Passive radar for measuring passive sensors: direct signal interference suppression on FPGA using orthogonal matching pursuit and stochastic gradient descent

Jean-Michel Friedt[a], Weike Feng[b], Stéphane Chrétien[c], Gwenhael Goavec-Merou[a], Motoyuki Sato[d]

[a]FEMTO-ST Institute, Time & Frequency department, Besançon, France;
[b]Graduate School of Environmental Studies, Tohoku University, Sendai, Japan;
[c]National Physical Laboratory, London, UK;
[d]Center for NorthEast Asian Studies (CNEAS), Tohoku University, Sendai, Japan

ABSTRACT

Passive wireless sensors provide an attractive technology in niche applications where battery powered sensors are not applicable. Surface acoustic wave technology provides an optimum implementation of passive wireless transducers acting as cooperative targets to short range radar systems: the signal carrying the information is delayed beyond clutter, with a piezoelectric substrate converting the electromagnetic wave to an acoustic wave whose properties are dependent on the environment. We have tackled the issue of short range radar certification by considering a passive radar approach in which existing non-cooperative radiofrequency sources are used to power the sensor, and the reader is made solely of passive receivers aimed at recording the reference signal generated by the non-cooperative source and the backscattered signal returned by the sensor. A passive radar approach only requires a cross-correlation between the reference and surveillance signals to identify the time delay between the incoming and backscattered signals and hence the recovery of the physical quantity sensed by the acoustic transducer through acoustic velocity modulation. Practical sources do exhibit some short term correlation. Hence, strong copies of the reference signal with delays shorter than the one introduced by the sensor must be canceled by the receiver to allow for the weak sensor response to be extracted. This classical problem is called Direct Signal Interference (DSI) suppression. In a post-processing approach with little computation time limitation, this problem is solved using a least square error optimization approach. In the context of real time sensor measurement using a Field Programmable Gate Array (FPGA) implementation, data recording, frequency transposition and decimation are readily implemented in the gate array matrix. DSI removal appears as the limiting factor for a full FPGA implementation of the short range passive radar reader. We address the challenge by the iterative process of DSI suppression using Orthogonal Matching Pursuit (OMP) algorithm, and additionnally consider the FPGA-friendly Stochastic Gradient Descent (SGD) approach to try to recover DSI coefficients from a pipelined algorithm using streams of measurement data.

Keywords: Passive sensor, surface acoustic wave, short range passive RADAR, orthogonal matching pursuit, stochastic gradient descent, interference suppression

1. INTRODUCTION

Passive RADAR provides an attractive solution to multiple remote sensing issues including not relying on a dedicated emitter to illuminate targets bathed in electromagnetic smog,[1] and being able to detect signals returned by a cooperative target whose signature is representative on a physical quantity (identification, temperature, strain ...).[2] The basic signal processing principle of passive RADAR is to collect on the one hand a reference signal of the pseudo-random signal – most passive radar systems use existing commercial communication or navigation signals with appropriate random properties – illuminating the target, and on

Further author information: send correspondence to J.-MF, E-mail: jmfriedt@femto-st.fr

the other hand the surveillance signal reflected by the target, possibly frequency shifted by a Doppler offset frequency if the target is moving[3] (Fig. 1). Unlike active RADAR, the emitted continuous signal is not known so that must be observed by a reference channel (or reconstructed when a single receiver is used[4]) rather than locally memorized. The emitted signal might leak on the surveillance channel, inducing a strong background correlation signal that might be broad enough to overwhelm the signals reflected by the targets.

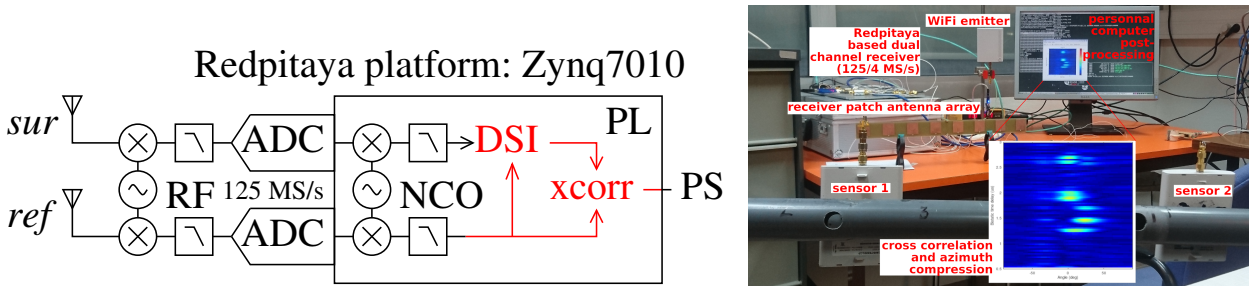

Figure 1. Left: data acquisition and processing principle: a Redpitaya board based Zynq System on Chip collects data from 125 MS/s analog to digital converters (ADC) following a radiofrequency to baseband frequency transposition (hardware mixer and filters). The sampled data are frequency transposed and filtered for adjacent WiFi channel data collection by the Programmable Logic (PL) part of the Zynq. At the moment, the resulting data are transferred to the Processing System (PS) to be sent to the host computer for further processing, limiting the measurement refresh rate. The present work aims (red) at including the DSI suppression and correlation in the PL. Right: experimental setup using an 8-patch antenna array as receiver for direction of arrival computation, making the DSI suppression load even heavier when probing multiple sensors. The antenna array (background) to sensor (foreground) distance is 1 m. The inset on the left is a screenshot of the computer display on the right exhibiting the azimuth-range map with the sensor echo delays (representative of the physical quantity) along the verical axis and the azimuth for spatially separating the sensor response along the horizontal axis.

