A Reliable and Cost-Effective Sand Monitoring System on the Field Programmable Gate Array (FPGA)

Xi-Yue Xiang, Itthichok Jangjaimon, Mohammad Madani, and Nian-Feng Tzeng

***Abstract*—Sand monitoring gives the benefits of avoiding equipment erosion and production failure in the oil industry. This paper presents the design and implementation of a reliable and cost-effective sand monitoring system for measuring sand production in gas and oil flows in real time. The designed monitoring system involves two acoustic emission (AE) sensors and one Doppler sensor for gauging the sand impaction and the velocity of sand particles in a nonintrusive manner, respectively. It is implemented on a field programmable gate array (FPGA) as a prototype for real-time data acquisition and processing, and evaluated using a testbed pump skid available in our laboratory. Relying on a low-cost FPGA board to integrate all acquisition and processing functionality, our monitoring system can measure the sand production amount on-the-fly reliably with high accuracy, according to our experimental evaluation. It is likely to achieve better accuracy than one without the Doppler sensor. The proposed monitoring system is affordable for wide deployment, given its high accuracy, good resilience to surrounding noise, and low cost (when compared with commercially available systems whose price tags can be some tenfold higher).**

***Index Terms*—Acoustic and Doppler measurement, field programmable gate arrays (FPGAs), finite impulse response (FIR) filters, non-intrusive monitoring, prediction models, prototyped systems, real-time data acquisition and processing, regression analyses.**

I. Introduction

SAND PRODUCTION plays an essential role in the oil and gas industry, not only because equipment can get damaged by sand erosion and bring about a tremendous loss to oil companies, if left unchecked, but also because unpredictable failures may lead to a consequent safety risk to the people on-site, and eventually to unstoppable ecological catastrophes.

Many solutions have been proposed to improve accuracy, reliability, and generality in monitoring and measuring sand production. Overall, sand detection and monitoring can be either erosion-based or acoustic-based [1].

Manuscript received July 10, 2012; revised September 27, 2012; accepted October 17, 2012. Date of publication April 26, 2013; date of current version June 5, 2013. This work was supported in part by the U.S. Department of Energy (DOE) under Award DE-FG02-04ER46136, and by the Board of Regents, State of Louisiana, under Contract DOE/LEQSF(2004-07)-ULL. The Associate Editor coordinating the review process was S. Mukhopadhyay.

The authors are with the Department of Electrical and Computer Engineering, and with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, Louisiana 70504, USA (e-mail: tzeng@cacs.louisiana.edu)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2013.2253991

The erosion-based sand management is handled in an intrusive manner by inserting sensors into pipes to sense the variation of the resistance as the flow passes through. In 1992, the first commercially available topside intrusive monitoring system was developed by CorrOcean ASA Company. The electrical resistance probe was utilized to record the change of the resistance of the sensing elements due to the reduction in their thickness caused by sand erosion [2]. This method allowed the sensing element to directly contact particles and thus was insensitive to system noise.

The acoustic-based technique is usually nonintrusive, avoiding the sensing instrument from intruding into the pipe or container. It clamps acoustic-based transducers at the bends of pipelines to measure sand production in a nonintrusive manner by monitoring the sand impaction to the pipe wall. The instrument measures the sand production rate, quantifying it in g/s.

Both current commercial intrusive and nonintrusive sand monitoring systems suffer from various deficiencies. For erosion-based monitoring, system accuracy and temperature-sensitivity are vital, rendering the system impossible for the long run. On the other hand, an acoustic-based monitoring system, while not sensitive to the temperature, often cannot achieve real-time monitoring, because data is sampled at the MHz-level rate and collected by a data acquisition unit for a PC to process via a certain signal processing tool. Given massive data is gathered for processing in a very short period of time, the real-time computation for continuous monitoring is usually infeasible. Furthermore, nonintrusive acoustic-based monitoring is prone to the surrounding noise induced by such sources as rain drops and wind blow to the pipelines, opening and closing valves, the operation of pumps and compressors [2]. With a full-fledged PC (which runs signal processing codes) plus additional circuits for data acquisition and noise filtering, a commercially available system now costs $20,000 to $25,000 [3]–[5]. Despite its high price tag, such a commercial monitoring system is known to result in erroneous sand detection caused by surrounding noise in the field, trigging a false alarm.

The field programmable gate array (FPGA) is fast and highly reconfigurable, enabling rapid prototyping of new algorithms and control mechanisms for evaluation and real-world applications. It has been adopted for such wide use as robust infinite-impulse-response (IIR) filter implementation

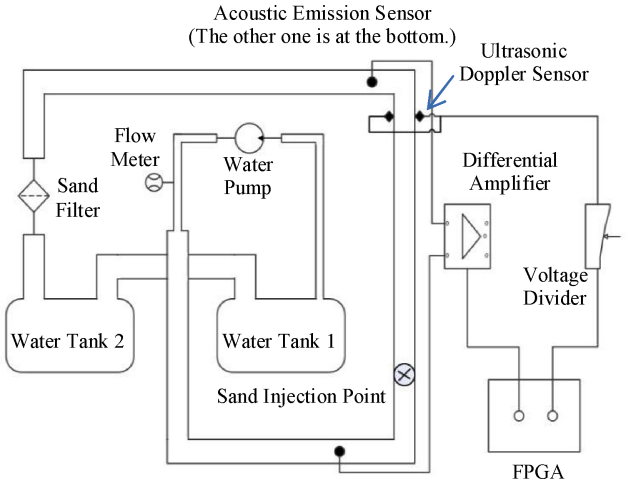


Fig. 1. Configuration of the testbed pump skid in our laboratory for evaluating the prototyped design.

[6], vibration analyses for industrial applications [7], and ionizing radiation detection for environmental awareness [8], among others. An FPGA-based design makes it possible to easily integrate other modules for an efficient system, to realize speedy and parallel data processing required by many real-time and continuous monitoring and control, and to support quick reconfiguration in response to constant updating so as to comply with new requirements or to accommodate improved algorithms. Many FPGA-based integrated systems have been proposed to conduct the monitoring and control processes for a variety of applications, such as multichannel data acquisition and monitoring [9], continuous computer numerical control [10], smart sensor wavelet and wavelet packet analyzing and monitoring [11], and fault detection and reconfiguration for power electronic converters [12].

This paper introduces an affordable sand monitoring system designed and prototyped on FPGAs, aiming to measure sand particles in the flows of gas and oil pipelines on-the-fly with high resilience to background noise and good accuracy. The FPGA board adopted for our implementation costs $243 [17], and implementing the entire sand monitoring functionality takes up only 45% of available lookup table (LUT) cells on the FPGA board. The sand prediction model implemented on the FPGA board is shown to yield accurate prediction results, based on experimental evaluation using the testbed pump skid available in our research lab (Figs. 1 and 12). The FPGA board facilitates the high sampling rate of differential acoustic emission (AE) signals at 1.2 million readings per second, and it can process those sampled AE signals plus the samples of the Doppler sensor in real time. With the FPGA board and two AE sensors plus one Doppler sensor, a prototyped system like this is estimated to cost roughly $2,230, amounting to some 1/10 of the price tag of a typical sand detection system now commercially available while avoiding false alarms triggered by surroundings. It is readily applicable for wide deployment in the field, thanks to its high accuracy, good resilience, and low cost.

Scientific contributions of this designed and implemented system are multifold. First of all, our system processes massive sampled data from AE and Doppler sensors in real-time by means of approximating the fractional multiplicands and of adopting Horner's shift & add algorithm [15], [16] to provide more efficiency and better accuracy for the floating point multiplications. This partially leads to improving the accuracy of our system to some 91% from 85% achieved by its commercial counterpart of ClampOn SandQ^{TM} Monitor [18]. Secondly, while most of the functions available for window filter design have fixed minimum stop-band attenuation for each of those functions, our design employs the Kaiser filter window instead [13], since all other functions have conflicting requirements for as narrow a transition region as possible and for as high stop-band attenuation as possible. For the Kaiser window function, its stop-band transition region is independent of its attenuation, particularly suitable for our system where the finite impulse response (FIR, detailed in Section IV-C) filter is incorporated, given that the current output in our system is relevant not only to the current input but also to the previous inputs. Thirdly, the distributive arithmetic (DA) architecture is used in our design due to its advantage that the throughput of DA FIR filter depends only on the bit precision of the input data and is independent of the filter length [14], as highlighted in Section IV-C. The implemented architecture is highly pipelined to yield a fast operation speed with high throughput. All of those features benefit to save FPGA resources dedicated for the filtering process. Finally, being FPGA-based, our design can adopt different sample data processing techniques or impulse response filters (like the stable IIR filter pursued earlier [6]) for other continuous monitoring or data acquisition applications.

