NDT&E International 50 (2012) 1–9

Contents lists available at SciVerse ScienceDirect

NDT&E International

journal homepage: www.elsevier.com/locate/ndteint

Online data compression of MFL signals for pipeline inspection

S. Kathirmani [a], A.K. Tangirala [a,*], S. Saha [b], S. Mukhopadhyay [b]

[a] Department of Chemical Engineering, Indian Institute of Technology Madras, India
[b] Control Instrumentation Division, Bhabha Atomic Research Centre, Mumbai, India

ARTICLE INFO

Article history:
Received 30 August 2011
Received in revised form
25 December 2011
Accepted 3 April 2012
Available online 4 May 2012

Keywords:
Magnetic flux leakage
Pipeline inspection
Mean Absolute Deviation
Discrete Wavelet Transform
Principal Component Analysis

ABSTRACT

The paper presents a novel three-stage algorithm for online compression of magnetic flux leakage (MFL) signals that are acquired in inspection of oil and gas pipelines. In the first stage, blocks of MFL signal are screened for useful information using a semi-robust statistical measure, Mean Absolute Deviation (μAD). The study presents guidelines for selecting a block size to deliver robust screening and efficient compression ratios. In the second stage, a multivariate approach is used to compress the data *across sensors* using Principal Component Analysis (PCA). The second stage is invoked only when an anomaly is detected by sufficiently large number of sensors. In the third stage, the signal is further compressed *within each sensor* (univariate approach) using Discrete Wavelet Transform (DWT). Implementation on real-time MFL signals demonstrates the algorithm's ability to achieve high compression ratios with low Normalized Mean Square Error (NMSE) while being fairly robust to baseline shifts.

1. Introduction

Carbon steel pipelines are widely deployed in many countries to transport oil and gas products across several thousands of kilometers. Owing to the large scale layout, even a small leakage in the pipelines results in large economic losses. More importantly, since stretches of these pipelines carrying inflammable products pass through highly populated areas, poor health of pipelines can give rise to safety concerns. Regular condition monitoring of pipelines is therefore necessary to ensure both public safety and proper transportation of products without loss of economy. MFL is a magnetic method of non-destructive testing that is widely used for both detection and characterization of metal loss defects using IPIG [1]. The IPIG consists of magnetic assembly, data storage and power modules. It travels by the pressure exerted by the flow of product that is being transported in the pipeline. The magnetic assembly consists of an array of permanent magnets and hall sensors.

The technique of MFL testing consists of (i) local magnetization of the pipewall to near saturation, and (ii) recording the leakage flux data in high end digital signal processors. It is a common practice to magnetize a pipeline axially. Consequently, defects oriented in a way that oppose magnetic flux (e.g. circumferential defects) are detected with greater ease than those oriented otherwise (for e.g. longitudinal defects). However if the width of the defect is sufficiently large, even axial magnetization can detect and characterize longitudinally disposed defects. A typical MFL signal in the absence and presence of metal loss is shown in Fig. 1. Ideally the recorded signal should stay constant in the absence of any anomalies. The presence of measurement noise, however, introduces fluctuations as shown in Fig. 1(a). Such segments of data do not carry any useful information and hence are termed as noisy blocks.

Interpretation of MFL signals and inversion techniques for defect characterization are discussed in numerous works [2–5]. The focus of this work is on the data acquisition stage of an IPIG operation, specifically the online compression of the large volumes of data that result during this stage. A typical 24 in IPIG tool generates 80 GB of data from a single run in the pipeline, which stretches up to 200 km. The success of the defect characterization (from data) naturally demands high quality (informative) data while operational constraints do not permit a large capacity storage device. A recent study recommends (i) increase in sampling frequency to achieve better characterization of defects, (ii) compactness of the storage components, and (iii) increase in inspection length of a single run [6]. The foregoing factors combined with the large volume of data that is generated calls for an efficient online data compression algorithm.

The problem of data compression has drawn the serious attention of academia and industry for several decades particularly because they arise in various important applications such as image processing, telecommunications, medicine, *etc.* [7]. Consequently, sophisticated algorithms for signal compression have emerged, each of them suiting a class of applications. The optimality of the compression algorithm is largely determined

* Corresponding author.
E-mail address: arunkt@iitm.ac.in (A.K. Tangirala).