The context of this study focuses on probing one particular case of targets designed as passive wireless sensors with echoes delay by 1 to 3 μs. A cooperative target illuminated by a non-cooperative emitter returns echoes whose delays are dependent not only on the bistatic range but also on some physical quantities of the sensor environment. Such a result is achieved by converting the incoming radiofrequency signal into an acoustic wave by patterning electrodes on a piezoelectric substrate in the classical Surface Acoustic Wave (SAW) transducer operation.[5] By a differential measurement in which multiple echoes induced by the acoustic transducer and returned by the sensor are sampled, subtracting delays between multiple echoes gets rid of the bistatic range delay dependence and provides means for remote sensing physical quantities using passive, cooperative targets probed by a passive RADAR receiver. The design of the SAW transducer aims at delaying echoes beyond clutter and environmental reflections: considering the receiver sensitivity, delaying echoes by 1 to 2 μs (150 to 300 m monostatic range) guarantees that the sensor response is well beyond any non-cooperative target reflection (Fig. 2).

Since the classical estimator for detecting echoes in a RADAR signal is the correlation, passive RADAR target identification heavily relies on correlating an unknown but measured reference signal from a non-cooperative illuminator with the surveillance signal collected by an antenna (or antenna array in our case) directed towards the sensor. However, any structure in the illuminating signal as often found in communication signals will depart from the random structure whose correlation would be a Dirac function and will introduce some long term memory effects with strong correlation side lobes. Thus, even if clutter has faded out by the time the sensor echoes are detected, the illuminating signal source autocorrelation might still overwhelm the sensor signal as will be demonstrated in the case of WiFi. This Direct Signal Interference (DSI) removal is the core requirement to extract the target (here cooperative with the sensor designed on purpose) signature from the correlation between the reference signal and the cleaned surveillance signal.

The classical approach to suppress this DSI signal is to use a least square weighing approach where time shifted copies of the reference signal are sought in the surveillance signal.[6] Assuming the echoes generated

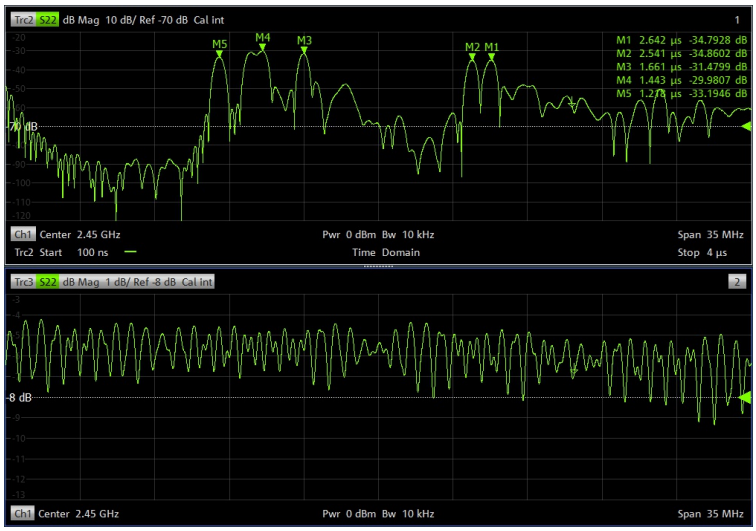

Figure 2. Surface Acoustic Wave (SAW) reflective delay line scattering coefficient response, bottom in the frequency domain and top in the time domain as computed through the inverse Fourier transform of the frequency domain characterization. Echoes are delayed by 1 to 2.5 μs beyond clutter of short range RADAR readers, while the actual delays encode an 8-bit identifier. The precise time delays between echoes informs on the sensor temperature through the acoustic wave velocity. The narrow bandwidth (35 MHz) was selected on purpose as representative of the passive RADAR bandwidth despite preventing the separation of two of the echoes.

by targets – whether non-cooperative or cooperative sensor targets for which the echoes are designed to lie beyond clutter – are delayed by more than a minimum time offset T, all copies of the reference signal delayed between 0 and T in the surveillance signal with estimated weights are subtracted before cross-correlating the reference and the surveillance signal to identify target. Practically, this operation is achieved by creating a matrix X with time delayed copies of the reference signal ref and identifying the weight w of the time-delayed copies of ref in the surveillance signal sur as

$$X = \begin{pmatrix} ref_1 & 0 & 0 & ... & 0 \\ ref_2 & ref_1 & 0 & ... & 0 \\ ref_3 & ref_2 & ref_1 & ... & 0 \\ ... & ... & ... & ... & ... \\ ref_N & ref_{N-1} & ref_{N-2} & ... & ref_{N-T} \end{pmatrix} : w = pinv(X) \cdot sur \tag{1}$$

where the pseudo-inverse $pinv(X)$ is $(X^H \cdot X)^{-1} \cdot X^H$ with ^{H} the conjugate transpose of the matrix. In this definition of X, signals before the first sample (ref_1) are assumed to be zero. A typical case is N large (numerous samples to improve signal to noise ratio) and T small (e.g. in a cooperative target short range RADAR system sampling the baseband at 100 MHz, removing clutter delayed by up to $T = 200$ ns (30 m monostatic range) is achieved with discrete delays from 0 to 20). The challenge of practically achieving this computation (Eq. 1) is that $X^H \cdot X$ is an $(T+1) \times (T+1)$ matrix which must be inverted before being multiplied by the $(T+1) \times N$ X^H matrix, resulting in a $(T+1) \times N$ matrix multiplied by the N element long surveillance signal to yield the $T+1$ weights w of the contribution of the delayed reference signal in the surveillance (Fig. 3). Subtracting the delayed reference from the surveillance is finally achieved with $sur - X \cdot w$. For example,[7] uses a Cholesky decomposition to compute the inverse, and hence does not benefit from the history of the matrix assembly.

