Mobile Ultrasonic Signal Processing System using Android Smartphone

Won-Jae Yi, Spenser Gilliland and Jafar Saniie

Department of Electrical and Computer Engineering

Illinois Institute of Technology, Chicago IL, USA

Abstract— **This study introduces a mobile ultrasonic signal processing (MUSP) system using an Android smartphone for remote ultrasonic testing and imaging applications. The Android smartphone has multiple wireless data communication options such as Bluetooth, Wi-Fi and cellular data networks. The smartphone receives the ultrasonic data using the Bluetooth connection from a data acquisition and communication unit (DACU). With the help of Android Native Development Kit (NDK) libraries, we developed two signal processing algorithms in C programming language which are processed by the Android smartphone to explore the smartphone computing capability for ultrasonic testing applications. Split Spectrum Processing (SSP) and Chirplet Signal Decomposition (CSD) algorithms are considered for benchmarking and signal analysis. The analyzed data is displayed in real-time on the smartphone screen and streamed to a central location via Wi-Fi or cellular data networks for storage and further data analysis. A Java programmed server application is implemented to communicate with the Android application over the Internet in order to display and save the retrieved signal data. This system brings the ability to analyze ultrasonic signals remotely and to transfer ultrasound data from one end to the other for extensive signal ultrasonic imaging at a central location. The accessibility of the ultrasonic data at the central location allows experts to review ultrasonic information and make decision about the state of the health of structures and critical components under test.**

I. INTRODUCTION

Ultrasonic measurements, imaging and analysis can be achieved with the help of various signal processing methods [1], [2]. These signal processing methods can be implemented in a discrete Digital Signal Processing (DSP) hardware alongside an ultrasonic transducer. This type of DSP hardware is normally built using Application Specific Integrated Circuits (ASIC) which can deliver high signal processing performance. Also, the DSP hardware can be constructed on a Field Programmable Gate Arrays (FPGA) which can be easily reconfigured. However, employing either an ASIC or FPGA as a signal processing system may be costly, non-ideal as a mobile system or could be a time-consuming task to build a dedicated hardware for only a signal processing device.

In recent years, a major paradigm shift has occurred in the portable device industry. A mobile phone is now equipped with a somewhat powerful RISC-based microprocessor, a user-friendly multi-tasking enabled operating system with touchscreen inputs, and built-in wireless data communication adapters. The biggest advantage of having this mobile device, a smartphone, relies on its wireless connection capability enabling users to connect to the Internet and to configure their own personal wireless network for various purposes.

In this study, we present a mobile system which can be used as a powerful signal processing device with wireless communication ability for retrieving and streaming data to a central location for storage and more intensive analysis using an Android smartphone.

II. SYSTEM DESIGN

Figure 1 shows the overview of our developed mobile ultrasonic signal processing system. Ultrasonic signal data used in this study is retrieved from an ultrasonic non-destructive testing experiment setup. Properties of a material can be found by observing the reflection of the ultrasonic pulse sent to the material where the reflection is called an echo. Two dominant echoes exist in this experiment setup when the wave enters the material and when the wave leaves the material. Also, more echoes are detected between these dominant echoes, in case of a defect in the sample between these dominant echoes.

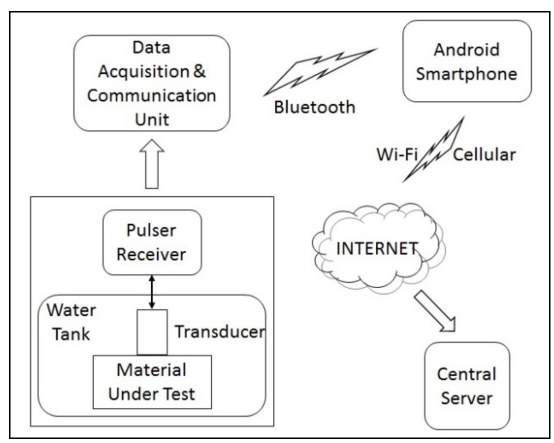


Figure 1. Mobile Ultrasonic Signal Processing (MUSP) System

These ultrasonic data, acquired by a data acquisition and communication unit (DACU), can be received on the Android smartphone using the Bluetooth protocol. A typical Bluetooth connection has the theoretical maximum throughput of 2.1-3.0 Mbit/s [3] which delivers sufficient amount of bandwidth to provide a real-time measured signal to the Android smartphone.