In the following, a brief review of related work is given in Section II. The system design overview is provided in Section III. All major functional modules and control logics are stated in Section IV. Section V describes the evaluation system setup, followed by our experimental procedure and measurement. The experimental results and their discussion are presented in Section VI.

II. Pertinent Work

The frequency of signals produced from sand particle impaction was found [19] to lie between 300 and 800 KHz. Previously, some monitoring systems employed AE sensors that operated in the narrow range of 300–400 KHz while others used broadband (200–800 KHz) AE sensors [2]. In 2003, data was collected from thirty sand detectors, with calibration trials performed in three offshore fields of East Coast Trinidad [20], to investigate the accuracy of those nonintrusive sand detectors in predicting sand concentration in different fluid profiles. However, the simple model proposed therein was only amenable for sand particles ranging between 30–120 μm, with the velocity of sand particles assumed to equal that of the fluid. Later, a solution for calibrating clamp-on ultrasonic sand detectors was proposed [21], where the sand amount was quantified based on the initial limit of signal profiling. It adopted an iterative method to reflect the true sand production by marking up the background noise value [21]. However, the study was still performed under the assumption that the velocity of sand particles equals that of the fluid.

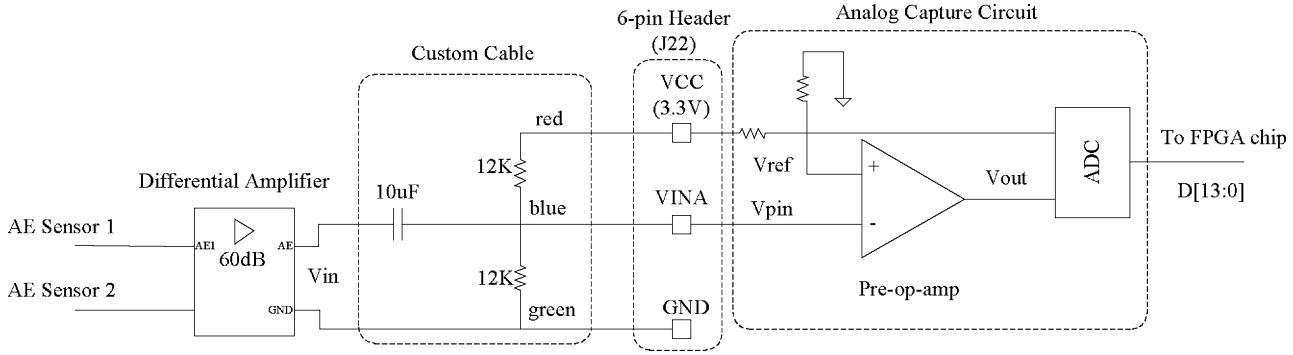


Fig. 2. I/O interface between AE sensors and the FPGA.

An accurate nonintrusive design was considered in our prior work by employing two AE sensors and an ultrasonic Doppler sensor [4]. It improved measurement accuracy by not only monitoring sand impaction to the inner side of the pipe wall, but also taking the actual velocity of sand particles into account, made possible by the use of Doppler sensor to remedy the root cause of inaccurate prediction in assuming the particles' velocity to equal that of the flow. While our prior design used two acoustic sensors in the differential mode to reduce the environmental noise, it still required an external acquisition system and a filtering circuit for data gathering, in additional to a powerful computer to process the gathered data [4].

III. System Design Overview

Fig. 1 shows the system configuration established in our lab for emulating the on-site environment for sand monitoring and measurement. This system employs two AE sensors, with one located at the vicinity of a pipeline elbow to capture the major sand impaction and another mounted far away from the sand injection and impaction points. The Doppler sensor, comprising a transmitter and a receiver, is incorporated to gauge the velocity of sonic particles (i.e., sand) flowing in the fluid. It works under the minimum particle concentration of 35 PPM, with solid particles sized at least 40 μm [22]. Both AE sensors and the Doppler sensor are coupled to the outer pipe wall with a greaser, sonic gel, or epoxy.

A. Sand Impaction Sensing

As sand impaction occurs, the kinetic energy carried by sand particles is converted to elastic waves whose frequencies are captured by the adjacent AE sensor and converted to electrical signals with different amplitudes. In earlier design [2], [19], AE sensors are selected for capturing all sand impaction whose frequency ranges from 100 to 900 KHz. The sum of the square value of the amplitude of the acoustic signals is used to compute the amount of sand flowing through the pipe during a specific time interval.

As depicted in Fig. 2, signals from two AE sensors in dual channels are connected to twin BNC port of the differential amplifier to counteract the background noise. The coupled signal is amplified by $10^6\times$ (i.e., 60 dB) and normalized within

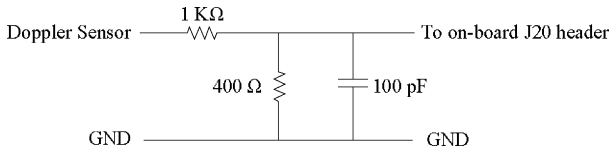


Fig. 3. I/O interface between the Doppler sensor and the FPGA.

± 1.25 V by a differential amplifier, before being scaled by one customized cable for connection to the FPGA board (model: Xilinx Spartan-3AN [17]). The circuit embedded in the customized cable provides a 1.65V DC bias to the original output of the differential amplifier, since the input of the on-board analog capture circuit requires the voltage within 1.65 ± 1.25 V [17]. The on-board analog digital converter (ADC) chip samples the output of the pre-op-amp at the rate around 1.2 MHz.

B. Sand Velocity Sensing

The sand velocity is unequal to that of the fluid in general practically, and it fluctuates slightly within a small range. The usage of one Doppler sensor provides an accurate real-time measurement of sand particle velocity.

The transmitter of the Doppler sensor emits 500 KHz signals, which pass through the pipe wall and the liquid before hitting moving particles for their velocity assessment. The reflected signals are captured by the sensor's receiver, which produces a pulse train with different frequencies reflecting varying frequency shifts induced by the movement of particles.

The frequency shift Δf is known to be 60 Hz for each 1 ft/s increment in velocity [4]. Since the frequency of pulse trains can be calculated by counting the rising or falling edges (N) over the time span of t_w, as

$$\Delta f = N/t_w, \tag{1}$$

the velocity of particles, V_f, can be determined by

$$V_f = K \cdot \Delta f = K \cdot N/t_w, \tag{2}$$

where K is 1/60 and t_w is the measurement interval, which is set to be 34.95 s per measurement instance in our evaluation (and can be any other value, depending on the real need in the field).