One initial approach to reduce the computational burden is to reduce the number N of samples by selecting a random subset of N' lines: the resulting under-sampled copy of X is named X'. This number of sub-samples is dependent on the targeted accuracy and rises as the logarithm of the number of delays

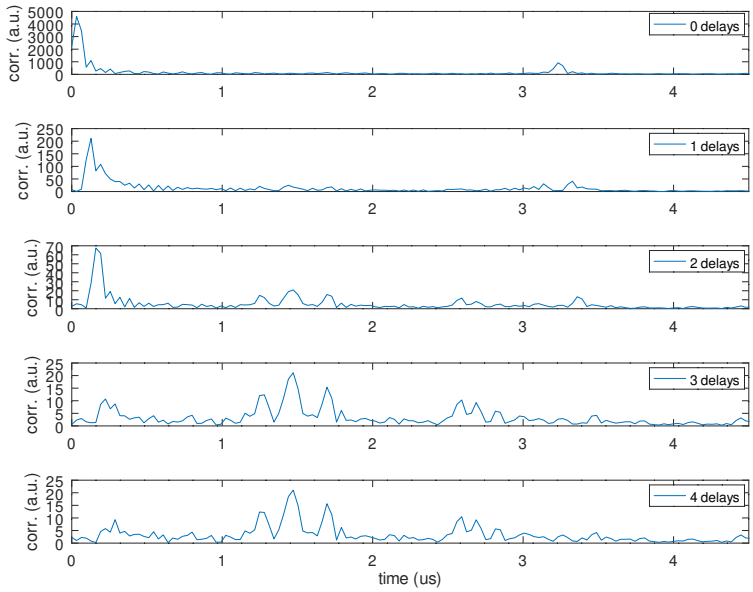

Figure 3. Passive RADAR interrogation of the SAW sensor characterized in Fig. 2 using a WiFi non-cooperative source, with the removal, from top to bottom, of none or 1 (sensor signal hidden under the sidelobes of DSI), 2 (sensor signal becoming visible between 1 and 3 μs), 3 (rejection of DSI and only sensor response left) and 4 (no improvement with respect to 3-sampling time delay subtraction) sampling delay copies of the normalized reference signal from the normalized surveillance signal. Such few delayed copies is consistent with the systematic analysis shown in Fig. 5. Note that, in the top three figures, the peaks at about 3.2 μs are generated by the properties of WiFi signal.

involved in the DSI.[8,9] In the experimental dataset shown in Fig. 3, only 3 time-delay copies (or 93 ns considering our sampling rate of 125/4=31.25 MS/s) of the reference signal need to be removed from the surveillance signal for the sensor to become dominant in the correlation between the reference and cleaned surveillance signal, while considering additional delays (Fig. 3, bottom chart) does not improve the sensor signal. Hence, only very few weights need to be computed to cancel DSI and allow for the sensor response to be analyzed.

Fig. 4 illustrates the strategy followed in this paper based on the DSI matrix representing the time-delayed copies of the reference signal sought in the surveillance signal. Here only positive delays are involved, but the practical algorithms consider positive as well as negative delays to account for correlation in the non-random reference signal. On the one hand a measurement set of the surveillance and reference channels has been collected and time-delayed copies of the reference are sought in the surveillance: the number of samples (lines) is fixed and the Orthogonal Matching Pursuit (OMP) algorithm iterates along the columns to sequentially identify and subtract the weighted copy of each time-delayed reference. On the other hand, the number of delayed copies of the reference signal is set and as more data are collected, the weights associated with each time-delayed copy of the reference are updated with the new observations: the Stochastic Gradient Descent (SGD) operates along the lines for a fixed number of delays (columns). Both algorithms aim at removing the computation burden of the matrix pseudo-inverse classically used to identify the weight of each time-delayed copy of the reference in the surveillance signal.

$$\begin{array}{c} \xrightarrow{\text{delays: OMP}} \\ \text{observations: SGD} \downarrow \begin{pmatrix} x_1 & 0 & \cdots & 0 \\ x_2 & x_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m-1} & \cdots & x_{m-T} \end{pmatrix} \end{array}$$

Figure 4: Direct Signal Interference matrix shaped by delaying the reference signal and search for the T weights (one for each column) of each copy in the surveillance signal.

This approach was used in[10] to demonstrate passive RADAR measurement of passive cooperative targets illuminated by a WiFi signal. However, the measurement update rate is limited by DSI suppression: the data acquisition, frequency transposition and filtering are processed in a pipeline running in the Field Programmable Gate Array (FPGA) of the Zynq system on chip (Fig. 1). Due to the computational burden of DSI suppression, the current implementation transfers the resulting reference and surveillance vectors to the host computer in charge of running the DSI suppression and cross-correlation. Since cross-correlation is readily achieved as a pipelined algorithm on an FPGA, running the full processing algorithm on FPGA for increased data acquisition rate requires solving the DSI suppression following an algorithm compatible with a pipelined approach applicable to a gate array.

2. ORTHOGONAL MATCHING PURSUIT FOR DIRECT SIGNAL INTERFERENCE REMOVAL

The computational burden of inverting the matrix remains high since it requires collecting the whole dataset before performing the DSI suppression. Here we consider the iterative subtraction process of subtracting sequentially the heaviest contributions of the reference signal in the surveillance channel, using the OMP algorithm.[11] Based on compressive sensing (CS) theory,[9] the DSI can be estimated by OMP algorithm with under-sampled measurement, which can help to reduce the computational complexity. Indeed, OMP implementation on FPGA has been demonstrated already[12–14] with reasonable computational power requirement with respect to those provided by the Zynq 7010 system on chip fitted on the Redpitaya platform we are using on our experiment.

In this algorithm, the contribution of each time delayed copy of X' in the under-sampled surveillance signal sur' is computed as $(X')^H \cdot sur'$ (Fig. 5). The time delay with highest contribution is selected at index p_1 and its weight is computed as $pinv(X'(p_1)) \cdot sur'$, where $X'(p_1)$ is the p_1th column of X' (appendix, first listing). Notice that in this first iteration, p_1 is a scalar so that $X'(p_1)$ is a vector and

$$pinv(X'(p_1)) \cdot sur' = \frac{< sur', X'(p_1) >}{||X'(p_1)||_2^2} \in \mathbb{C} \tag{2}$$

with $<,>$ the dot product $\sum_n sur'_n \cdot X'(p_1)^*_n$ since $pinv(X'(p_1)) = (X'(p_1)^H \cdot X'(p_1))^{-1} \cdot X'(p_1)^H$ and $X'(p_1)^H \cdot X'(p_1) = ||X'(p_1)||_2^2$.

