

# Compressive Sampling and Real-Time Data Transportation in MPAR Backend System

Xining Yu  
Intelligent Aerospace Radar Team  
Advanced Radar Research Center  
University of Oklahoma  
Norman, OK  
xining.yu@ou.edu

Yan (Rockee) Zhang  
Intelligent Aerospace Radar Team  
Advanced Radar Research Center  
University of Oklahoma  
Norman, OK  
UAS@ou.edu

Mark E. Weber  
National Severe Storm Laboratory  
and CIMMS  
University of Oklahoma  
Norman, OK  
markw@ou.edu

**Abstract**—Massive data transportation has been an important bottleneck for the Multi-functional Phased Array Radar (MPAR), as the diversified waveforms require increasing sampling speeds for traditional quadrature sampling. In this work, we propose a method that “compresses” the quadrature (I/Q) data to reduce the data link bandwidth and transaction time between the receiver front-end and the processing unit of a phased array system.

**Keywords**—quadrature sampling; data compression; phased array system

## I. INTRODUCTION

A traditional coherent radar receive channel generates in-phase and quadrature data signals in either analog or digital forms. These signals are transported to the specialized signal processor units for pulse compression, detection and tracking. With the extensive usage of advanced waveforms, the bandwidth of the transmitting pulses can be quite large. Accordingly, based on Nyquist–Shannon sampling theorem, the ADC’s increasing sampling speed may introduce massive amount of data transactions. As an example, for a single dual-polarized channel, if the signal bandwidth is 20 MHz and ADC’s resolution is 12 bits, the transmitting rate should be at least 960 Mbps per channel. For an envisioned MPAR system with 200 dual-pol channels, the data rate at first line of beamformer can be higher than 192 Gbps. Even with advanced data link technologies today, this is still a tremendous challenge.

Currently, RapidIO, Ethernet and PCIe are the main options of fundamental data link protocols. RapidIO is reliable, efficient, and flexible. Compared with PCIe, which is optimized for hierarchical bus structure, RapidIO is designed for both point to point and hierarchical models. This feature can make the interconnection fabric more flexible. Further, RapidIO have a better flow control mechanisms than PCIe. At the physical layer, RapidIO offers PCIe-style flow control retry mechanism, which is based on tracking credits inserted in the packet headers. In addition, RapidIO includes a virtual output queue backpressure mechanism. Based on this capability, switches and endpoints can learn whether destinations are congested or not [1].

Even with these advanced data link technologies, the large communication bandwidth is still a tremendous challenge. In

order to further reduce the data stream rate without loss of information, we propose to incorporate compressive sampling (CS) concept into the MPAR backend system. According to the compressive sampling concept, when the signal matrix is sparse, we can sample the radar signal incoherently with a much slower rate than the Nyquist sampling rate [2], which may translate into saving of communication bandwidth, reduction of signal processors, and eventually lower costs. When we introduce the CS concept into array signal sampling, there are two specific issues we may pay attention to: (1) Robustness of signal recovery from noisy data, especially for the received signals before pulse compression. Indeed, CS processing can tolerate a proper level of noise. However, for actual radar data, useful signal may be immersed in the noise, which may lead to errors or distortions in signal recovery. (2) Recovery computing time and resources requirement, and whether it can fit in a digital receiver’s front-end needs to be concerned. Also, the computational resources required by CS processing, and the additional latency that adds in the MPAR receiver chain, should not offset too much the benefits it brought in for data transportation bandwidth reduction.