978-1-4799-0066-4/13/$31.00 ©2013 IEEE

The Android smartphone in this study uses a Samsung Galaxy S III (SGH-T999), a very common Android smartphone in the market, to carry out the signal processing and data transmission tests. This device embeds a Qualcomm MSM8960 dual-core 1.5GHz CPU [4] which is powerful enough to deliver satisfactory computing performance results. Also, the smartphone is equipped with a Bluetooth transceiver which enables local wireless network formation. A Wi-Fi adapter and a cellular data network radio transceiver for Internet connections are also available to freely transmit data to the central server.

In our design, the central server can be used as a data storage center to collect all sensor data histories via the Internet for remote sensor devices. A Java server application is established to communicate with the Android smartphone through the Internet connection. This application displays and saves the received data in a text file.

The term "real-time" in this paper is used to express that the system meets a real-time requirement of 5 measurements per second in echo estimation and signal decomposition applications [5], and 100 or more measurements for split spectrum processing [6].

III. Signal Analysis Algorithms and Implementations

A. *Split Spectrum Processing (SSP)*

Split Spectrum Processing (SSP) is a technique used for reducing the scattering echoes that are found in material microstructures [7]. Scattering echoes found in ultrasonic non-destructive tests where the microstructures of the material under test are smaller than the excitation wavelet's wavelength. The scattered interference pattern is governed by the wavelet frequency bands. As shown in Step 2 to 4 of Figure 2, the SSP decomposes a broadband source to generate multiple sub-band signals. Then, these sub-band signals are reunified through a post processing algorithm to reduce the energy of the scattering echoes as shown in Step 5 of Figure 2.

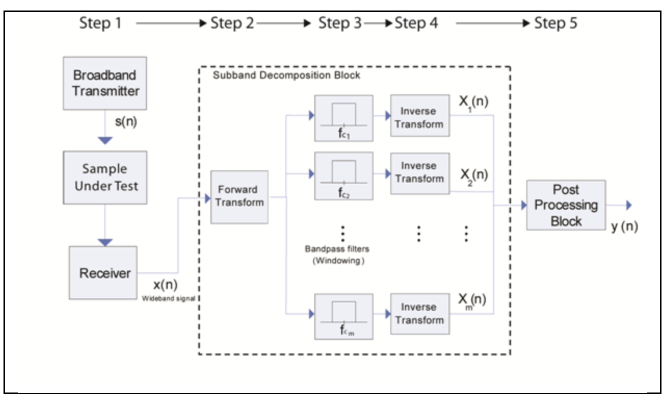


Figure 2. Split Spectrum Processing algorithm

This algorithm is coded in C programming language and integrated to the Android application using the Android Native Development Kit (NDK) toolset. This toolset allows users to interface with their own native-code languages (e.g., C and C++) to provide a more in depth code libraries for the Android application [8].

B. *Chirplet Signal Decomposition (CSD)*

Chirplet Signal Decomposition (CSD) is a technique used for raw data compression by estimating chirplets using six parameters [5], [9]. Since ultrasonic echoes are similar to chirplets, these echoes can be also illustrated using the six parameters which define a chirplet. By using this technique, a significant reduction in signal data representation can be achieved. The estimated chirplets can be used for material characterization, data compression, pattern recognition, and signal classification. Correlation operations are the weakness in the CSD algorithm; CSD requires a chirplet to be estimated multiple times. This can be improved by maintaining a table of pre-computed values which leads to a significant reduction of computation time. Figure 3 describes the CSD algorithm in a flowchart.

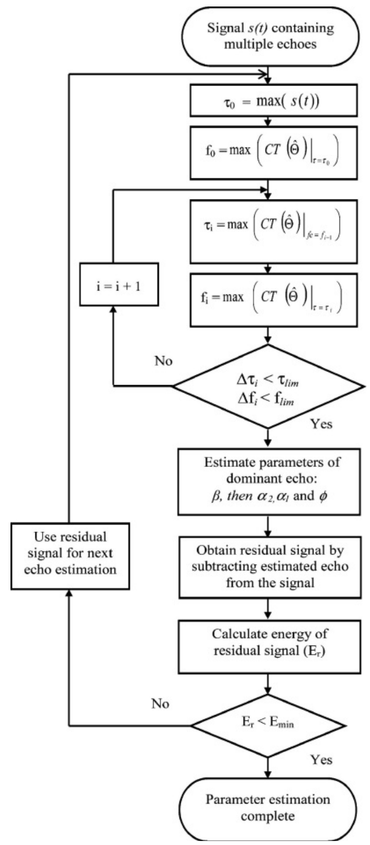


Figure 3. Chirplet Signal Decomposition algorithm

As shown in Figure 3, the CSD algorithm requires the following steps. First, the initial chirplet time of arrival is estimated to be the location of the maximum amplitude in the time domain. Then, the frequency is estimated based on a short-time Fourier transform performed on a discrete number of samples in the time domain. The time of arrival and frequency are re-estimated after the initial estimation. The algorithm then estimates the remaining parameters through co rrelation. This is the main computationally intensive part of the algorithm. Finally, the estimated chirplet is removed from the time domain signal so that another chirplet can be found using the same algorithm. This CSD algorithm is also implemented in C programming language and linked to the Android application using the Android NDK toolset to display and stream the signal data.