In the next step of the OMP algorithm, the surveillance signal cleaned from this initial contribution of the reference signal $sur'_1 = sur' - X'(p_1) \cdot (pinv(X'(p_1)) \cdot sur')$ is then again analyzed for finding the next heaviest contribution of X' in sur'_1. This requires searching for the next heaviest contribution p_2 of X' in sur'_1 as the column for which $(X')^H \cdot sur'_1$ is maximized. It is known that subtracting only column p_2 of X' from sur'_1 will only converge to cleaning sur' from time delayed copies of the reference if p_1 and p_2 are different enough for the contributions to be independent. Practically, we expect the strongest contributions of the reference signal in the surveillance to lie at time delays close to 0 and not to be independent from each other. Hence, a global weight computation must be addressed by aggregating p_1 and p_2 in a vector p and identifying the optimal weight of both columns as $pinv(X'(p)) \cdot sur'$. This computation will grow in complexity towards the initial problem since p will grow as more contributions of the reference signal are sought in the surveillance. However in this iterative approach, $pinv(X'(p))$ can be deduced from the previous iteration by considering the block-wise inverse following the Sherman-Morrison formula:

$$(X'(p)^H \cdot X'(p))^{-1} = \begin{pmatrix} X'(p_1)^H \\ X'(p_2)^H \end{pmatrix} \cdot \begin{pmatrix} X'(p_1) & X'(p_2) \end{pmatrix}^{-1} = \begin{pmatrix} X'(p_1)^H \cdot X'(p_1) & X'(p_1)^H \cdot X'(p_2) \\ X'(p_2)^H \cdot X'(p_1) & X'(p_2)^H \cdot X'(p_2) \end{pmatrix}^{-1} \tag{3}$$

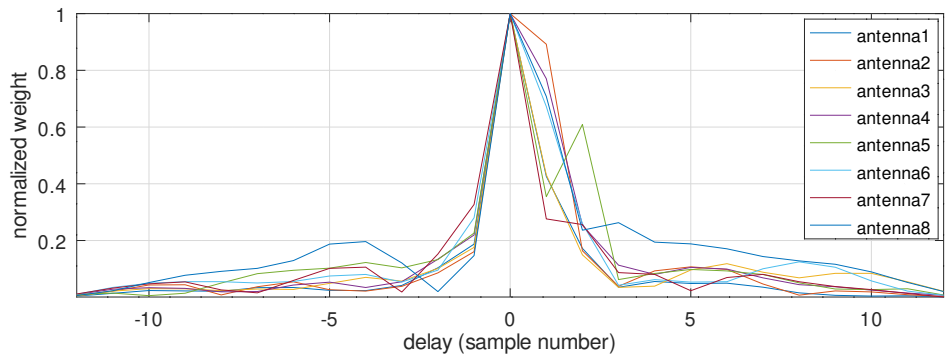

Figure 5. Experimental dataset analysis exhibiting the normalized weights of the delayed copies (delay in sample number as abscissa) of the reference in the surveillance channel. Notice that, as expected, weights are negligible in the negative delay and sharply drop to negligible values in the positive delays above 5 timesteps, consistent with Fig. 3.

2.1 Block matrix inversion

Since the new matrix inverse results from a previous iteration with the addition of one more vector from the reference signal to be subtracted from the surveillance, the problem of finding this new inverse can be expressed with the following shaped matrices:

$$\begin{pmatrix} A & B \\ B^H & C \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} + \frac{1}{k} A^{-1} B B^H A^{-1} & \frac{-1}{k} A^{-1} B \\ \frac{-1}{k} B^H A^{-1} & \frac{1}{k} \end{pmatrix} \tag{4}$$

where $k = C - B^H A^{-1} B$. Hence, in this equation A^{-1} is already known from the previous iteration, and only matrix products remain, no inversion (appendix, second listing).

The next section will compare both the least square estimate of the weights of the time-delayed copies of the reference signal in the surveillance signal using the pseudo-inverse (appendix listing 1) and the result of iterative inverse matrix computation (appendix listing 2) in which no explicit matrix inversion is needed but only matrix products. The data have been collected from an experimental WiFi passive RADAR setup with an 8-path antenna array receiver so that eight independent measurements of the the surveillance signals are analyzed.[2] The reference signal will either be used to synthesize known surveillance signals for algorithm assessment, or will be compared with experimental measurements of surveillance signals for sensor characteristics extraction.

2.2 Experimental demonstration

Using OMP and the iterative matrix inversion implementation, the DSI is indeed suppressed as shown in Fig. 6. In this figure, the result of DSI suppression using the pseudo-inverse (top) is favorably compared with the OMP result (middle) as emphasized by the difference between the two curves (bottom).

Thus, the iterative matrix inversion implementation of OMP is functional as an algorithm for extracting sensor echoes from the correlation between the DSI-cleaned surveillance signal and the reference signal. The implementation remains computationally intensive and requires keeping a memory of the whole reference and surveillance signals, using most resources on the low-end FPGAs (e.g. Zynq-7010 fitted on the Redpitaya board) considered for assembling an embedded reader with real time sensor echo detection capability.