0963-8695/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ndteint.2012.04.008

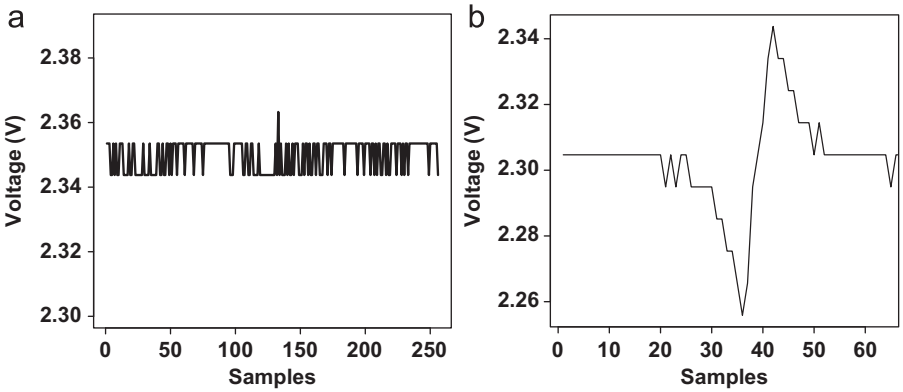

Fig. 1. Real-time MFL signals. (a) No anomaly: measurement noise, (b) Metal loss.

by the nature of the signal and the end-use of the information contained in the signal, both of which being highly dependent on the application. There is hardly a universal algorithm that provides best compression for all types of signals. The literature on the compression of MFL signals is relatively scarce. Widely used compression algorithms in other applications serve as suitable candidates for the application under study; yet, there is a need to provide a fresh treatment to the problem for two reasons. Firstly, the signals encountered from the IPIG carry features that differ considerably from those encountered in other arenas. Secondly, the end-use of data is towards detection of corroded areas of the pipelines placing our interest only in those parts of signal that capture some anomalies. Naturally there is hardly an incentive to store or retain signals that lack any information. Motivated by these reasons, a formal development of an efficient and a simple online compression algorithm is taken up in this work.

The main contribution of this paper is an efficient three-stage online algorithm for the compression of MFL signals. The first stage consists of a feature detection exercise that is used to screen out uninteresting portions of the signal. The algorithm is based on a statistical measure of variance (excitation), namely, the Mean Absolute Deviation (μAD). At this stage only those portions of the raw MFL data that contain some useful information about the pipeline anomalies are retained, while discarding other blocks of data. This step contributes significantly to the overall compression that is achieved with the proposed algorithm. The effectiveness of this stage, as it turns out, depends on the size of the block being screened relative to the axial length of the feature.

The second stage consists of compression across sensors where a multivariable dimensionality reduction technique is employed. For this purpose, the Principal Component Analysis (PCA), a well-established dimensionality reduction technique is applied. The assumption of linear relationships between sensor readings may not be valid, but the objective is not to extract the relationships. The aim is to deploy an effective multivariable compression tool that can be easily implemented online. Among the numerous techniques that suit the needs, PCA is a competitive choice because of its computational simplicity and theoretical efficacy [8]. The idea in PCA is to represent the same information (contained in the raw data) in a lower-order *virtual* sensor or principal component space. The virtual sensor spaces are orthogonal to each other. The benefits of compression achieved in this stage naturally depends on the number of sensors spanned by the anomaly (sensor span) and the correlation across those sensors. Therefore, this step is invoked only when the sensor span of the anomaly is large. Such a strategy avoids computational burden that is not worth the effort. Consequently, when an anomaly such as metal loss that usually has a small sensor span, is detected, the compression algorithm retains the raw MFL data, without transforming it to principal components.

In the third and final stage of compression algorithm, the strategy is to exploit the correlation within a single sensor's reading for compression. A natural choice of technique that efficiently achieves this task is the wavelet transform [9]. Wavelets have proved to be very effective compression tools in numerous applications such as image compression, biomedical signal compression and process data compression to name a few [10]. A Discrete Wavelet Transform (DWT) using Daubechies wavelets is employed for this purpose. Compressing the signal using DWT also denoises the signal, which improves the characterization of the pipeline anomalies.

The success of the overall compression is measured by an appropriate metric, namely, the compression ratio (see Section 2 for the mathematical definition). Implementation of the proposed method on the field data shows the effectiveness of this method. The actual compression ratio achieved in each dataset depends on the extent and type of anomalies present in the data. A dataset with fewer anomalies will yield a larger compression ratio.