## II. MPAR BACKEND SYSTEM MODELING

### A. System Architecture

CS as well as data transportation protocols are evaluated on a generic phased array radar backend testbed, which consists of a realistic software simulator and a partially-implemented, small scale hardware DSP network. Figure 1 shows the top-level, generic MPAR backend system diagram. After the signal is sampled by ADC, board to board connections within a simulated chassis using RapidIO is used to transport the I/Q samples, which results in better performance than using Ethernet for the box-to-box interconnections. For a large scale phased array system such as MPAR, it may contain about 200 receive channels per face [3]. For such scale, we may combine the data from small number of channel first, and then transmit the entire face data out by using RapidIO through back panels. An illustration of the simulated interconnection fabric is also shown here. In our model, each front-end ADC board (which can be inserted in the back-panel) can contain several FPGAs. Each FPGA packs a number of channels of digitized data, and sends the packaged data through a data link. As operations of

these front-end boards are parallel and independent, we may easily adjust the numbers of front-end board inserted into chassis backplane according to the number of array channels. On the back-panel, a small number of high performance RapidIO switches can be used to handle the heavier traffic. In summary, this model represents a robust, scalable, and efficient generic backend system, and is compatible with current military and civilian open system standards.

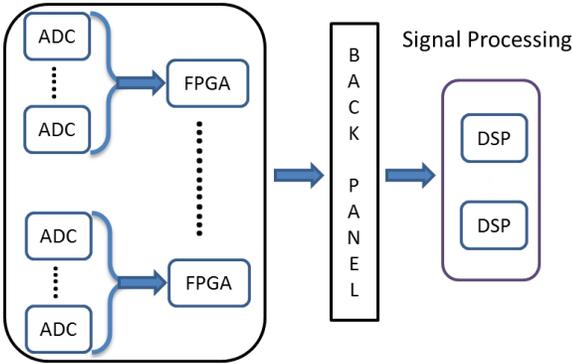


Figure 1: Top level diagram of the MPAR backend system.

A more detailed realization and breakout architecture for the current simulation testbed is shown in Figure 2. Three different types of backend boards are used in the fabric, and as the data moves more towards the back, more DSPs are utilized. The boards are: (1) Receiver front-end/data transmitting board, (2) DSP processing board, and (3) back panel. With current low-cost COTS components, the data transmitting board can capture up to 40 independent channels, and these signals are digitalized by ADC and packed up by FPGAs. On the back panel, 12 RapidIO switches can handle up to 2.8 Tera-bit per second data rate. The materials costs for an actual implementation of a 200-channel backend will be less than \$400K.

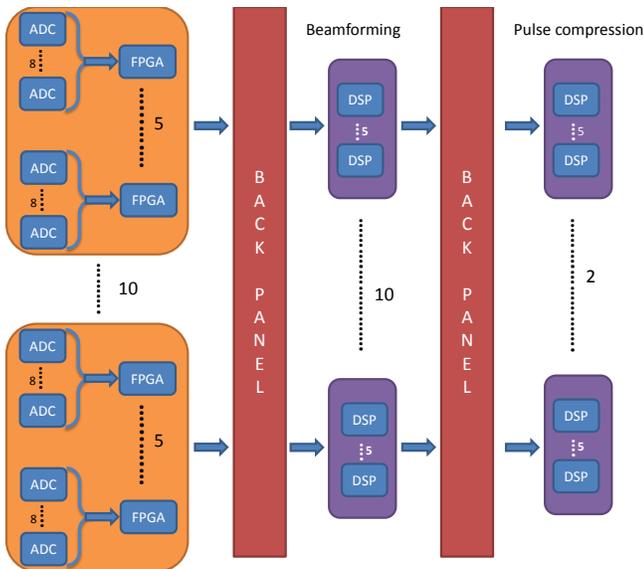


Figure 2: Break-out of back-panel system for the simulation testbed.