3. STOCHASTIC ITERATIVE GRADIENT DESCENT WITH HARD THRESHOLDING

The OMP approach aims at identifying which time-delayed copy of the reference signal lies in the surveillance signal through an iterative process on the time delays, removing from the strongest to the weakest copy of

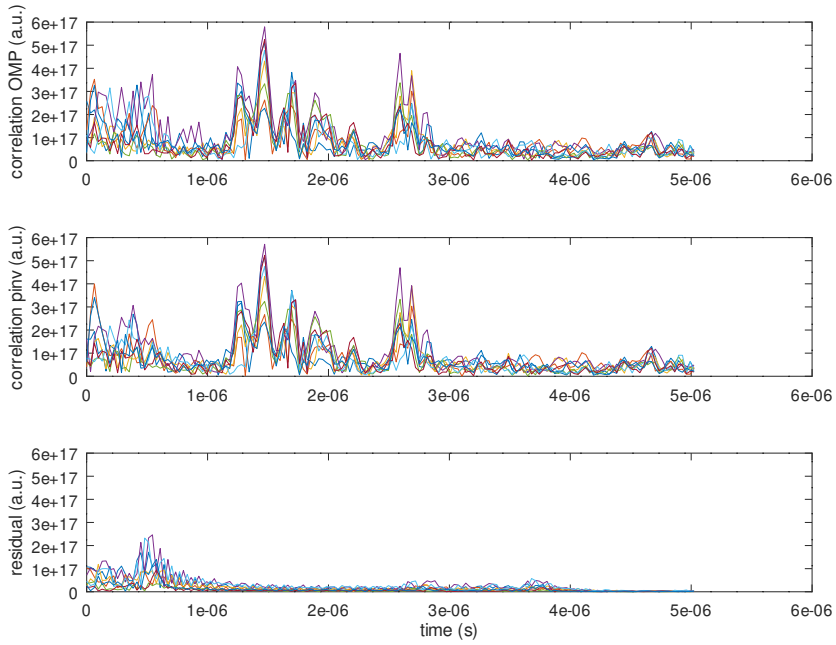

Figure 6. Comparison of the weight estimated through the pseudo inverse function (top) and through OMP (middle) for time delays ranging from -16 to +16 sampling intervals (-0.5 to +0.5 μs) for all 8 antenna measurements. The difference between the two results is shown as the bottom chart. In all cases the echoes of the sensors are visible on all 8-antenna signals between 1 and 3 μs, with some leftover DSI in the 0.5 to 1 μs delay range that does not impact our ability to recover the sensor measurement. The correlation peak at 3.2 μs is intrinsic to the WiFi signal structure and not part of the sensor response.

the reference signal in the surveillance signal. Thus, OMP assumes that the whole surveillance and reference datasets have been collected and stored in memory. Such an approach is poorly suited to FPGA implementation which is best designed for pipelined processing of new samples as they are collected. Furthermore, the iterative matrix inversion algorithm depicted earlier, despite removing the need for matrix inversion, still requires computing the scalar inverse $1/k$, a challenging operating on an FPGA.

Thus, such an approach does not address the practical measurement issues that 1/ the time delays and weights of the reference signal copies in the surveillance signal hardly vary from one measurement to another considering the fast (radiofrequency) sampling rate and 2/ data from the surveillance and reference channels are continuously streamed from the two analog-to-digital converter channels. Hence, as shown in Fig. 4, rather than addressing the DSI suppression issue along the time-delay (matrix column) direction, we consider the challenge of identifying time-delayed copies of the reference signal in the surveillance signal along the acquisition (matrix lines) dimension.

The SGD method[15] aims at solving the least-squares problem using an iterative optimization scheme which avoids matrix inversions and can proceed incrementally, yielding a highly scalable algorithm. In our setting, identifying the weights of the time delayed copies of the reference signal in the surveillance signal can be rephrased as a least squares problem

$$\min_{w \in \mathbb{C}^{T+1}} \frac{1}{2} \|sur - Xw\|_2^2 \tag{5}$$

The SGD algorithm[16] performs gradient steps by updating the weight vector w of the form

$$w^{(l+1)} = w^{(l)} + \alpha^{(l)} X_i^H (sur_i - X_i^t w^{(l)}) \tag{6}$$

where i is randomly selected among the observed time indices. Standard choices for the learning rate $\alpha^{(l)}$ include the constant step strategy and the $\alpha^{(l)} = 1/\sqrt{l}$ schedule. Throughout these investigations we have selected α ranging from $5 \cdot 10^{-3}$ to $3 \cdot 10^{-2}$ on synthetic data (Fig. 7) and $\alpha = 10^{-2}$ on experimental datasets (Figs. 8 and 9).

Our new approach to the problem combines the stochastic gradient scheme (6) with the Iterative Hard Thresholding Scheme of[17] by setting to 0 the weights below a given threshold defined as a fraction of the maximum of the weight values to accelerate the convergence of the algorithm.

$$w^{(l+1)} = T_{\lambda^{(l)}} \left(w^{(l)} + \alpha^{(l)} X_i^H (sur_i - X_i^t w^{(l)}) \right) \tag{7}$$

where T_λ is the thresholding operator that sets all values below the threshold λ to zero and $\lambda^{(l)}$ is chosen so as to select the values below 1% of the largest in absolute value.

Not only does the thresholded stochastic gradient descent method provide the weights of the DSI coefficients for future subtraction, it also allows for identifying the sensor echoes if a long enough horizon is allowed for the algorithm search. The assumption for stochastic gradient descent though is the sparsity of the weights, assuming few echoes in the returned signal (a well met hypothesis) spaced apart (an hypothesis to be discussed).