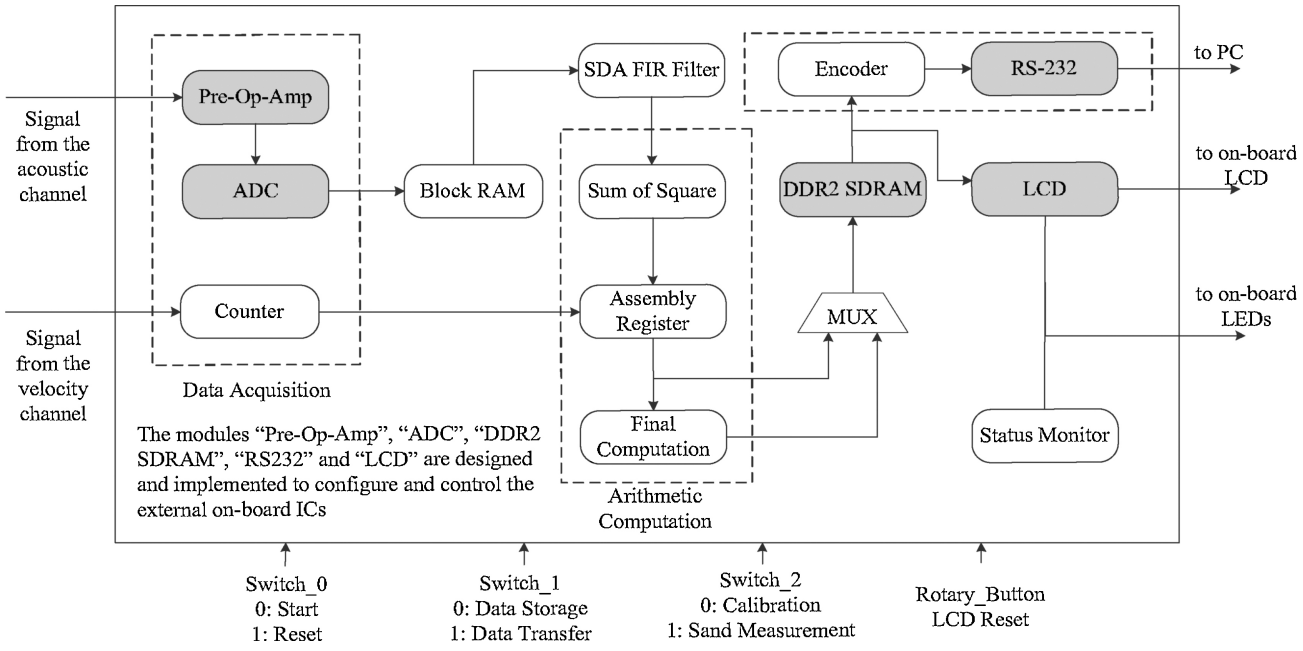


Fig. 4. Implemented logic schematic on FPGA.

Fig. 3 shows the circuit embedded in the customized cable for connecting the Doppler sensor (receiver) to the 6-pin accessory header (J20) on the Xilinx development kit. Two resistors behave as a voltage divider to attenuate the output of the Doppler sensor from 12 V down to about 3.4 V (peak-to-peak) following the I/O standard of our adopted Spartan-3AN FPGA evaluation board. A 100 pF capacitor is added to remove signal spikes induced mainly by the impedance mismatch between the signal line and the input impedance of the pre-op-amp on the Xilinx board.

IV. Implementation Details on FPGA

The whole design code is written in Verilog HDL under the development environment of Xilinx ISE 11.5. ISIM and ModelSim are employed to perform post synthesis simulation. Meanwhile, MATLAB, and Python are used to assist the design of the FIR filter. PuTTY is installed on PC to collect the data transferred through the RS-232 serial interface and generate a reader-friendly text file for the modeling process. Fig. 4 illustrates the overall design schematic flow of all the major functional modules, which are elaborated next in sequence.

A. Data Acquisition

1) *Acoustic channel:* As discussed in the previous section, the acoustic signal is first scaled in the pre-op-amp with the negative gain of -1. Then, the ADC digitizes the signal at the rate of 1.2 MHz with the resolution of 14-bit in 2's complement binary form, as

$$D[13:0] = -V_{in} \times 2^{13}/1.25 \tag{3}$$

where V_{in} (shown in Fig. 2) stands for the output voltage of the differential amplifier. The digitized samples are stored in block random access memory (BRAM) inside the FPGA chip. Fig. 5 depicts the Finite State Machine (FSM) for the acoustic data acquisition and the digitization process. The operations in each state of the FSM are described below:

1) INIT: The process is initiated after power on.

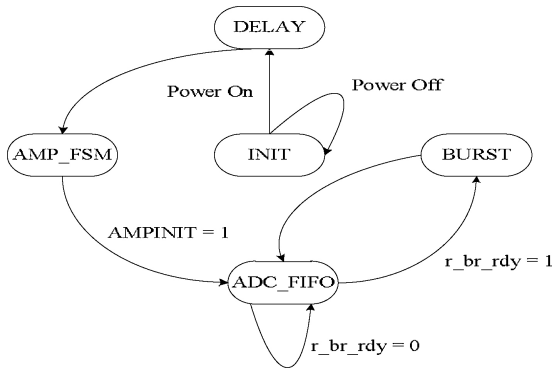


Fig. 5. FSM for signals sampling process in the acoustic channel.

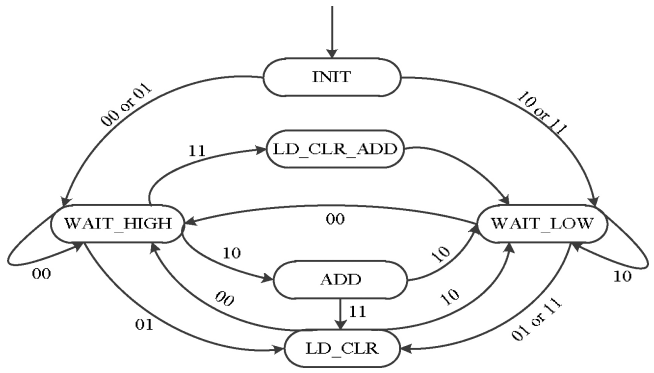


Fig. 6. FSM for counting rising edges of Doppler signals.

2) DELAY: The whole process is halted for about 180 clock cycles at 5 MHz to globally initialize all the involved components.
3) AMP_FSM: Configuration of the pre-amplifier is initiated, and is done only once by setting its gain to be −1 through sending '00010001' on SPI_MOSI bus in the bit serial format [17]. The triggering signal of AMPINIT is kept asserted after completing the configuration process.
4) ADC_FIFO: Toggle the digitization process in the ADC and clock the result through ADC_OUT into the ADC FIFO. Assert ADC_FIFO_RDY when the FIFO is full, indicating that one full sampling process is accomplished.
5) BURST: The FIFO data is burst into a register when the FIFO is full. The register then forwards its contents to BRAM before any new data is written into it.

2) *Velocity channel:* We use a 42 MHz master clock CLK_42 MHZ to capture the rising edges of the slave signal VEL_SENS. An active high signal of VEL_LD_CLR triggers loading the current count value into a register and then resetting the counter. Since VEL_LD_CLR may arrive at any time during the VEL_SENS pulse train, different conditions have to be taken into account in order to prevent any rising edge from being counted twice during the same slave signal interval. Also, the load and clear operations need to be executed in turn with one clock cycle apart.

Fig. 6 illustrates how the FSM is involved in counting the number of rising edges of Doppler signals. The numbers on transition arrows represent the statuses of the slave signal VEL_SENS (the Doppler signal used to calculate sand particle velocity) and of the toggling signal VEL_LD_CLR. For example, 01 means VEL_SENS = 0 and VEL_LD_CLR = 1. The toggling signal VEL_LD_CLR is asserted after completing the overall acoustic signal sampling process during each time window (of 34.95 seconds). The corresponding operation of each state is outlined below:

1) INIT: The initial state decides the commencing status of the counter logic.
2) WAIT_HIGH: When the slave signal of VEL_SENS is low, the counter is halted until VEL_SENS is asserted.
3) WAIT_LOW: When the slave signal of VEL_SENS is high, the counter is halted and waits for the falling edge of VEL_SENS. A full cycle of square waveform consists of a high level followed by a low level of VEL_SENS.
4) ADD: The counter is added with 1.
5) LD_CLR_ADD: Add 1 to the counter and assign the value of the counter to the output register before clearing the counter to 0.
6) LD_CLR: Assign the value of the counter to the output register and then clear the counter.

B. Block Random Access Memory

One dedicated dual-port memory is configured into simple dual-port BRAM as the data buffer to coordinate the operations of ADC and the downstream FIR filter module. The data buffer has the capacity of 2048 samples, each with 14 bits. Write to (or reads from) the buffer is via port A (or port B).

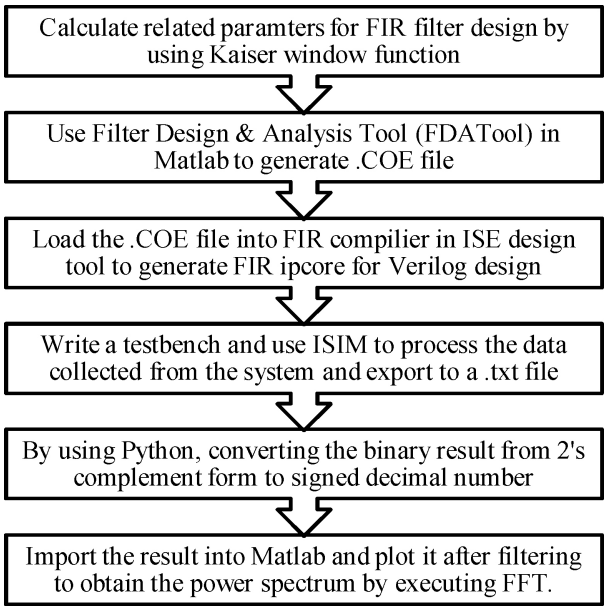


Fig. 7. Methodology for FIR filter design.