### B. Simulated Data transportation performance

Testbed simulation was done using MATLAB, Code Composer Studio (CCS) integrated development environment and RapidIO System Modeling Tool provided by Integrated Device Technology, Inc. In a small scale model, we simulated realistic sampling data traffic from RF front end to Beamforming DSP, as well as Beamforming DSP to Pulse Compression DSP. In this system, we have 24 C66x DSP cores, and 4 ARM cores to handle 24 channels with 8192 range gates. The data coming from 24 channels will be generated into 20 beams by 16 DSP cores, and then 8 DSP cores is responsible for Pulse Compression and Doppler Processing. ARM cores would handle the data traffic between Pulse Compression and Doppler Processing. Assuming Nyquist sampling speed at front-end, the estimated transmission time from a front end FPGA to Beamforming DSP is 904  $\mu$ sec for one pulse. This latency performance can meet most of the real-time surveillance requirements of NSWRC [4], however, limited signal bandwidth is assumed and only basic matched filtering algorithm is utilized.

One of the key challenges to achieve hard-real-time using embedded DSP processors is how to assign tasks to each DSP core. As the hardware resources on DSP core is limited, Beamforming weight vector is calculated from outside and transmitted to DSP core. For the rest of beamforming, which multiplies each channel's samples by weight vectors, DSPs needs  $\sim 1.5$  msec. After that, Pulse Compression needs another 1.5 msec to complete. The data transmission between Beamforming and Pulse Compression is done by using EDMA (Enhanced DMA), which could send or receive the data without interfering with DSP. With help of EDMA, calculation can be performed in parallel with data communication. As 20 beams cannot evenly divided into 8 DSP cores, so some of DSP cores would be free from Pulse Compression task. Doppler Processing is performed on those stall cores during this period. The data corner turn, in which rearrange range-aligned data to pulse-aligned, is performed by ARM core using EDMA, which costs 1.1 msec. The following Figure 3 sketches the real-time time-line for initial radar processing. Further processing will be then added, such as weather product generation and target tracking.

According to the Figure 3, the shortest PRI (Pulse Repetition Interval) being supported is  $\sim 1.5$  msec, or 667 KHz for PRT. Faster processing can be achieved by adding more parallel computing hardware (similar to the boards shown as in below). Calculation load of Doppler Processing is much less computational demanding compared to the Beamforming and Pulse Compression.

As the front-end FPGAs and the DSPs have abundant resources that can be used to implement data encoding/compression and data decoding/recovery, we would consider a CS module added into both ends to help reduce the communication bandwidth. The testbed also simulates the realistic radar returns through each individual array channel, which supports evaluation of basic compression and evaluation performance in this work.

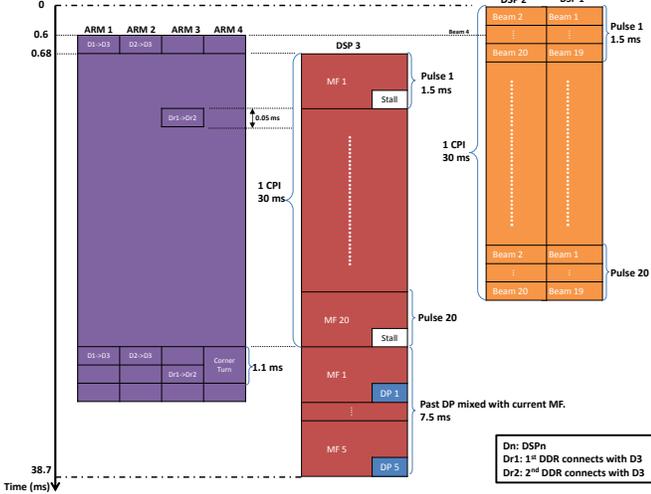


Figure 3: Small scale system timeline

### III. COMPRESSIVE SAMPLING

#### A. Introduction

Compressive sampling or CS has been introduced since 2006 [5]. This novel sampling method challenges the commonly well know Nyquist sampling rate that requires the sampling speed higher than two times of signal bandwidth [2], and it can recover certain signals by using fewer samples. For the concise representations of original signal, two fundamental premises should be met: sparsity and incoherence. For the sparsity, it means that when the signal is projected onto a suitable basis, a large number of coefficients of signal should be small enough to be ignored. For a certain signal, if it has  $s$  non-zero coefficients, it is said to be  $s$ -sparse. As  $s$  increases, it becomes harder to sense and reconstruct the original signal [6]. The incoherence implies that any two elements in the sensing basis  $\Phi$  and representation basis  $\Psi$  should have *low* coherence. The coherence between  $\Phi$  and  $\Psi$  is measured by