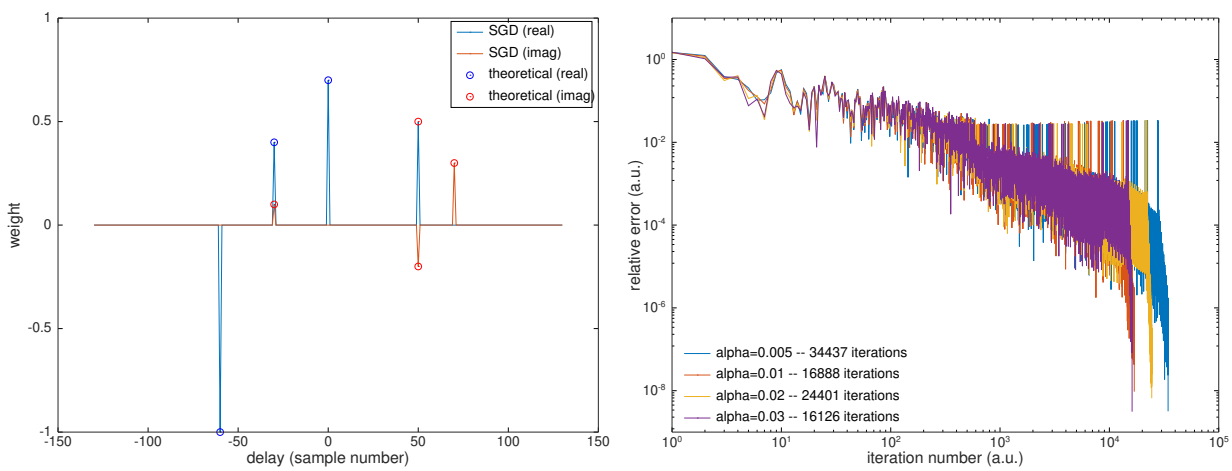

Figure 7. Left: stochastic gradient descend (SGD) applied to the synthetic dataset described in the text, with a hard threshold of $\lambda = 0.05$ times the maximum weight value. The red and blue circles indicate the complex weights applied to the delayed copies of the reference signal to synthesize the surveillance signal, emphasizing the excellent match between the targeted value and the weights identified by SGD. Right: evolution of the relative error (logarithmic scale) as a function of iteration number (abscissa – logarithmic scale) for various values of α. The iterations were stopped when the relative error variation reached 10^{-8}, indicating no evolution of the weights during the gradient descent iterations.

Fig. 7 demonstrates the application of the stochastic gradient method (appendix, listing 3: note the real implementation of the proposed method for complex values) to synthetic surveillance data generated by adding time delayed copies of experimental WiFi reference signals. In these examples, the time delays introduced were -60, -30, 0, 50 and 70 with arbitrary complex weights of representative of typical SAW echo delays since at a rate of 125/4=31.25 MS/s, an echo delayed by 1 to 2 μs is located at sample numbers 32 to 64. Fig. 7 (right) emphasizes the dependency of the convergence rate on the free parameter α which is here selected arbitrarily to yield convergence as quickly as possible: a rationale approach[18] to identifying α will be part of further investigations.

Hence, in the case of the iteratively thresholded stochastic gradient method, the objective might no longer DSI suppression to allow for the cross correlation, but sparse distributions of delayed copies of the reference signal in the surveillance signal position and weight identification. However, we have observed that sensor echoes are too weak to be identified by the SGD algorithm and are cancelled by the hard thresholding due to their low power with respect to the DSI. Indeed with a echoes 25 to 50 times weaker than DSI (Fig. 3), either the threshold λ is too high and cancels the weight of the sensor echo copies of the reference signal in the surveillance, or λ is too small at the level of the noise and prevents SGD from converging. Hence, the demonstration on experimental data remains on DSI weight identification (Fig. 8) followed by DSI subtraction and cross-correlation of the cleaned surveillance signal with the reference signal (Fig. 9) for sensor echo extraction.

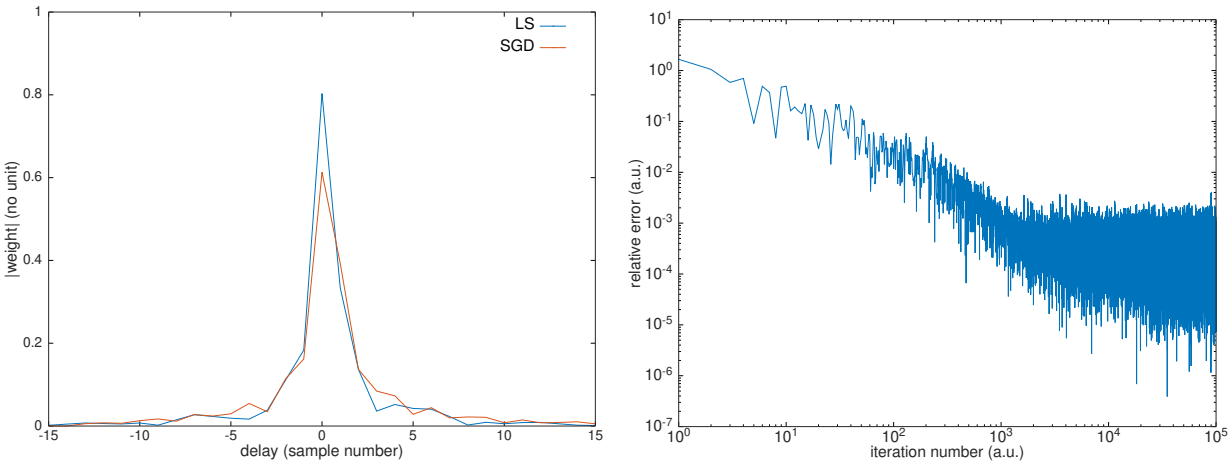

Figure 8. Stochastic Gradient Descent (SGD) applied to experimental WiFi passive RADAR signals collected as a SAW sensor was illuminated by a non-cooperative emitter. Left: the weights of DSI identified by SGD match the least-square (LS) analysis found by applying the pseudo-inverse of the time delayed copies matrix to the surveillance signal. Right: relative error (logarithmic scale) as a function of iteration number (logarithmic scale) demonstrating convergence in about 2000 iterations.

Fig. 8 demonstrates convergence of the SGD algorithm in about 2000 iterations. At a sampling rate of 125/4=31.25 MS/s or 32 ns/cycle, convergence will require 64 μs of data acquisition, well below typical time constants for temperature or chemical sensing using passive wireless SAW sensors. Furthermore, for faster physical quantities such as strain sensing,[19] the sensor environment is expected not to vary significantly over sub-100 μs durations and the DSI identification using SGD considered to be still valid for extracting sensor echoes whose phase is representative of stress levels in the sensor.