Write operations are clocked by CLKA with 42 MHz, while read operations uses CLKB of 50 MHz to clock out the data. Since the operating modes for simple dual-port BRAM are not selectable, they are fixed as READ_FIRST to deal with data collisions. During a collision, the write mode of port A can be configured in such a way that the read operation on port B either produces a correct or undefined data (X_s) [23].

BRAM can be viewed as two halves, each with 1024 bins. It starts to clock out the data after the upper half (of 1024 bins) is filled up. While reading out the data, the lower half is used for holding the incoming sampled data. Thereafter, BRAM behaves in a typical dual-buffer manner, with one half in the read mode (to read out its filled data) and the other half in the write mode (to keep incoming sampled data). BRAM is large enough so that data stored in its one half is read out fully before the other half is filled up. The functional module provides two control signals: A_UPPER and A_LOWER to acknowledge the downward FIR filter module that the upper ([1023:0]) or the lower ([2047:1024]) half of BRAM is full and data therein is ready to be clocked out. Since acknowledging signals are transferred to the 50 MHz domain through the 42 MHz domain, they are maintained for two clock cycles to avoid data metastability caused by undesirable clock domain crossing (CDC).

C. Finite Impulse Response (FIR) Filter

The ultrasonic Doppler sensor regularly emits 500 KHz signals, which are captured by the nearby AE sensors. Thus, a band-stop filter is designed to abate this noise so produced in the acoustic frequency range. Fig. 7 illustrates the complete design and validation procedures of the FIR band-stop filter developed.

The number of taps (M) and the shape parameter (β) of the Kaiser window FIR filter are calculated as follows [13]:

$$M = \frac{aa - 7.95}{2.285 \cdot \Delta\omega} = 79, \tag{4}$$

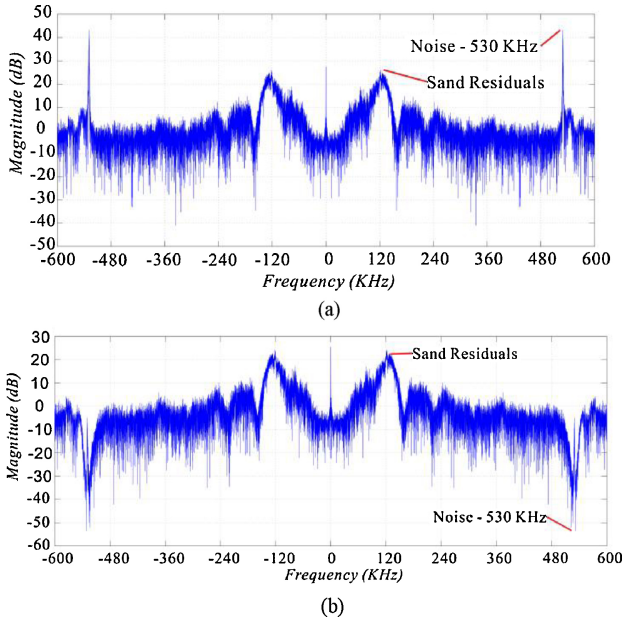


Fig. 8. Power spectrum comparison before and after filtering. (a) Before filtering. (b) After filtering.

$$\beta = 0.5842\,(aa - 21)^{0.4} + 0.07886\,(aa - 21) = 3847, \qquad (5)$$

where aa stands for the attenuation in dB and $\Delta\omega$ is the normalized bandwidth.

Based on the values of M and β, the COE coefficient file is generated by the filter design and analysis (FDA) tool in MATLAB. Xilinx LogiCORE^{TM} IP FIR Complier v5.0 is employed to generate the FIR filter module.

Unlike the conventional FIR filter, our design follows the distributive arithmetic (DA) architecture, so that its sample rate is related only to the bit precision of the input data alone. The DA architecture realizes a sequence of such basic operations as table look-ups, additions, subtractions, and shifts of the input data, with a look-up table to hold all possible partial products over the filter coefficient space [14]. The DA FIR filter has 79 taps with symmetric structure. The filter has a latency of 24 clock cycles, after receiving an input sample (available one in 15 clock cycles).

Fig. 8 demonstrates the power spectrum results of signals with 2^{14} points before and after filtering when no sand is injected into the system. The peak at 120 KHz shows the impaction produced by the sand residuals left in the system. Evidently, a sharp peak in Fig. 8(a) signifies one noise frequency centered around 530 KHz at the sampling rate of 1.2 MHz. Note that while the nearby Doppler sensor contributes to primary noise of 500 KHz, the observed noise frequency of 530 KHz represents a frequency shift induced by the DC blocking capacitor on our customized cable (see Fig. 2). The specification of the filter design is listed in Table I. Removing the peak frequency of 530 KHz is resulted from our FIR filter, as confirmed by Fig. 8(b), where the target noise of 530 KHz is effectively eliminated.

TABLE I
SPECIFICATIONS OF THE FIR FILTER

Filter Type	FIR band-stop filter
Window	Kaiser
Sampling Rate (f_s)	1.2 MHz
Cut-off frequency	510 — 550 KHz
Attenuation	45 dB

D. Arithmetic Computation

As shown previously in Fig. 4, a computational module is designed to calculate the amount of the sand flowing through the pipe by using the sum of the square (SS) value of AE signals and the number of rising edges (N) of velocity signals. For effective computation on FPGA cells, we constitute the module in three steps, as described next.

1) *Step 1- Independent Variable Computation:* The functional module is designed to compute SS and N in the FPGA during the selected time interval (of 34.95 seconds) and transfer the results to a PC through the on-board RS-232 serial interface for Step 2 stated next. Therefore, the downstream Double Data Rate Synchronous Dynamic Random-Access Memory (DDR2 SDRAM) plus the RS-232 Encoding and Transfer module is designed and tested prior to implementing the computational module. Let the output of the FIR filter be y_i, then SS is expressed by

$$SS = \sum_{i=0}^{global\_count} y_i^2 \qquad (6)$$

where global_count is the amount of acoustic samples processed in a unit time interval, chosen as 41,943,040 in our current design. For sampling this amount of input, it takes about 34.95 seconds under the sampling rate of 1.2 MHz. This is also the reason of designating the measurement time interval to be 34.95 seconds.

2) *Step 2- Model Construction:* This is the one-time training process for a given field installation, performed at the start of the monitoring operation with an aid of a PC to construct the sand production model. It involves running the experiment a few times under different settings to gather experimental data. With Microsoft Excel Data Analysis Toolpak, the PC performs multiple regression analysis on gathered data, with respect to the actual injected sand amounts under which the data was collected, to get independent variables of SS and N for the mathematical model. The model obtained from the multiple regression analysis on data collected using the testbed pump skid in our laboratory is

$$G_1 = 1.845 \times SS/2^{64} + 1.393 \times N/2^{10} - 114.248. \qquad (7)$$

3) *Step 3- Dependent variable computation:* The floating point multiplications in (7) can be carried out by using Horner Shift and Add Algorithm, which is an efficient method for multiplying binary numbers without any hardware multiplier [15], [16]. It is realized by trading the cost and speed for accuracy, as follows. For obtaining the first 6 factional bits of the coefficients in the product of $1.845 \times SS$ and $1.393 \times N$, one can approximate the original two numbers

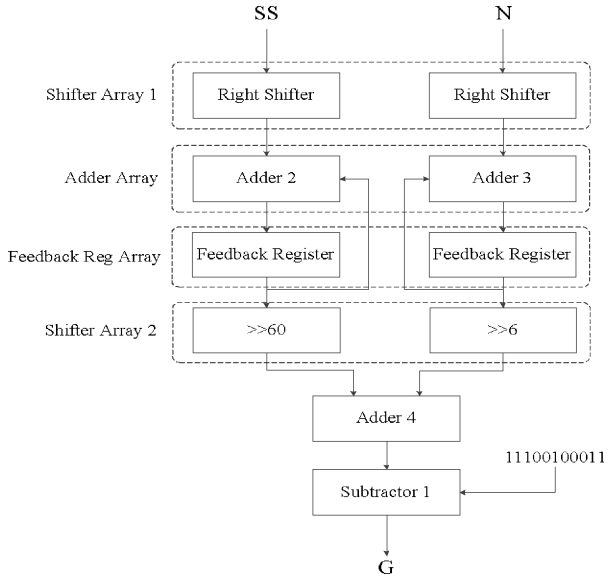


Fig. 9. Architecture of the final computation module.