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq k, j \leq n} |\langle \phi_k, \psi_j \rangle| \quad (1)$$

In which  $\mu$  is the incoherence property,  $n$  is the number of elements in the original signal, and  $k, j$  are indices of the basis functions. In other words, the sampling and representation basis should be concerned as low coherence pairs. For example, we may choose spike basis  $\phi_k(t) = \delta(t - k)$  as sensing matrix, and Fourier basis  $\psi_j(t) = \sqrt{n}e^{-i2\pi jt/n}$  as representation basis [4].

Besides sparsity and incoherence, in order to analyzing the performance of different CS algorithms, restricted isometry property (RIP) is introduced. RIP characterizes isometry constant  $\delta_{2s}$  of a matrix such that

$$(1 - \delta_{2s})\|x_1 - x_2\|_{l_2}^2 \leq \|\Theta(x_1 - x_2)\|_{l_2}^2 \leq (1 + \delta_{2s})\|x_1 - x_2\|_{l_2}^2 \quad (2)$$

$\Theta$  is the reconstruction matrix, which is the product of  $\Phi$  and  $\Psi$ . If  $\delta_{2s}$  is sufficiently less than one, This implies that the all pairwise distance between  $s$ -sparse signals, such as vector  $x_1$

and  $x_2$ , can be well preserved in the measurement space. That means measurement matrix contains the sufficient information in signal of interest.

#### B. CS performance in array channels

Different application may have various requirements or limitation to use CS. In the communication system, it requires the CS algorithm for speedy spectrum sensing; in medical imaging processing, like magnetic resonance imaging (MRI), with benefits for patients economics, the scan time reduction is the thing researchers pay more attention to. In radar application, the signal to noise ratio (SNR) may be so low that the signal can be immersed within the noise; hence, robust signal recovery from noisy data is a crucial point for radar sampling. Figure 3 shows the mean square error (MSE) between reconstruction data and original signal (noise-free), and error compared to the original signal with noise. We can see the CS can actually suppress noise when SNR is low. This is because this signal (pulse) is sparse, and the noise is widely spread the entire spectrum, as a result, the reconstruction process would ignore those small variation produced by the noise.

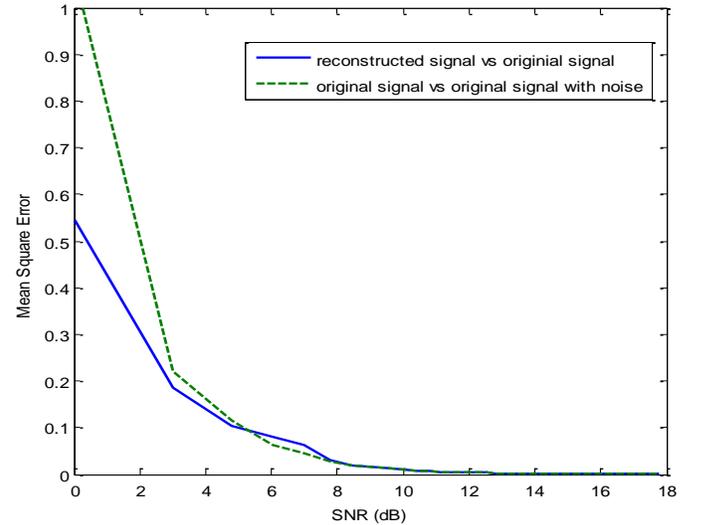
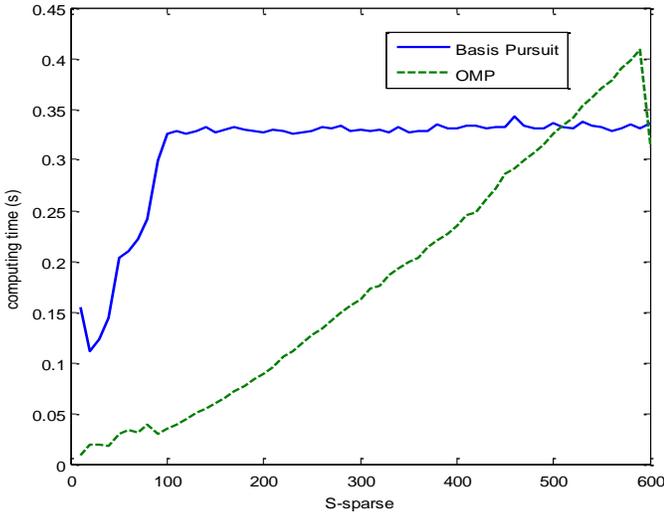