The paper is organized as follows. Section 2 provides a brief review of the existing compression algorithms. The three-stage algorithm is presented in Section 3. Results obtained using the three-stage algorithm on field data are presented in Section 4. The paper ends with concluding remarks in Section 5.

2. Overview of compression techniques

Broadly speaking, one encounters two families of compression techniques based on their ability to exactly reconstruct the original data, namely, the *lossless* and *lossy* compression techniques. The performance of any compression algorithm can be evaluated using two indices—Compression Ratio (CR) and Normalized Mean Square Error (NMSE), defined as

$$\mathrm{CR} = \frac{\text{No. of original samples}}{\text{No. of retained samples}} = \frac{N}{R}$$
$$NMSE = \frac{\sum_{n=1}^{N}(x[n]-\overline{x}[n])^2}{\sum_{n=1}^{N}(x[n])^2} \quad (1)$$

where $x[n]$ is the original signal, $\overline{x}[n]$ is the estimated signal (*i.e.*, the decompressed signal), N is the length of $x[n]$ and R is the length of the compressed signal.

The basic difference between the lossless and the lossy compression techniques is the trade-off that they provide between the NMSE and the compression ratios that can be achieved. Lossless techniques insist on zero reconstruction error whereas lossy techniques are willing to sacrifice the zero reconstruction error property to achieve much higher compression ratios. Given this incentive, lossy

techniques are popularly deployed with the implicit understanding that the reconstruction error is within pre-defined tolerance levels. The lossy compression techniques are broadly classified into four categories as discussed below.

Direct Data Compression Techniques (DDCT) exploit correlation among successive samples to reduce redundancy. The Turning point (TP) algorithm replaces every three data samples with two that best represents the slope. The compression ratio achieved using TP is usually limited to two. The Amplitude Zone Time Epoch Coding (AZTEC) converts the signal into horizontal lines (plateaus) [11]. The reconstructed signal has discontinuities and distortion with large NMSE. The Coordinate Reduction Time Encoding (CORTES) is a combination of both TP and AZTEC. In CORTES the choice of either TP or AZTEC depends upon the nature of underlying signal.

Model-based Compression Techniques (MCT) rely on a model that represents the signal. Instead of the original data, the model parameters are stored. During the reconstruction stage, each data sample is predicted or interpolated using the model parameters. Vector Quantization (VQ) maps a set of vectors into predefined vector set in the codebook [12]. *Parameter extraction-based Compression Techniques* (PCT) detect and preserve only the required properties of the signal (see for example, extrema capture method by [13]).

In *Transform-based Compression Techniques* (TCT), the idea is to transform the signal into a domain which facilitates signal representation in terms of very few coefficients. For example, a sine wave typically requires large number of samples to represent it in time domain. However, it is most compactly represented in terms of merely three parameters—the amplitude phase and frequency. The Fourier transform of the signal achieves this representation since it uses sines as basis functions. Walsh–Hadamard Transform (WHT) is one of the simplest and fastest transform to be implemented. WHT is unitary and orthogonal transform composed by rectangular waveforms with values $+1$ and -1. The Discrete Fourier Transform (DFT) projects the signal onto a set of orthogonal sine and cosine basis functions [14]. Discrete Cosine Transform (DCT) is also similar to DFT except that the cosine wave is used as basis function. For narrowband signals, good compression can be achieved using DFT, DCT and WHT since the basis functions have similar properties. Several real-life applications generate signals that have time-varying frequency content. The aforementioned techniques are not ideally suited for compression of such signals. Of the several extensions that exist for handling time-varying frequency content, the Discrete Wavelet Transform (DWT) stands out as an excellent tool for compression. The DWT essentially represents the signal in terms of projections onto a set of non-redundant basis functions that have (near) compact spread in the time–frequency plane [15]. In fact, DWT is also suited for compressing broadband signals [16].

All of the foregoing data compression algorithms exploit the correlation among samples in a single channel. The across-sensor correlation can also be exploited calling for a deployment of multivariate data compression tools. Principal Component Analysis, which was introduced by Pearson [17] as a method for analyzing data in a lower-dimensional space emerges as a ubiquitous choice for multivariable data compression [18]. The driving engine for PCA is the Singular Value Decomposition (SVD) (of the data matrix) or the eigenvalue decomposition (EVD) of the sample covariance matrix. PCA has a striking resemblance with the Karhunen–Loeve transform (KLT) [19], which works on covariance matrices of jointly stationary multivariable random processes.