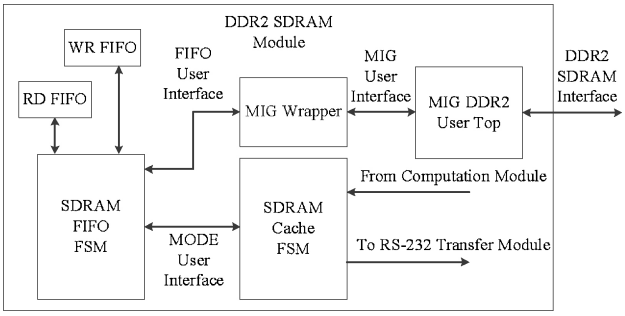


Fig. 10. Architecture of the DDR2 SDRAM data interaction interface.

as $1.84375\,(=1.110110_2)$ and $1.39065\,(=1.011001_2)$ before following repeated shifts and add operations using the FPGA cells. In order to compensate for the lost precision resulted from the Horner's algorithm, both sides of (7) are multiplied by 2^4, so that the first (or the second) division operation in the right-hand side expression of (7) is realized by right-shifting 60 bits (or 6 bits), instead of 64 bits (or 10 bits) and the constant 114.248 becomes $114.248 \times 2^4 \approx 1827$. This way increases the operation precision by four bits, yielding its final implementation on the FPGA as

$$G_2 \approx (SS + SS \gg 1 + SS \gg 2 + SS \gg 4 + SS \gg 5) \gg 60 + (N + N \gg 2 + N \gg 3 + N \gg 6) \gg 6 - 1827 \quad (8)$$

where G_2 represents the amount of sand computed using FPGA cells with four bits precision after the binary point.

Fig. 9 illustrates the architecture of the final computation module for calculating the dependent variable shown in (8). Shifter Array 1 contains a number of right-shifts to form the partial products; the floating point addition computations are done by sequentially summing up the partial products in Adder Array and Feedback Reg Array.

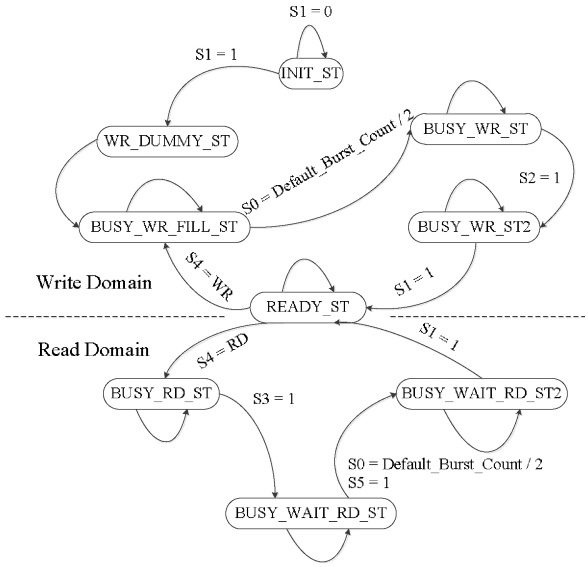


Fig. 11. FSM for the SDRAM FIFOs data interaction.

E. DDR2 SDRAM Storage

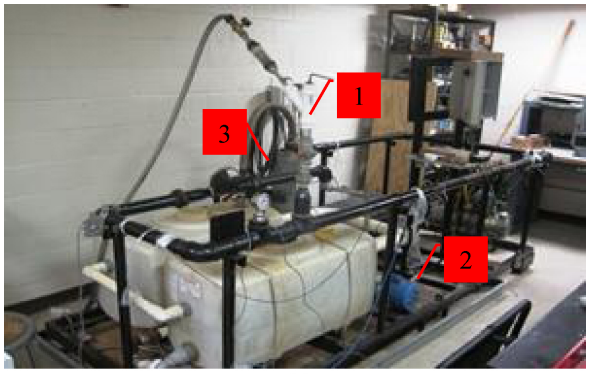
DDR2 SDRAM is used to store the intermediate and the final computational results during the course of operation. The Spartan-3AN FPGA Evaluation Board incorporates a 512 Mbit (32 M × 16) Micron Technology DDR2 SDRAM (part no. MT47H32M16) with a 16-bit data interface [17].

Fig. 10 illustrates the architecture which contains the major modules involved in the DDR2 SDRAM data interaction. MIG DDR2 User Top is generated by Xilinx Memory Interface Generator (MIG) to instantiate the main DDR2 SDRAM infrastructures. MIG Wrapper provides the simple interface for the back-to-back memory operation, which is required for higher performance. This module is rectified based on the original prototype provided by Xilinx [24] to match our requirements. The SDRAM FIFO FSM contains a state machine which controls the read or the write operation of FIFOs. The SDRAM Cache FSM behaves as the data cache for the SDRAM FIFO FSM, during the write operation of SDRAM. It accepts the data from the computation module and assigns it to the SDRAM FIFO FSM. During the read operation of SDRAM, it accepts the data from the SDRAM FIFO FSM and assigns them to the downward RS-232 module.

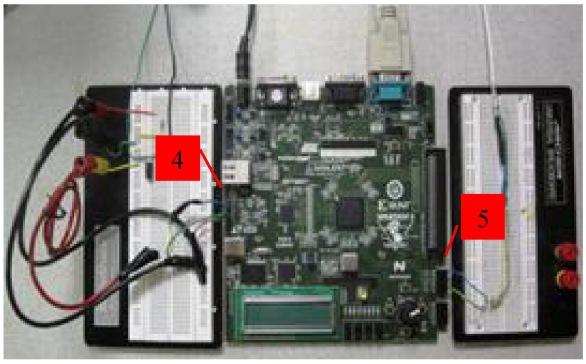
Two 18 KB BRAMs are configured as a 17×32 Write FIFO and a 17×32 Read FIFO which are generated by Xilinx LogiCore IP FIFO Generator. Back-to-back burst is applied with configurable burst length. First-word fall through (FWFT) feature will allow the FIFO to fan out the output right after the data is available in the FIFO. FWFT is available when FIFOs are created with BRAM.

Fig. 11 illustrates the SDRAM FIFO finite state machine (FSM) , which coordinates the operation between SDRAM Cache and FIFOs. The signals on the transition arrows are labeled by Si (with $0 \le i \le 5$) and explained in turns, as follows:

1) S0: The count of bursts for writing data into the WR FIFO and the RD FIFO (see Fig. 11).



(a)



(b)

Fig. 12. Prototyped system for onsite measurement. (a) Testbed pump skid. (b) FPGA board with the peripheral circuits and interfaces 1 sand injection point 2-horizontal pump 3-sand filter 4-acoustic channel input 5-velocity channel input.

2) S1: Active high as the data has been burst into DDR2 SDRAM and the WR FIFO is ready for new input. The WR FIFO cannot accept new input data if S1 is low.
3) S2: Active high as the write operation has been done.
4) S3: Active high as the read operation has been done.
5) S4: Command bus to DDR2 SDRAM.
6) S5: Active high when checking if the Read FIFO is empty. Used in the debugging process and maintained high during the normal monitoring process.

The corresponding operations in each state of the SDRAM FIFO FSM depicted in Fig. 11 are described below.

1) INIT_ST: Initialize all registers and control signals.
2) READY_ST: Decide the operation phase of the WR/RD FIFO based on the incoming command signals issued for SDRAM.
3) WR_DUMMY_ST: Add one clock cycle delay as the guarding slot.
4) BUSY_WR_FILL_ST: Store the incoming data into a register, from which the WR FIFO will read.
5) BUSY_WR_ST: Write the data into the WR FIFO.
6) BUSY_WR_ST2: Listen to the flag signal S1 to check if the writing operation in SDRAM is completed.
7) BUSY_RD_ST: Listen to the flag signal S3 to check if the data is ready at the SDRAM read port.
8) BUSY_WAIT_RD_ST: Flush the data at the output port of SDRAM into the RD FIFO.
9) BUSY_FILL_RD_ST2: Listen to the flag signal S1 to check if the reading operation in the RD FIFO is completed.