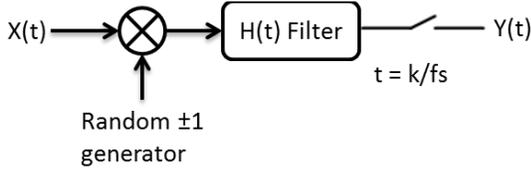
Figure 4: Reconstruction error vs SNR.

As SNR increase, the MSE decreases. From Figure 3 we can notice that when SNR is larger than 6 dB, the reconstruction data have the similar result as original data with noise. In other words, the compressive sampling can be used in the radar application even the SNR is low. Another important aspect for CS implementation is the algorithm efficiency. There are so many reconstruction algorithms existing, such as Basis Pursuit, Matching Pursuit, and Message Passing. Among those algorithms, the *greedy iterative algorithm* is easy to implement and has high speed of signal recovery. It solves the reconstruction problem by finding a optimal result iteratively. Within the framework of greedy pursuing, we select the Orthogonal Matching Pursuits (OMP) [7] as our key compressive sensing algorithm. For a signal with length  $n$  and  $s$  sparsity, OMP can reliably recover this signal by using  $O(s \log n)$  measurements. The complexity of OMP algorithm is

$O(sm)$ .  $m$  is the number of measurements. Figure 4 shows the comparison between the OMP and Basis Pursuit, where  $n=600$ , and  $m=4s$ . It can be seen that OMP have better performance than the basis pursuit. However, when the signal is not sparse, the recovery becomes costly.



**Figure 5: Computational time comparison of two CS algorithms on AMD Opteron 6128 (2GHz)/MATLAB regarding to different degrees of channel signal sparse (S).**



**Figure 6: Structure of a compressive sampler.**

### C. Structure of low rate sampling system

To facilitate implementation, canonical or spike basis can be used. A proposed sampling structure is shown in Figure 5. The low rate sampling system performs as random demodulation scheme [9]. The signal,  $x(t)$ , is multiplied by chipping sequence, which alternates between -1 and +1 at the Nyquist rate or higher. The purpose of this operation is to spread the baseband frequency content to the entire spectrum. And then, the altered signal goes through a bandpass filter centered at center frequency  $f_0$ . At last, ADC can sample the signal at the  $f_{cs}$ .  $f_{cs}$  is  $k$  times lower than the signal center frequency  $f_0$ .