Evidently, there exists no ideal universal compression algorithm that is suited for all classes of signals. In fact, the choice of any compression technique depends on factors such as (i) nature of the underlying signal, (ii) important parameters to be retained while reconstructing the signal and (iii) the end-use of the data.

The compression algorithm presented in this paper is developed in light of the aforementioned factors. The remainder of the paper is devoted to the development and demonstration of the proposed three-stage compression algorithm.

3. Proposed online compression methodology

The three-stage algorithm that is developed for the online compression of MFL signals is presented in this section. The main contribution of this paper, *i.e.*, the feature detection algorithm is briefly discussed. The implementation of the three-stage algorithm is schematically shown as a flowchart in Fig. 2.

3.1. Stage I: MFL feature detection algorithm

The idea is to apply a screening algorithm that explores the variability in the MFL signal to differentiate between a noisy block and an informative block. The tool required for this purpose should be sensitive to variations and robust to baseline shifts. Statistics offers a variety of measures such as variance, Median Absolute Deviation (MAD) and Mean Absolute Deviation (μAD) as effective measures of variability. Among these three measures, MAD offers the maximum robustness to outliers and shifts. For a univariate series $\mathbf{x}=\{x[1],x[2],\ldots,x[N]\}^T$, median absolute deviation is defined as the median of the absolute deviation from the data's median,

$$MAD = median(\mathbf{x} - median(\mathbf{x})) \quad (2)$$

The robustness of MAD is attributed to the property of median, which can robustly estimate the average of a sequence in the presence of noise and outliers. For the application in hand, MAD is not a desirable candidate for screening since it is not sensitive to small amounts of deviation present in a signal. Consider the signal shown in Fig. 1(b), where more than 50% of the signal is constant. The value of MAD is zero, which indicates that there is no

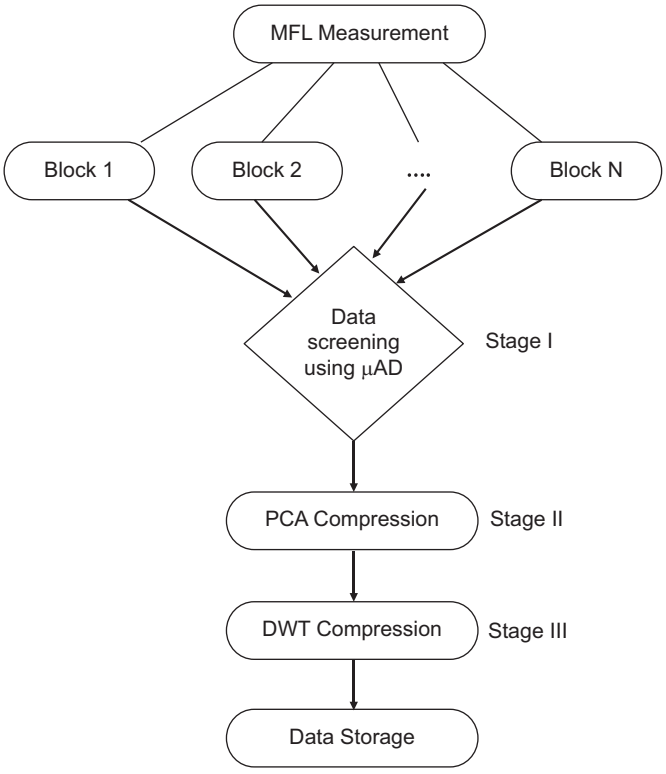

Fig. 2. Flow chart for the proposed data compression algorithm.

deviation in the signal. Thus MAD gives poor estimation of variability present in the signal, when more than half part of the signal is constant.

In contrast, the Mean Absolute Deviation, defined as the mean of absolute deviation from the median,

$$\mu\text{AD} = k*mean(\mathbf{x}-median(\mathbf{x})) \quad (3)$$

where k is a scaling factor and depends upon the probability distribution of the noise present in the signal, offers a good mix of robustness and sensitivity that is desired for the application. For Gaussian distribution, the scaling factor is estimated as 1.25 using Monte-Carlo simulations, which also approximately equal to the theoretical one [20]. The μAD calculated for the signal shown in Fig. 1(b) is 94.95×10^{-4}, whereas the MAD for the same signal is zero, supporting the choice of μAD as a suitable choice for screening the MFL signals.