V. Evaluation System and Measurement

A. Evaluation System Setup

According to the configuration schematic design (see Fig. 1), we have the experimental setup available in our lab as demonstrated in Fig. 12. The sand is loaded into the cylinder (Part 1) and pressurized with water in order to make sand suspended in the water. Sand injection is controlled by a ball valve positioned under the cylinder. The horizontal pump (Part 2) produces a flow rate from 20 gallons per minute (GPM) to 35 GPM by regulating the inlet valve. The injected sand will be collected by the sand filter (Part 3) before water flows back into another tank. The acoustic signals (which carry sand impact information) and the Doppler signals (which indicate the flow rate of the sand particles) are conditioned before fed to the FPGA board through Part 4 and Part 5, respectively.

B. Experimental Procedure

The experiments have to follow certain procedures to emulate the real-world oil and gas production conditions for obtaining meaningful readings. A sequence of operations needs to be conducted in order, as stated below:

Step 1: Close the ball valve after loading sand to the cylinder before measurement.
Step 2: Pressurize the cylinder with water, with the ball valve kept closed at all times.
Step 3: Run the pump.
Step 4: Wait 20 seconds for the system to stabilize. It can be observed from the readings of the flow meter.
Step 5: Start running the sand amount calculation processes realized on the FPGA device.
Step 6: Open the ball valve to inject water for flushing the residual sand. It takes about 5∼22 seconds for flushing water from the injection point to reach the pipe elbow, dependent upon the flow rate and the sand residual amount.
Step 7: Continue to run the pump for 30 seconds before shutting it down.

C. Result Measurement

The measurement results can be displayed on the on-board LCD and sent to PC through the RS-232 interface simultaneously. The data (of 283 bits) collected and transmitted to the PC also include the related parameters, with their formats illustrated in Fig. 13, where *GC* stands for the variable "global_count." G_I and G_F represent the integer part and the fractional part of the sand amount G, respectively (also applies to the velocity of the sand particles V). The binary point "." and the delimiter / are sent in the ASCII code as 0x2E and 0x2F, respectively. In the calibration (or measurement) mode, gathered results are for model establishment (or are recorded for processing based on the established model).

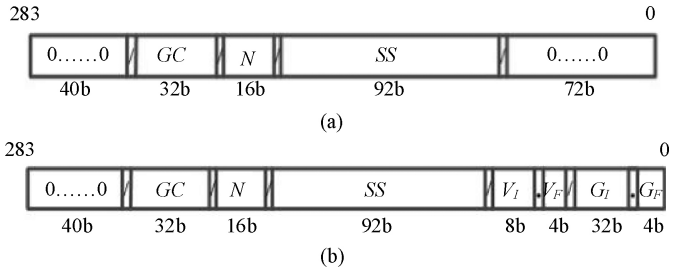


Fig. 13. Measurement data format, (a) calibration mode, (b) measurement mode.

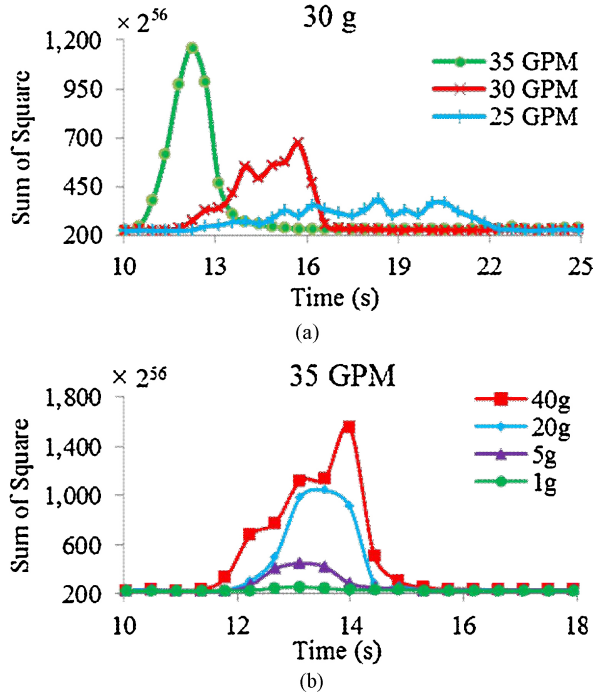


Fig. 14. Results of the sum of the square value with (a) 30g of sand injection under different flow rates and (b) different amounts of sand injection under the flow rate of 35 GPM.

VI. Experimental Results and Discussion

This section first provides insights about the relationship between the sum of the square value (*SS*) and the sand impaction, before comparing different prediction models established out of sensor measures obtained using the testbed pump skid (shown in Fig. 12) under sequences of experimental runs with varying sand injection amounts and flow rates. Specifically, 1 g, 5 g, 10 g, 15 g, 20 g, 25 g, 30 g, 35 g and up to 40 g of sand were injected under 25 GPM, 30 GPM and 35 GPM, respectively. Their results were gathered and consolidated into nine comparison groups. The values of *SS* were collected during runs of 0.437 second each.

The *SS* results as a function of time are demonstrated in Fig. 14, where similar trends are observed for all 9 comparison groups. Fig. 14(a) depicts the *SS* results over time as 30 g of sand were injected under 25 GPM, 30 GPM and 35 GPM. It is evident that with each specific amount of sand injection, the apex of the curve is larger for a higher flow rate, since stronger sand impaction is then experienced. Fig. 14(b) demonstrates that the total area under the sum of the square value curve is larger when more sand is injected under the specific flow rate (of 35 GPM).

A. Model Establishment and Discussion

1) *Accuracy:* Fig. 15 illustrates the comparative results of different models over 17 experimental runs observed, with G_0 being the actual injected sand amount. G_1, given by (7), is obtained from the multiple regression analysis, whereas G_2, expressed in (8), is an approximation for the efficient implemention on FPGA, as stated in Section IV-D earlier. In order to assess accuracy enhancement attributed by the Doppler sensor, we also established a model considering only the measures of AE sensors and ignoring those of the Doppler sensor, denoted by G_3, as expressed by (9) below:

$$G_3 = 9.78 \times 10^{-20} SS - 103.41. \tag{9}$$

Note that G_1 and G_2 take into account the measures of both AE and Doppler sensors. It is apparent from Fig.15 that the model implemented on the FPGA by using the measures of both AE and Doppler sensors, G_2, makes better prediction on the sand production amount. In fact, the approximated results by G_2 are consistently very close to those of its G_1 counterpart, which computed the sand amount using complex expression given in (7). Additionally, G_2 provides more accurate amounts (i.e., closer to the actual values indicated by G_0) than its G_3 counterpart, except for Observations #1, #4, and #5 in the figure, because it takes the measures of the Doppler sensor into consideration. The observed outcomes give rise to the average accuracy of some 91%, obtained according to

$$\overline{Accuracy} = \frac{\sum_{i=1}^{n} \left[1 - \left|\frac{G_0 - G_2}{G_0}\right|\right]}{n} \times 100\%, \tag{10}$$

where n, the total number of trials, equals 17. As a result, including the Doppler sensor in our implemented monitor system is favorable, likely to yield better sand production estimation.

2) *Timing:* To illustrate the key feature of our FPGA-based system in its ability for providing sand production estimation in real time based on gathered sensor measures, we calculate the time required to process those 41,943,040 acoustic measures (sampled at 1.2 M/sec) within a time window of some 34.95 s (= T_2), according to the expression next

$$T_1 = (51 + 15 \times (m - 1)) \times t_{clk}, \tag{11}$$

where m equals 41,943,040 and t_{clk} is the operational clock duration (= 1/50 MHz = 20 nanoseconds) of our implemented system on the FPGA. The above expression gives rise to $T_1 \approx$ 12.58 s, clearly signifying that the implemented system can provide sand production estimates on-the-fly in real time.

