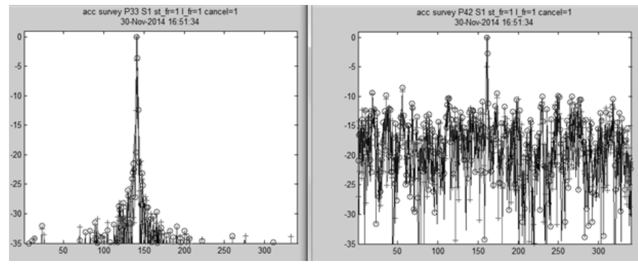

Figure 9: P42@-20 dB; PIC

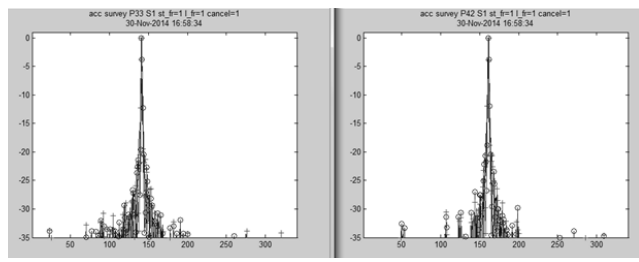

Figure 10: P42@-20 dB; SIC

6. PERFORMANCE TEST: SYNTHESIZED PREAMBLES

In this series of tests, a number of synthesized preambles were combined in a signal; then the signal was processed either without applying joint detection, or with PIC or SIC.

6.1 Two preambles; level difference at 12 dB

Two preambles, P33 at 0 dB (all levels are relative, the absolute level does not matter in this simulation), P42 at -12 dB (same as in Fig. 5). Fig. 6 shows the results of simulation when PIC algorithm was applied with both simulated preambles included in the joint set (two joint preambles in the set). Comparing this with Fig. 5, one can see that P42 is now more discernible.

6.2 Two preambles; level difference at 20 dB

Same two preambles, but now the level difference is 20 dB. Fig. 8 shows the result when no joint detection is used. P42 has been lost in noise. Fig. 9 shows how the use of PIC ameliorates the situation. P42 is now discernible, albeit with a modest margin. Finally, Fig 10 shows the result of using

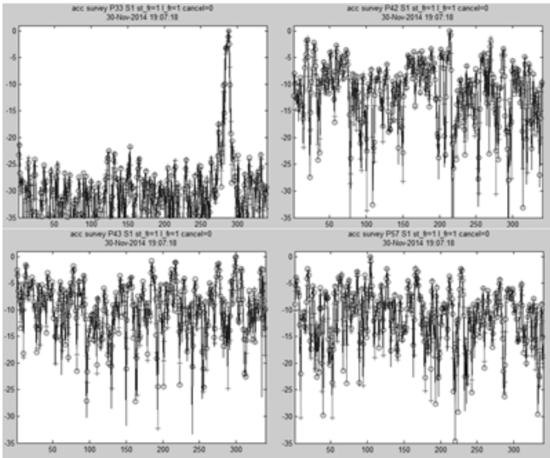

Figure 11: Germantown location, P33, P42, P43, P57; no JD

SIC in the same situation. There is no point in increasing further the level difference while applying SIC in this simulation, as SIC works perfectly when no real-life degradations are present.

7. FIELD PERFORMANCE TESTS

7.1. Car not moving, suburban location

In this particular location (Germantown, MD), there were four preambles present from cells that had been previously identified and geo-located; all from the same WiMAX segment (i.e. one of the three orthogonal preamble subcarrier sets). Matlab algorithm was used in this instance, as it can show more intermediate processing steps.

Fig.11 presents the result of processing without applying joint detection. Of four preambles sought only P42 is detectable. Fig. 12 shows same four preambles, from the same signal file, but processed with PIC (all four preambles were included in the joint set). All four are detectable based on the 5-dB criterion. Fig. 13 shows the results of using SIC to process data. The correlation traces are visibly cleaner, with deeper dynamic range.

7.2. Complete drive-test; Longear used to collect and process signals

This is the ultimate performance test, with the tool working in its production mode. The drive started in downtown Gaithersburg, MD and followed Rt355 and Quince Orchard Rd to Rt117; the area is predominantly urban, with high signal levels. As thermal noise lied well below the levels of cross-correlation interference, the dynamic range of reception was interference-limited and had a potential of deepening after applying JD methods.