IV. Experimental Results and Performance Evaluation

In this study, SSP and CSD algorithms are implemented and tested under the Android 4.1.1 operating system where the ultrasonic signal data are received through the Bluetooth connection, analyzed using the two algorithms and transmitted to a central location for storage and assessment through the Internet connection via Wi-Fi or cellular data networks.

A. SSP Signal Analysis

The results of the SSP algorithm are displayed on the Android smartphone where 2048 samples of data are analyzed using 4 sub-band channels as shown in Figure 4. Channel 1 covers 0 to 4MHz frequency band, channel 2 covers 0.5 to 5.9 MHz frequency band, channel 3 covers 1 to 5.8MHz frequency band and channel 4 covers 1.5 to 5.9MHz frequency band.

In this example, the Android smartphone was able to analyze the data in approximately 6.81ms and the flaw was located at about 1700th sample across the channel sub-band results and output of the SSP algorithm. After analyzing the input signal like in Figure 4, the Android system then transmits all 4 channel sub-bands and output data to the central server through the Internet connection. Figure 5 describes the server displaying the received individual channel and output data on the screen.

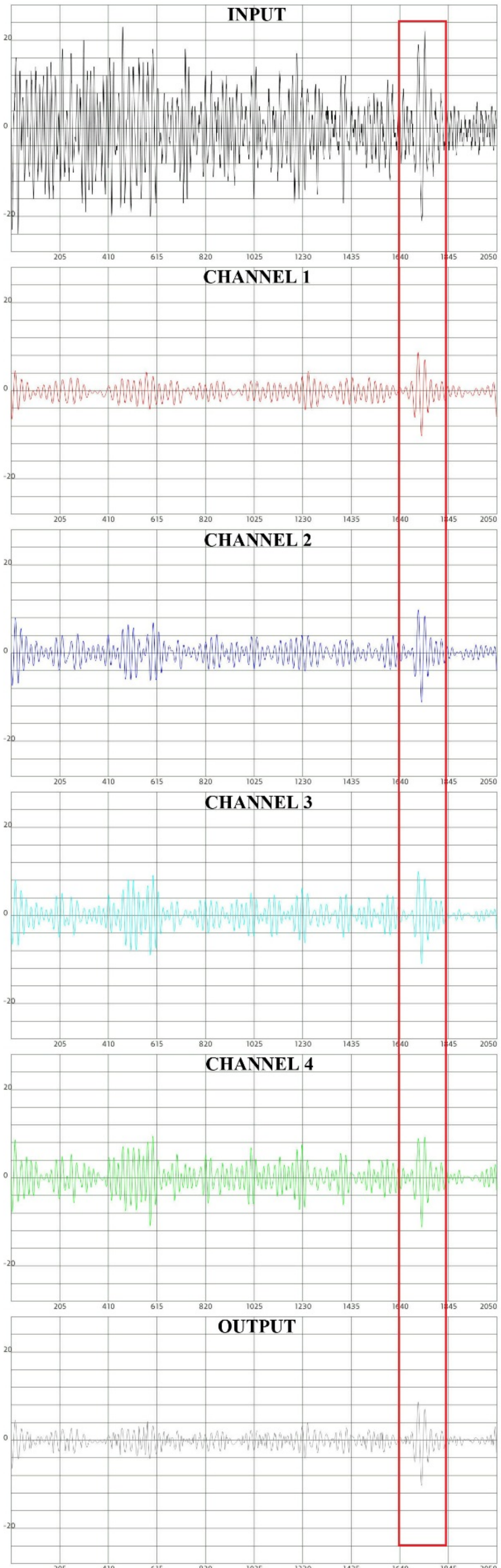
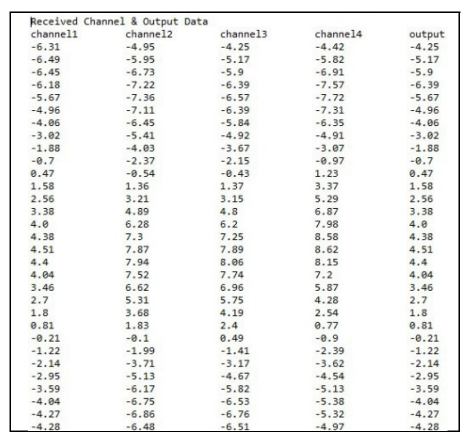