B. Statistical Analyses of Established Prediction Models

For statistical assessment and contrast among different prediction models (G_1, G_2, and G_3) derived by the multiple regression analysis on sensor measures as stated in the prior subsection, we carried out 77 additional experimental runs under various flow rates and sand injection amounts (which were all different from those used earlier for establishing those

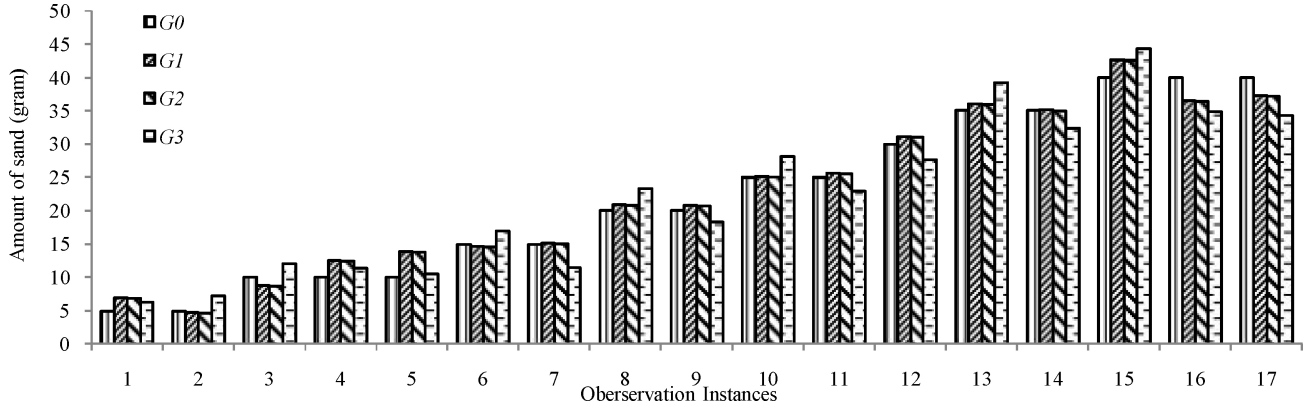


Fig. 15. Comparison among different prediction models over 17 observation instances.

TABLE II
STATISTS FROM MULTIPLE REGRESSION ANALYSIS

Parameter	Model G_2	Model G_3
Multiple-R^2	0.951	0.859
Standard Error	4.352	5.181
P-value (all)	0	0
Significance F	0.000	0.000

three models present in Section V.A) in order to examine four major testing parameters: Significance test, ANOVA test, and Linearity test, and Normality test. Results gathered from those 77 extra experimental runs aim to characterize the testing parameters of established prediction models, as tabulated in Table II. The multiple-R^2 value of G_2 listed in the table is seen to be extremely close to 1.0, indicating at (8) provides very accurate estimation on the sand amount in the pipeline of our testbed pump skid. The same holds true for G_1, which yields very similar results (omitted in Table II for simplicity) as those of G_2. By contrast, the multiple-R^2 value of G_3 is relatively smaller, reflecting its inferior accuracy in sand amount prediction due to the lack of sand velocity data made available by the Doppler sensor. Likewise, the standard error in Table II indicates that G_2 outperforms G_3, achieving better prediction accuracy, thanks to the use of the Doppler sensor.

1) *Significance Test:* The P-value is to indicate if the results are statistically significant: if it is less than the significant level of α (often selected as 0.05 [25]), the results are significant. Given the P-value of all three factors (intercept, N, and SS) are equal to 0 in Table II, G_2 is deemed statistically significant for the prediction model.

2) *ANOVA Test:* The analysis of variance (ANOVA) test is usually employed to determine the appropriateness of the multiple regressions by examing significance F. As shown in Table II, the value of significance F for G_2 is seen to approach 0.0, revealing that G_2 gives rise to accurate estimation for the dependent variable (which is the sand amount).

3) *Linearity Test:* As the multiple regression analysis assumes that the relationship between the dependent variables and the independent variables is linear [26], it is imperative to inspect the linearity of related variables. The sum of the square value versus the sand injection amount is depicted in Fig. 16,

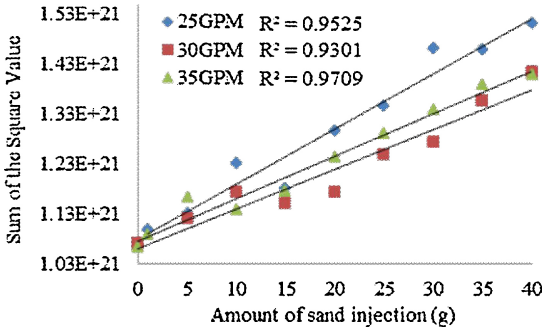


Fig. 16. Sum of the square value versus injected sand amount.

where each data point is the average of three runs under the same flow rate and the same sand amount. As can be observed in the figure, strong linear relationship exists between the two variables for every flow rate examined, indicating again the high accuracy of G_2.

4) *Normality Test:* The regression analysis assumes that variables have a normal distribution, which can be validated by a normality test based on the residual histogram [27]. A residual r refers to the difference (in the absoulate value) between the predicted and the actual amount of sand injected, expressed by

$$r = |G_2 - G_0|. \tag{12}$$

In other words, the residual indicates a prediction error amount, which varies from one run to another. Those total 77 runs represent 29 different cases (of sand amount and flow rate combinations, with repeated runs to get the mean predicted sand amount for a given case). The residues of the 29 cases were grouped into seven intervals according to their values. The residual histogram, depicted in Fig. 17, demonstrates the occurrence frequencies of those seven residual intervals. A symmetric bell-shaped histogram is observed in the figure, with a normally distributed asymptotic (dashed) curve confirming that the predicated sand amounts possess a normal distribution.

C. Resources Utilization and Overall Cost

The sand monitoring system is implemented on the Xilinx XC3S700AN FPGA board, priced at $243. The total cost of

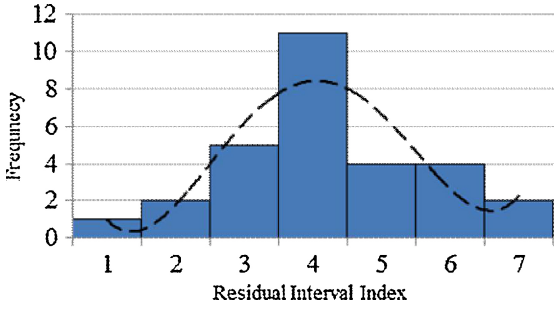


Fig. 17. Residual histogram of predicted sand amounts.

TABLE III
FPGA ON-BOARD RESOURCE UTILIZATION RESULTS

Resources	Used	Avail.	Utilization
# of 4 input LUTs	5,341	11,776	45%
# of bonded IOBs	69	372	18%
# of BUFGMUXs	7	24	29%
# of DCMs	3	8	37%
# of MULT 18×18 SIOs	4	20	20%
# of RAMB 16BWEs	4	20	20%

our system, including two AE sensors (of $300 each) and one Doppler sensor (of $1,390), approximates $2,230. The available and consumed resources of the FPGA board are shown in Table III, where the whole sand monitoring system implementation takes up only 45% of available LUT cells. As a result, more features may be added to the adopted Xilinx FPGA board to realize future monitoring systems. Given a commercially available sand monitoring system (which includes acoustic sensors, a PC to run digital signal processing and sand production estimation software, plus a data acquisition unit), often costs some $20,000 to $25,000 [3, 4, 5], our prototyped system on the FPGA has a price tag of about 1/10 of its commercial counterpart. This huge cost advantage is to make the developed sand monitoring system affordable, besides its high accuracy and its real-time estimation of the sand production amount.

VII. CONCLUSION

A real-time and cost-effective sand monitoring system has been realized on the Xilinx Spartan-3AN FPGA Evaluation Board, involving two AE (acoustic emission) sensors and a Doppler sensor. It estimates the sand production amount of the oil and gas flow in a production pipeline reliably, resulting from surrounding noise avoidance using the difference of the two AE measures. By approximating the multiple regression analysis closely to facilitate FPGA implementation, the prediction model given by (8) is found to yield accurate sand production estimation, with the multiple-R^2 parameter over gathered sensor measures equal to 0.951 which confirms its high prediction accuracy. When compared with its counterpart without a Doppler sensor, our monitoring system tends to produce better sand production estimation, as a result of taking the mean sand velocity (which is different from the flow speed) into consideration. Costing about 1/10 of that of a commercially available system and attaining improved accuracy to some 91% from 85% [14], our FPGA-based monitoring system stands ready for widespread adoption in the field to avoid catastrophes resulting from sand damage to the oil/gas production pipeline and related equipment.

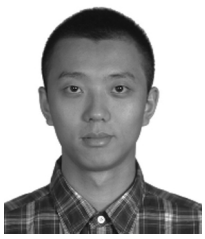
ACKNOWLEDGMENT

The authors would like to thank Mohammad Kazemi and Wade Falcon for their assistance in our early system development.