Fig. 9 (left) compares the cross-correlation of the cleaned surveillance signal with the reference signal either using the least-square solution (LS) obtained by applying the pseudo-inverse of the time-delayed copies of the reference signal matrix to the surveillance signal, or using the SGD algorithm. While both results exhibit excellent match, the SGD algorithm is well suited for implementation in an FPGA. In all these demonstrations, the classical random selection of the sample in the dataset when applying SGD is replaced with a sequential selection of each element of the dataset (cyclic buffer with the sample index selected modulo the dataset length – 32768 samples in these examples) representative of a continuous stream of data collected by the analog to digital converters connected to the surveillance and reference channels of the passive RADAR receiver. Running a C++ implementation of 2000 iterations of the SGD algorithm on the Cortex-A9 ARM processing system of the Zynq-7010 fitted on the Redpitaya board lasts 33.5 ms (168 ms for 10000 iterations), emphasizing the need for an FPGA implementation to achieve the 500-fold speed gain targeted for real time processing (Fig. 9, right).

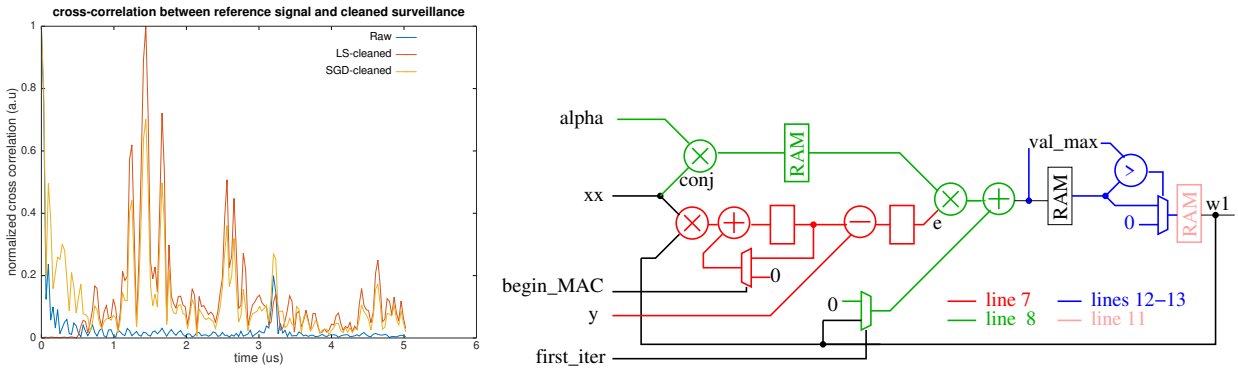

Figure 9. Left: comparison of the normalized cross-correlation of the raw surveillance signal and reference signal in which the sensor response is hidden below interference and WiFi autocorrelation structure, surveillance signal cleaned from DSI using the least square (LS, pseudo-inverse matrix calculation) classical technique and the proposed Thresholded SGD algorithm. Excellent match is observed on the sensor echoes using the latter techniques. Right: FPGA SGD with Hard Thresholding implementation principle. Colours refer to the line numbers in Listing 3 in the appendix. Iteration control logic is not shown on this chart focusing on arithmetic processing.

4. CONCLUSION

We have addressed the issue of real time direct signal interference removal in a passive RADAR setup designed to measure passive sensor responses. Since radiofrequency datastreams are collected using a field programmable gate array, algorithms compatible with such architectures are considered, namely with minimal memory consumption and lack of matrix inversion. We have demonstrated that an iterative matrix inversion applied to Orthogonal Matching Pursuit will remove the need for matrix inversion in the FPGA and restrict computations to vector multiplications. However, a scalar value inverse remains to be computed, a potentially challenging task in the FPGA hardly supporting division. As an alternative, we have explored the Stochastic Gradient Descent algorithm with Hard Thresholding which demonstrates comparable performance with an algorithm suitable for processing streams of data and adapting continuously the weights of the Direct Signal Interference for subtraction and correlating the cleaned surveillance signal with the reference signal.

ACKNOWLEDGMENTS

This work is supported by the OscillatorIMP ANR (French Agence Nationale de la Recherche) Equipex grant as well as the FAST-LAB Labcom ANR grant. J.-MF is grateful to Tohoku University (Sendai, Japan) for a visiting scientist grant during winter 2017 which prompted this investigation.

REFERENCES

[1] Griffiths, H. D. and Baker, C. J., [*An introduction to passive radar*], Artech House (2017).

[2] Feng, W., Friedt, J.-M., Goavec-Merou, G., and Sato, M., "Passive radar delay and angle of arrival measurements of multiple acoustic delay lines used as passive sensors," *IEEE Sensors Journal* **19**(2), 594–602 (2018).

[3] Kuschel, H., Cristallini, D., and Olsen, K. E., "Tutorial: Passive radar tutorial," *IEEE Aerospace and Electronic Systems Magazine* **34**(2), 2–19 (2019).

[4] Searle, S., Davis, L., and Palmer, J., "Signal processing considerations for passive radar with a single receiver," in [*2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*], 5560–5564, IEEE (2015).

[5] Morgan, D., [*Surface acoustic wave filters: With applications to electronic communications and signal processing*], Academic Press (2010).

[6] Garry, J. L., Baker, C. J., and Smith, G. E., "Evaluation of direct signal suppression for passive radar," *IEEE Transactions on Geoscience and Remote Sensing* **55**(7), 3786–3799 (2017).

[7] Rabah, H., Amira, A., Mohanty, B. K., Almaadeed, S., and Meher, P. K., "FPGA implementation of orthogonal matching pursuit for compressive sensing reconstruction," *IEEE Transactions on very large scale integration (VLSI) Systems* **23**(10), 2209–2220 (2015).

[8] Kane, D. M. and Nelson, J., "Sparser Johnson-Lindenstrauss transforms," *Journal of the ACM (JACM)* **61**(1), 4 (2014).