Figure 4. Real-time SSP sub-bands and results displayed on the Android smartphone (Red box indicates the flaw)



Received Channel & Output Data

channel1	channel2	channel3	channel4	output
-6.31	-4.95	-4.25	-4.42	-4.25
-6.49	-5.95	-5.17	-5.82	-5.17
-6.45	-6.73	-5.9	-6.91	-5.9
-6.18	-7.22	-6.39	-7.57	-6.39
-5.67	-7.36	-6.57	-7.72	-5.67
-4.96	-7.11	-6.39	-7.31	-4.96
-4.06	-6.45	-5.84	-6.35	-4.06
-3.02	-5.41	-4.92	-4.91	-3.02
-1.88	-4.03	-3.67	-3.07	-1.88
-0.7	-2.37	-2.15	-0.97	-0.7
0.47	-0.54	-0.43	1.23	0.47
1.58	1.36	1.37	3.37	1.58
2.56	3.21	3.15	5.29	2.56
3.38	4.89	4.8	6.87	3.38
4.0	6.28	6.2	7.98	4.0
4.38	7.3	7.25	8.58	4.38
4.51	7.87	7.89	8.62	4.51
4.4	7.94	8.06	8.15	4.4
4.04	7.52	7.74	7.2	4.04
3.46	6.62	6.96	5.87	3.46
2.7	5.31	5.75	4.28	2.7
1.8	3.68	4.19	2.54	1.8
0.81	1.83	2.4	0.77	0.81
-0.21	-0.1	0.49	-0.9	-0.21
-1.22	-1.99	-1.41	-2.39	-1.22
-2.14	-3.71	-3.17	-3.62	-2.14
-2.95	-5.13	-4.67	-4.54	-2.95
-3.59	-6.17	-5.82	-5.13	-3.59
-4.04	-6.75	-6.53	-5.38	-4.04
-4.27	-6.86	-6.76	-5.32	-4.27
-4.28	-6.48	-6.51	-4.97	-4.28

Figure 5. Received real-time SSP results on the server

B. CSD Signal Analysis

Similar to the SSP algorithm experiment, the CSD algorithm results were displayed in real-time on the Android smartphone. The CSD algorithm in this experiment takes 512 samples per measurement to estimate the chirplet to be expressed in six parameters. As shown in Figure 6, the Android smartphone displayed two outputs, the original ultrasonic input signal and the reconstructed ultrasonic signal based on the estimated echoes through the CSD process.

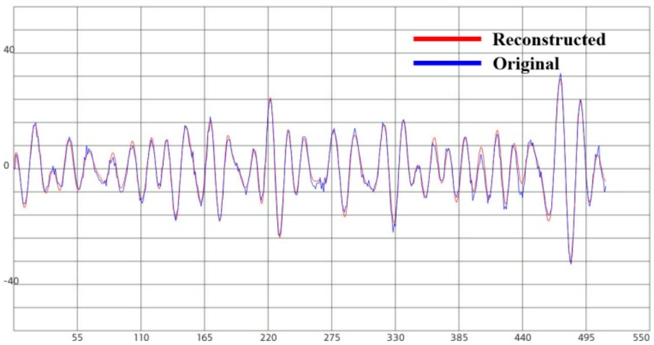


Figure 6. Real-time original and reconstructed ultrasonic signal output displayed on the Android smartphone

In Table I, the performance results are described and it shows that it is possible to achieve real-time ultrasonic signal processing on the Android system for the CSD algorithm with 512 samples of datasets. Furthermore, the result shows that of the Signal-to-Noise (SNR) ratio exceeds 10dB for the combination of 17 estimated echoes.

TABLE I. CSD EXECUTION TIME AND SNR FOR 512 DATASET SAMPLES

Echoes	*Execution Time*	*SNR*
1	34 ms	1.365 dB
2	38 ms	2.073 dB
4	49 ms	3.010 dB
8	123 ms	5.679 dB
16	158 ms	9.841 dB
17	169 ms	10.245 dB

The data which needs to be transmitted to the server is not the reconstructed signal output, but only the six parameters used for the reconstruction. Figure 7 illustrates the received echo parameters on the server in real-time from the Android smartphone. Depending on the number of echoes requested in the CSD algorithm configurations, the received data will be in multiples of six parameters.