REFERENCES

[1] Stephen P. Mathis, "Sand management: A review of approaches and concerns," in *Proc. SPE Eur. Formation Damage Conf., Soc. Petroleum Eng.*, May 2003, paper SPE 82240.
[2] M. Salama, "Performance of sand monitors,"in *Proc. Int. Conf. Corrosion*, Mar. 2000, paper 00085.
[3] Sand Monitoring Services, Ltd. (2010). *Sand Monitoring Solutions* [Online]. Available: http://www.sand-monitoring.co.uk/
[4] M. Kazemi, "Accurate non-intrusive sand measurement during oil and gas production," M.S. Thesis, Dept. Electr. Comput. Eng., Univ. Louisiana at Lafayette, USA. May 2011.
[5] *Ultrasonic Intelligent Sensors ClampOn DSP-06 Particle Monitor Digital Signal Processing*, ClampOn, Inc., Houston, TX, USA, 2011.
[6] S. Pan, "Evolutionary computation on programmable robust IIR filter pole-placement design," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 4, pp. 1469–1479, Apr. 2011.
[7] L. Contreras-Medina, R. de Jesus Romero-Troncosco, E. Cabal-Yepez, J. de Jesus Rangel-Magdaleno, and J. R. Millan-Almaraz, "FPGA-based multiple-channel vibration analyzer for industrial applications in induction motor failure detection," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 1, pp. 63–72, Jan. 2010.
[8] T. Buerkle, J. B. LaMeres, T. J. Kaiser, E. Gowens, L. Smoot, T. Heetderks, K. Schipf, L. Clem, S. Schielke, and R. Luhr, "Ionizing radiation detector for environmental awareness in FPGA-based flight computer," *IEEE Sensors J.*, vol. 12, no. 6, pp. 2229–2236, Jun. 2012.
[9] M. Abdallah, O. Elkeelany, and A. Alouani, "A low-cost stand-alone multichannel data acquisition, monitoring, and archival system with on-chip signal preprocessing," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 8, pp. 2813–2827, Aug. 2011.
[10] J. Rangel-Magdaleno, R. Romero-Troncoso, R. A. Osornio-Rios, E. Cabal-Yepez, and A. Dominguez-Gonzalez, "FPGA-based vibration analyzer for continuous CNC machinery monitoring with fused FFT-DWT signal processing," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 12, pp. 3184–3194, Dec. 2010.
[11] R. Romero-Troncoso, M. Pena-Anaya, E. Cabal-Yepez, A. Garcia-Perez, and R. A. Osornio-Rios, "Reconfigurable SoC-based smart sensor for wavelet and wavelet packet analysis," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 9, pp. 2458–2468, Sept. 2012.
[12] M. Shahbazi, P. Poure, S. Saadate, and M. R. Zolghadri, "FPGA-based fast detection with reduced sensor count for a fault-tolerant three-phase converter," *IEEE Trans. Ind. Informat.*, to be published.
[13] Z. Milivojeviæ. *Digital Filter Design* [Online]. Available: http://www.mikroe.com/eng/products/view/268/digital-filter-design/
[14] *FIR Compiler Product Specification v5.0*, Xilinx, Inc., San Jose, CA, USA, 2009.
[15] J. P. Deschamps *et al.*, "Arithmetic operations: Multiplication," in *Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems*, NJ: Wiley-Interscience, 2005, ch. 5, sec. 1, pp. 83–86.
[16] Y. C. Lim, R. H. Yang, D. Li, and J. Song, "Signed power-of-two term allocation scheme for the design of digital filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 5, pp. 577–584, May 1999.
[17] *Spartan-3A/3AN FPGA Starter Kit Board User Guide (UG334) v1.1*, Xilinx, Inc., San Jose, CA, USA, 2008.
[18] *Ultrasonic Intelligent Sensors ClampOn SandQ™ Monitor Digital Signal Processing*, ClampOn, Inc., Houston, TX, USA, 2012.
[19] T. Folkestad and S. Mytvaganam, "Acoustic measurements detect sand in north sea flow lines," *Oil Gas J.*, vol. 88, no. 35, pp. 33–39, Aug. 1990.

[20] I. A. Allahar, "Acoustic signal analysis for sand detection in wells with changing fluid profile," in *Proc. SPE Latin Amer. Caribbean Petroleum Eng. Conf. Soc. Petroleum Eng.*, Apr. 2003, paper SPE 81002.
[21] H. M. Ibrahim, "Optimum procedure for calibrating acoustic sand detector, gas field case," in *Proc. Canadian Int. Petroleum Conf.*, Jun. 2008, paper 2008-0-25.
[22] *Sono-Trak Doppler Ultrasonic Flow Meter User Manual*, Advanced Energy Industries, Inc., Fort Collins, CO, 2004.
[23] *Block Memory Generator Product Specification v3.3*, Xilinx, Inc., San Jose, CA, USA, 2009.
[24] Xilinx, Inc. *DDR2 MIG Reference Design* [Online]. Available: http://www.xilinx.com/products/boards/s3astarter/reference$_{-}$designs.htm
[25] T. Siegfried, "Odds are, it's wrong: Science fails to face the shortcomings of statistics," *Sci. News*, vol. 177, no. 7, pp. 26–29, Mar. 2010.
[26] A. Choudhury. (2009). *Multiple Regression Analysis* [Online]. Available: http://www.experiment-resources.com/multiple-regression-analysis.html
[27] T. Hill and P. Lewicki, *Statistics: Method and Applications*, 1st ed. Tulsa, OK, USA: StatSoft, 2005.



Xi-Yue Xiang received the B.E. degree in electrical engineering from Xi'an Shiyou University (XSYU), Xi'an, China, in 2008, and the M.S. degree in telecommunication from the University of Louisiana at Lafayette, LA, USA, in 2012. Currently, he is pursuing the Ph.D. degree in computer engineering at the Center for Advanced Computer Studies, University of Louisiana at Lafayette.

From 2008 to 2010, he was with China National Offshore Oil Corporation (CNOOC), China, as an Electrical Engineer. His current research interests include FPGA-based embedded system design, computer communications and networks, and VLSI circuit design.

Itthichok Jangjaimon (SM'09) received the B.E. degree in computer engineering from Kasetsart University, Bangkok, Thailand, in 2004, and the M.S. degree in computer science from the University of Louisiana at Lafayette, LA, USA, in 2008. Currently, he is pursuing the Ph.D. degree at the Center for Advanced Computer Studies, University of Louisiana at Lafayette.

From 2004 to 2006, he worked on grid infrastructure establishment at the Thai National Grid Center, Thailand. He was also a Visiting Researcher at the National Institute of Advanced Industrial Science and Technology, Japan, in 2004. His current research interests include peer-to-peer systems, network coding, and distributed systems with an emphasis on energy efficiency and fault tolerance.

Mohammad R. Madani received the B.S. and M.S. degrees from Texas Tech University, Lubbock, TX, USA, and Southern Methodist University, Dallas, TX, USA, respectively, and the Ph.D. degree in electrical engineering from Louisiana State University (LSU), Baton Rouge, in December 1990.

He is currently the Director of Microelectronic Research Laboratory, and a Distinguished Associate Professor at the Department of Electrical and Computer Engineering, University of Louisiana at Lafayette. He has seven years of industrial experience in VLSI device and fabrication, working as a Senior Research and Development Engineer, Yield Enhancement Engineer, and Test and Characterization Engineer at Mostek Corporation, Carrollton, TX, USA. His current research interests include microelectronic circuits, sensors, MEMS processing, and sensor applications.

Nian-Feng Tzeng (M'86–SM'92–F'10) has been with the Center for Advanced Computer Studies, University of Louisiana at Lafayette, LA, USA, since 1987. His current research interests include the areas of computer communications and networks, high-performance computer systems, and parallel and distributed processing.

Dr. Tzeng was on the Editorial Board of the IEEE TRANSACTIONS ON COMPUTERS from 1994 to 1998 and on the Editorial Board of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS from 1998 to 2001. He was elected to Chair Technical Committee on Distributed Processing, the IEEE Computer Society, from 1999 to 2002. He was the recipient of the Outstanding Paper Award of the 10th International Conference on Distributed Computing Systems, in May 1990, and of the University Foundation Distinguished Professor Award in 1997.