[9] Donoho, D. L. et al., "Compressed sensing," *IEEE Transactions on information theory* **52**(4), 1289–1306 (2006).

[10] Friedt, J.-M., Feng, W., Goavec-Merou, G., Martin, G., and Sato, M., "Passive RADAR acoustic delay line sensor measurement: demonstration using a WiFi (2.4 GHz) emitter and WAIC-band (4.3 GHz)," in [*WiSEE*], (2018).

[11] Tropp, J. A. and Gilbert, A. C., "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory* **53**(12), 4655–4666 (2007).

[12] Ge, X., Yang, F., Zhu, H., Zeng, X., and Zhou, D., "An efficient FPGA implementation of orthogonal matching pursuit with square-root-free qr decomposition," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2018).

[13] Morales-Pérez, C., Rangel-Magdaleno, J., Cruz-Vega, I., Ramirez-Cortes, J., and Peregrina-Barreto, H., "FPGA implementation of orthogonal matching pursuit algorithm," in [*2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*], 1–6, IEEE (2016).

[14] Polat, Ö. and Kayhan, S. K., "High-speed FPGA implementation of orthogonal matching pursuit for compressive sensing signal reconstruction," *Computers & Electrical Engineering* **71**, 173–190 (2018).

[15] Frotzscher, A. and Fettweis, G., "A stochastic gradient LMS algorithm for digital compensation of Tx leakage in Zero-IF-Receivers," in [*VTC Spring 2008-IEEE Vehicular Technology Conference*], 1067–1071, IEEE (2008).

[16] Bertsekas, D. P., "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," *Optimization for Machine Learning* **2010**(1-38), 3 (2011).

[17] Blumensath, T. and Davies, M. E., "Iterative hard thresholding for compressed sensing," *Applied and computational harmonic analysis* **27**(3), 265–274 (2009).

[18] Chrétien, S., Gibberd, A., and Roy, S., "Hedging parameter selection for basis pursuit," *arXiv preprint arXiv:1805.01870* (2018).

[19] Goavec-Merou, G., Chrétien, N., Friedt, J.-M., Sandoz, P., Martin, G., Lenczner, M., and Ballandras, S., "Fast contactless vibrating structure characterization using real time field programmable gate array-based digital signal processing: Demonstrations with a passive wireless acoustic delay line probe and vision," *Review of Scientific Instruments* **85**(1), 015109 (2014).

Appendix

OMP algorithm implementation using GNU/Octave's `pinv()` pseudo-inverse function:

Listing 1. Explicit matrix pseudo-inverse computation least square method

```
function [x,Ind]=omp(PSI_und,y,iter_max)
% PSI is the pre-defined under-sampled dictionary (delayed copies of the reference signal)
% y is the under-sampled received signal
% x is the coefficients of the selected column of PSI
% Ind is the index of the selected column of PSI
x=zeros(size(PSI_und,2),1);
res=y;Ind=[];
for iter=1:iter_max
    ll=PSI_und'*res;    % 1) Matched filter
    lg=abs(ll);         %    T+1 long vector
```

```
    [~,IX]=max(lg);     % 2) Column selection
    Ind=[Ind' IX]';     %    iter long vector
    Psi=PSI_und(:,Ind); % 3) Least square: length(PSI_und) x iter matrix
    weights=pinv(Psi)*y %    pinv() /!\  iter long vector
    res=y-Psi*weights;  % 4) Residual update
end
x(Ind)=weights;
```

OMP algorithm implementation (GNU/Octave) using block inversion function:

Listing 2. Iterative matrix inversion orthogonal matching pursuit method

```
function [x,Ind]=omp(PSI_und,y,iter_max)  % see parameter definition above
x=zeros(size(PSI_und,2),1);
res=y;Ind=[];
for iter=1:iter_max
   ll=PSI_und'*res;              % 1) Matched filter
   lg=abs(ll);                   %    T+1 long vector
   [~,IX]=max(lg);               % 2) Column selection
   Ind=[Ind' IX]';               %    iter long vector
   if (iter==1)
     Psi=PSI_und(:,IX);
     invA=1/(Psi'*Psi);          %    save 1/A
     weights=(Psi'*y)/(Psi'*Psi) %    this matches xx=pinv(Psi)*y
   else
     A=Psi'*Psi;                 %    |Psi(n-1)|
     b=Psi'*PSI_und(:,IX);       %    b=concatenation of new vector to old matrix
     c=abs(PSI_und(:,IX)'*PSI_und(:,IX));
     k=1/(c-b'*invA*b);          %    actually 1/k
     invA=[invA+k*invA*b*b'*invA -invA*b*k ; -k*b'*invA k]; % newA=[A b; b' c];
     Psi=PSI_und(:,Ind);
     weights=(invA*Psi')*y       % 3) Least square
   end
   res=y-Psi*weights;            % 4) Residual update
end
x(Ind)=weights;
```

SGD method with Iterative Hard Thresholding (IHT):

Listing 3. Stochastic Gradient Descent implementation for complex datasets

```
% assumes vector x to be reference channel and vector y to be surveillance -- returns w1 weights
for k = 1:number_of_iterations % typically 1000-10000
  i = mod(k,N-2*p-1)+1;      % should be random, but systematic iteration simulating datastream
  i = i+p;                   % allow for negative delays
  xx = [real(x(i-p:i+p)).',-imag(x(i-p:i+p)).';...
      imag(x(i-p:i+p)).',real(x(i-p:i+p)).'];% real and imaginay
  e = [real(y(i));imag(y(i))] - xx*w1;  % real and imaginary, since weights w1 are complex
  w1 = w1 + alpha  * xx.' * e;          % gradient descent
  w1_real = w1(1:end/2);
  w1_imag = w1(end/2+1:end);
  w1_com = w1_real +1j * w1_imag;
  I = find(abs(w1_com)<threshold*max(abs(w1_com))); % hard thresholding @ threshold*max(weights)
  w1_com(I) = 0;
  w1 = [real(w1_com);imag(w1_com)];
end
```