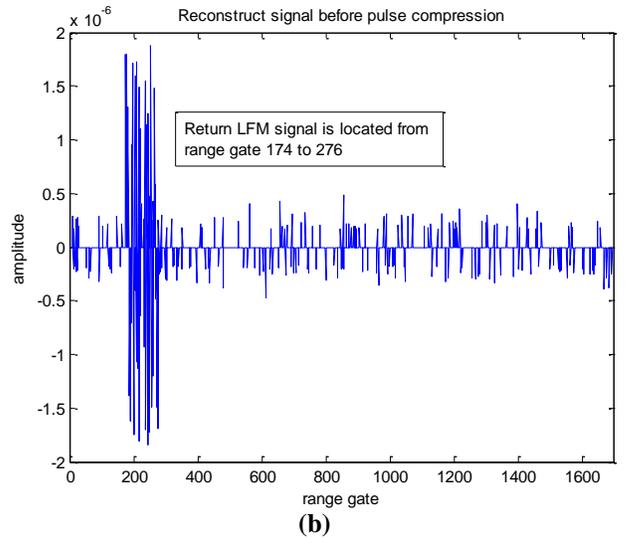
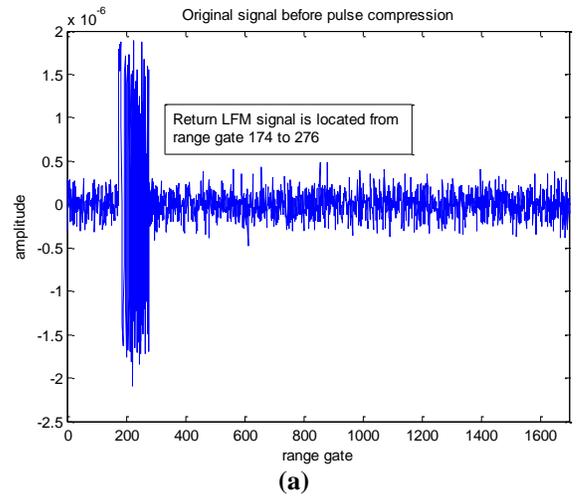
### D. Recovery from the undersampled data

Once the compressed signal samples reach the receiver end-node, the original signal can be recovered by using OPM. A simple simulation using backend testbed is shown here for illustration. Only one channel is used for this test, and the radar parameters are listed in Table 1. Figure 6 and 7 show the comparison between original received signal and reconstructed signal for a point target. Figure 6(a) shows the scenario when SNR is 9 dB before pulse compression. There is no significant difference between the original and reconstructed signals

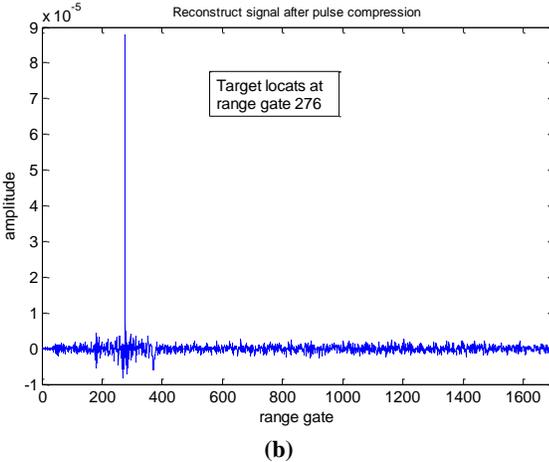
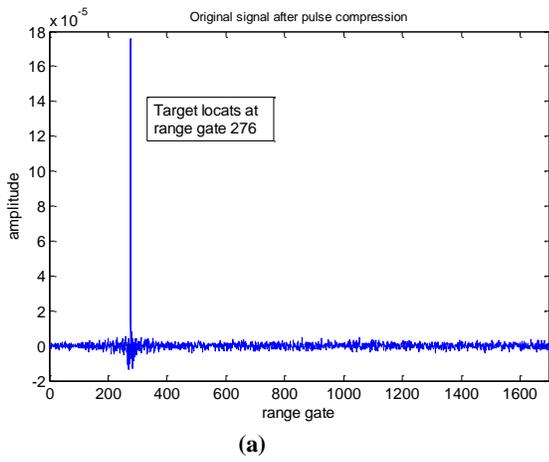
(noise shows some difference but not our concern here). In the second case shown in Figure 8 and 9, the SNR is down to -2 dB. Figure 7 and 9 also compare the reconstructed post-compression signals. In both cases, the target signature can be recovered, while better performance is expected for higher SNR.