Received Echo Parameters

% t	f	g2	g1	phi	amp
4.7000	4.5000	24.0000	22.0000	-0.3402	0.9000
2.2000	6.0000	10.0000	22.0000	-0.3402	0.6000
1.3300	7.0000	-25.0000	24.0000	0.6603	0.4000
3.2000	6.0000	24.0000	32.0000	0.2601	0.5000
2.9300	6.0000	23.0000	16.0000	-0.5403	0.4000
0.3300	7.0000	24.0000	11.0000	-0.3402	0.2500
4.1700	7.0000	0.0000	10.0000	-0.1401	0.3500
1.6800	6.0000	24.0000	22.0000	-0.5403	0.5000
3.1700	4.5000	-8.0000	7.0000	0.4602	0.2000
2.1700	3.5000	-13.0000	7.0000	0.6603	0.2500
0.4500	4.0000	8.0000	7.0000	0.0600	0.2000
4.5700	3.5000	-12.0000	49.0000	0.2601	0.3000
0.8300	5.0000	19.0000	8.0000	0.0600	0.2000
4.8900	6.0000	3.0000	32.0000	0.2601	0.3000
3.3800	8.5000	-9.0000	7.0000	0.4602	0.2000
2.2200	7.5000	-10.0000	7.0000	0.0600	0.1500
2.8900	7.5000	-12.0000	36.0000	0.2601	0.2000
0.0200	7.5000	-4.0000	14.0000	0.0600	0.2000

Figure 7. Received real-time CSD echo parameters on the server

V. CONCLUSION

In this study, we have explored a different perspective of the usage of a common Android smartphone as a mobile ultrasonic signal processing system. This system has the wireless data communication ability through the Bluetooth connection with the DACU, a unit that retrieves ultrasonic signal data from the sensor, analyzes the ultrasonic signal using two different computationally intense algorithms and streams the output result to the central server through the Internet connection via Wi-Fi or cellular data network connections. To test the signal processing capability of the Android smartphone, SSP and CSD algorithms implemented in C programming language are linked to the Android application level using the Android NDK libraries. The Android smartphone delivered satisfactory performance for both SSP and CSD algorithms where the algorithm execution times were approximately 7ms and 170ms to accomplish successful material flaw detection, and raw data signal decomposition, compressions, and reconstruction, respectively. All output data from both signal processing algorithms were then transmitted to the central server which was employed by a Java server application to display and save the incoming data. Major advantage of having an Android smartphone as a signal processing unit is that the signal data can be analyzed using the powerful CPU embedded inside the smartphone, and also allows a mobile processing system to be tightly coupled with multiple data transmission capabilities and providing a user-friendly interface.

REFERENCES

[1] J. Saniie; E. Oruklu; S. Yoon; , "System-on-chip design for ultrasonic target detection using split-spectrum processing and neural networks," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol.59, no. 7, pp.1354-1368, Jul. 2012

[2] E. Oruklu; J. Weber; J. Saniie; , "System-on-chip subband decomposition architectures for ultrasonic detection applications," *Journal of Signal Processing*, vol. 68, no. 3, pp. 367-377, Sep. 2011

[3] E. Ferro; F. Potorti; , "Bluetooth and Wi-Fi wireless protocols: a survey and a comparison," *IEEE Wireless Communications*, vol.12, no.1, pp.12-26, Feb. 2005

[4] GSMArena. (2012) Samsung Galaxy S III T999. [Online]. Available: http://www.gsmarena.com/samsung_galaxy_s_iii_t999-4804.php

[5] Y. Lu; R. Demirli; G. Cardoso; J. Saniie; , "A successive parameter estimation algorithm for chirplet signal decomposition," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 53, pp. 2121–2131, Nov. 2006

[6] J. Saniie; D.T. Nagle; K.D. Donohue; , "Analysis of order statistic filters applied to ultrasonic flaw detection using split-spectrum processing," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 38, no. 2, pp. 133-140, Mar. 1991

[7] J. Weber; E. Oruklu; J. Saniie; , "FPGA-based configurable frequency-diverse ultrasonic target-detection system," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 871–879, Mar. 2011

[8] Android Developers. (2013) Android NDK. [Online]. Available: http://developer.android.com/tools/sdk/ndk/index.html

[9] A. Kasaeifard; J. Saniie; E. Oruklu; , "Chirplet parameter estimator based on ellipse fitting in time-frequency distributions for ultrasonic NDE applications," *2010 IEEE International Ultrasonics Symposium (IUS),* pp.2024-2027, Oct. 2010