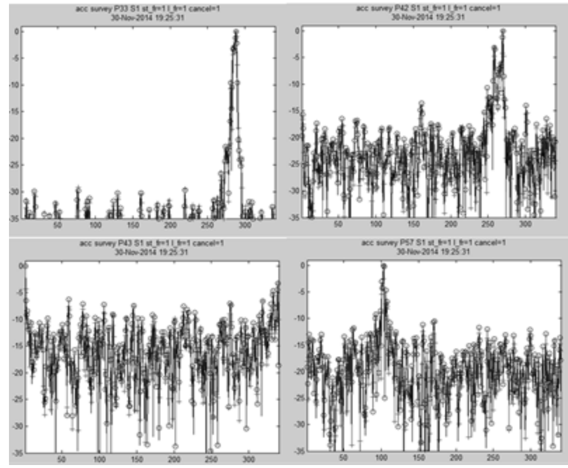

Figure 12: Germantown, P33, P42, P43, P57; PIC

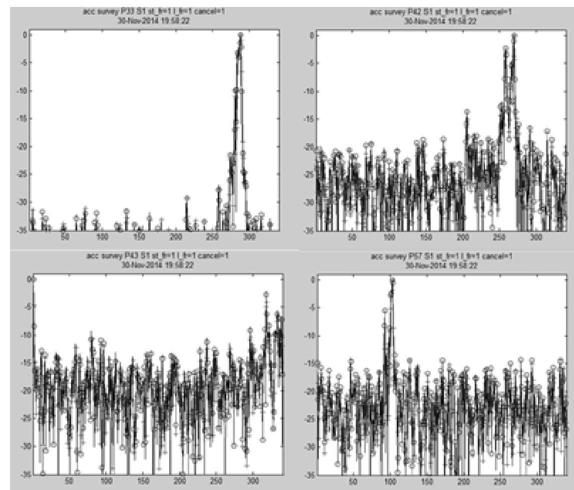

Figure 13: Germantown, P33, P42, P43, P57; SIC

Fig. 14 shows the map for P43, which is the weaker of the two signals in segment 1. The dominating signal is P57 (not shown). Scanner coverage for P43 is poor.

Fig 15 shows results for P43 for the same drive, but when SIC was applied. Coverage is much improved. Some remaining gaps in coverage were caused by insufficient sophistication of the algorithm when dealing with two correlation peaks that were coming from two different basestations in the area transmitting P43. The remaining interruption of coverage seemingly coincides with the part of the drive route where the scanner experienced a “hand-over” from one station to the other. This issue will be addressed in our further development of the tool.

Finally, Fig. 16 presents two histograms, for simple and SIC cases, generated by the tool in the course of processing. The increased dynamic range of detection of SIC processor is obvious; processing with SIC yields 84289 total measurement points for this drive whereas simple processing yields only 31097. The improved coverage in Fig. 15 is just

Figure 14: Drive-test, P43; simple processor

another aspect of the better dynamic range. Rough estimates for effective dynamic ranges of detection derived from these two charts are 12 dB for simple and 24 dB for JD processing.

8. CONCLUSION

This report describes the tool for OFDMA signal collection and processing developed by Wavenetix. The tool has an improved dynamic range of simultaneous signal detection under co-channel interference because of its iterative interference cancelling functionality. Another useful feature is its storing signal samples on a hard drive, which permits reproducing and analyzing signal propagation channels in the lab.

Further development work under way or in planning stages will produce more versions of the tool for 4G technologies, such as TD-LTE.

9. REFERENCES

[1] D. Divsalar, M.K. Simon, and D. Raphaeli, “Improved Parallel Interference Cancellation for CDMA,” *IEEE Trans. Commun.*, Vol. 46, No. 2, pp. 258-268, February 1998.

[2] Agilent Technologies E7475A GSM Drive-Test System. Product Overview, page 8. Downloaded from site: http://literature.cdn.keysight.com/litweb/pdf/5980-0439E.pdf

[3] S.L. Dickey, “Method and Apparatus for Co-channel Interference Measurements and Co-channel Components Separation…,” US Pat. 7,013,113.

[4] “New Solution with a Tried-and-Tested Platform,” News from Rohde & Schwarz, Number 186, pp 5-6 (2005/II).

[5] S.L. Dickey, “Use of SCH Bursts for GSM Co-channel Interference Measurements,” US Pat. 7,639,985.

[6] A.F. Molisch, *Wireless Communications*, John Wiley & Sons, Chichester, West Sussex, PO198SQ, UK, 2011.

Figure 15: Drive-test, P43, SIC processor

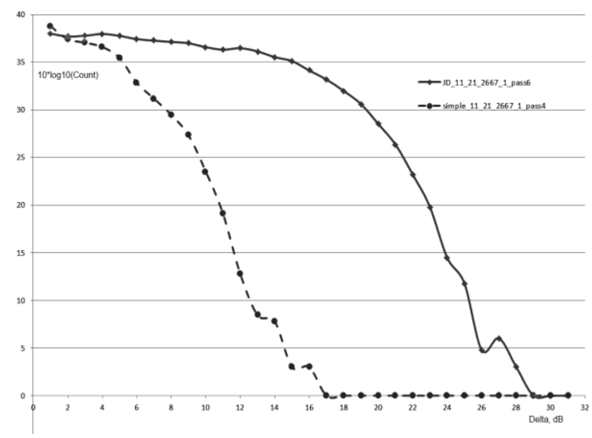

Figure 16: Level difference histograms for simple and SIC modes; both axes are scaled in decibels

[7] Kyujin Lee, Kyesan Lee, “Iterative Interference Cancellation for a PB/MCCDMA System with MMSE-FDE in the Downlink,” *Onlinepresent.org Proceedings,* Vol. 3, 2012.