**Table 1: Array channel simulation parameters**

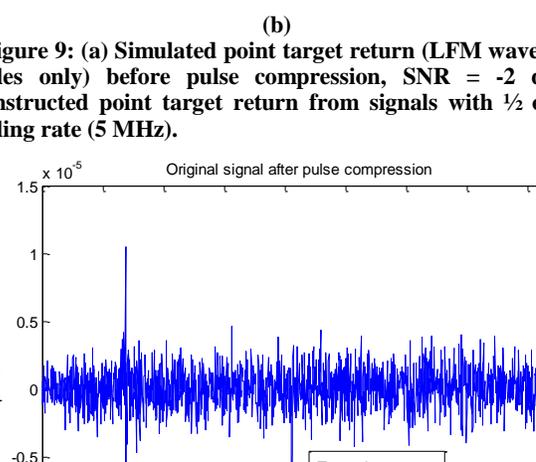
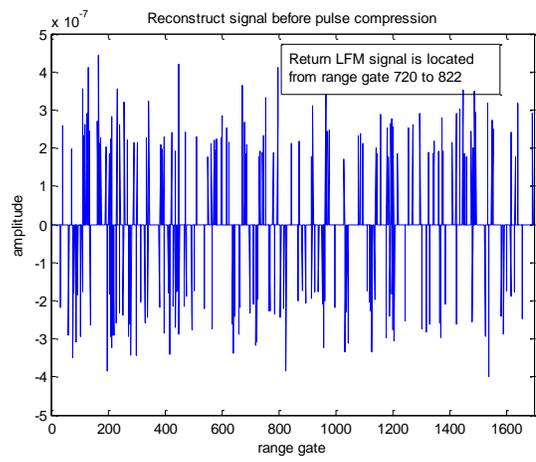
Parameter	Values
Pulse bandwidth	5 MHz
Pulse width	10 $\mu$ sec
PRF	6 KHz
Nyquist sampling rate	10 MHz
CS sampling rate	5 MHz
Number of range gates	1700
Number of range gates for LFM pulse	102



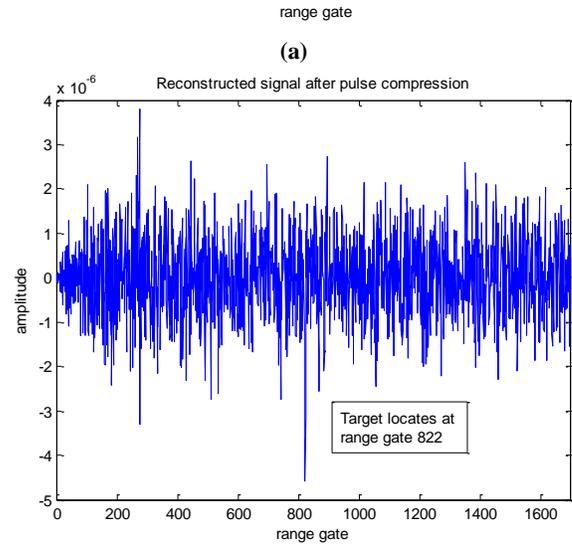
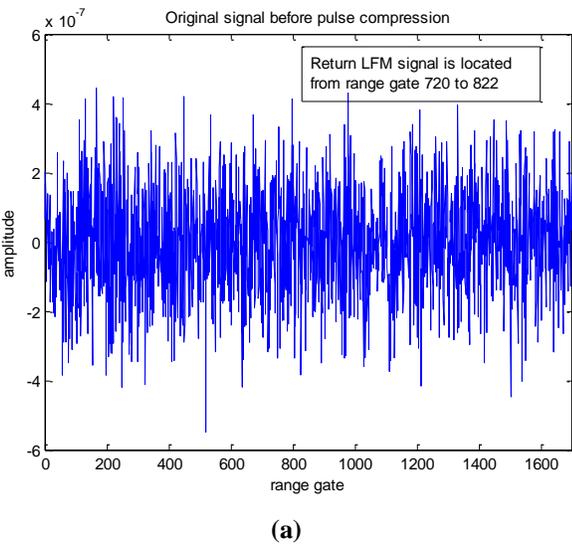
**Figure 7: (a) Simulated point target return (LFM waveform, I samples only) before pulse compression, SNR= 9 dB. (b) Reconstructed point target return from signals with  $\frac{1}{2}$  original sampling rate (5 MHz).**



**Figure 8: (a) Simulated point target return (amplitude) after pulse compression, pre-compression SNR= 9 dB. (b) Reconstructed pulse compression output from signals with  $\frac{1}{2}$  original sampling rate (5 MHz).**



**Figure 9: (a) Simulated point target return (LFM waveform, I samples only) before pulse compression, SNR = -2 dB. (b) Reconstructed point target return from signals with  $\frac{1}{2}$  original sampling rate (5 MHz).**



**Figure 10: (a) Simulated point target return (amplitude) after pulse compression, pre-compression SNR= -2 dB. (b) Reconstructed pulse compression output from signals with  $\frac{1}{2}$  original sampling rate (5 MHz).**

#### IV. CONSLUSIONS AND FUTURE WORK

The potentials of using compressive sampling technology in large scale MPAR to reduce the data transportation bandwidth requirements are being analyzed. Initial results using software backend modeling testbed are promising, while there are much more work can be done, such as (1) Trade analysis for more realistic, end-to-end tests using the backend testbed to precisely predict benefits vs overheads. (2) CS algorithms may be improved to better handle situations where SNR is low. (3) In order to reduce the computation loads for MPAR, we may transform the low rate sampling data into frequency domain first, measure them, and then reconstruct the data back into the frequency domain. This can reduce half of Fourier transform computation in the following pulse compression stage. (3) We may use available hardware DSP system to evaluate the CS performance in actual embedded processor, and demonstrate real-time radar data transportation applications.

#### REFERENCES

- [1] Barry Wood, "Backplane tutorial: RapidIO, PCIe and Ethernet," EE Times, Jan 2009.
- [2] Emmanuel J. Candès and Michael B. Wakin, "An Introduction To Compressive Sampling", IEEE Signal Processing Magazine, 2008, pp 21-30.
- [3] Mark Weber, John Cho, James Flavin, Jeffrey Herd, and Michael Vai, "MULTI-FUNCTION PHASED ARRAY RADAR FOR U.S. CIVIL-SECTOR SURVEILLANCE NEEDS", American Meteorological Society.
- [4] [https://www.fbo.gov/index?s=opportunity&mode=form&id=8ac04973ba91424476d63fba783dd951&tab=core&\\_cview=1](https://www.fbo.gov/index?s=opportunity&mode=form&id=8ac04973ba91424476d63fba783dd951&tab=core&_cview=1)
- [5] E. Candes, J. Romberg, and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information," IEEE Trans. Inform. Theory, vol. 52, no.2, pp.489-509, Feb. 2006.
- [6] S. Qaisar, R.M. Bilal, W. Iqbal, M. Naureen, and S. Lee. , "Compressive sensing: From theory to applications, a survey," Journal of Communications and Networks, vol.15, issue. 5, pp.1229-2370.
- [7] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Transactions on Information Theory, vol.53, no. 12, p. 4655,2007
- [8] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction", Information Theory, IEEE Transactions on, vol. 55, no. 5, pp2230-2249, 2009
- [9] J. Laska, S. Kirolos, M. Duarte, D. Baron, and R. Baraniuk, and Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," Proc. IEEE Int. Symp. Circuits Syst., New Orleans, LA, 2007, pp. 1959-1962