Steps involved in the screening algorithm

1. Divide the signal into blocks and calculate μAD for each block.
2. If the μAD value is insignificant, reject that particular data block.
3. Else retain the data block along with the spatial location.

3.1.1. Parameters influencing the screening algorithm

The effectiveness of the screening algorithm is significantly influenced by two parameters, namely, the block size and the choice of threshold. The discussion below defines these parameters and presents guidelines for the parameter settings

3.2. Block size

Block size is defined as the number of samples per block. All the block sizes that are further discussed are in the power of two to enable easy computation of the DWT in the third stage. The efficiency of the screening algorithm to detect the presence of any anomaly is highly dependent on the Feature to Block ratio (FBR) as defined below,

$$\text{FBR} = \frac{\text{No. of samples spanning the feature}}{\text{Block size}} = \frac{FS}{BS}*100 \quad (4)$$

where the signal feature is defined as that part of the signal showing significant leakage. The feature length is not identical to the length of the anomaly/defect; in fact, it is usually longer than that of the defect. In some sense, therefore, it is reflective of the length of the anomaly. For a fixed feature length, FBR decreases as the block size (BS) is increased thereby increasing the possibility of the feature being undetected by the feature detection algorithm. The reason is the increased contribution of noise to the overall variability. Guidelines to choose the "right" block size can be provided by considering the extremes. A very small block size allows accurate detection of the smallest feature but has a reverse impact on the online signal processor. The processor should complete the screening cycle before the next block of data arrives. The complexity of operations involved in the proposed algorithm is briefly discussed in Section 4. An additional risk with a very small block size is that the noisy blocks may also be classified as informative blocks. A large block size provides a healthy margin between the times to screen and acquire a block of samples, but reduces the screening efficiency of the screening algorithm. Thus, there is a trade-off involved.

Two metal loss regions are taken up to demonstrate the effect of block size. These two metal loss regions are captured in the blocks of different sizes. The variation in the values of μAD for different FBRs are listed in Table 1. It can be observed that as the FBR decreases (with increase in block size) the value of μAD becomes insignificant even for informative blocks.

Table 1
Effect of block size on FBR & μAD.

No	Size-64		Size-128		Size-256	
	FBR	μAD(in 10^{-4})	FBR	μAD(in 10^{-4})	FBR	μAD(in 10^{-4})
1	17.18	97.31	8.59	57.22	4.29	32.4
2	73.43	135.54	36.71	79.23	18.35	47.25

Table 2
μAD for different pipeline features.

No	Pipeline feature	μAD(in 10^{-4})
1	Noise	85.36
2	Metal loss	94.95
3	Weld	543.53
4	Flange	1370.6

From a detailed examination of the field data, it was observed that the smallest (in length) feature spanned 25 samples. Taking into account the aforementioned factors and the observations in Table 1, a block size of 64 samples, that is, about more than twice the smallest feature size is chosen.

3.3. Threshold

The role of threshold, as with several other algorithms, is to practically differentiate between a noisy block and an informative block. Theoretical determination of the threshold is unimaginably difficult and impractical. The natural alternative is to use an empirical approach using field data. The choice of threshold depends upon the μAD value of the smallest pipeline feature and the data block which contains maximum level of noise. The value of μAD for several data blocks containing different type of pipeline anomalies is listed in Table 2. The μAD value for the smallest pipeline feature is found to be 94.95×10^{-4}, whereas the same is found to be 85.36×10^{-4} for a feature-free data block exhibiting maximum noise variance. The average of these two these two values (90×10^{-4}) is taken as the threshold for detecting the pipeline feature using μAD. A higher threshold value can increase the compression ratio but also increases the probability of a defective region going undetected. Similarly, a smaller threshold will increase the probability of retaining data containing anomalies but will decrease the compression ratio.

3.3.1. Robustness

The baseline of the MFL signal may vary due to changes in pipeline thickness, permeability, etc. To test the robustness of μAD towards changing baseline, two MFL signals are analyzed. The first MFL signal, shown in Fig. 3(a), contains a simple baseline shift. The calculated μAD values are below the threshold as shown in Fig. 3(b). In the second MFL signal, small metal loss defects are present in the vicinity of baseline change as shown in Fig. 3(c). The calculated μAD values are above the threshold as shown in Fig. 3(d). Thus, the proposed algorithm is able to judge the blocks containing features as informative despite the presence of baseline shifts. Furthermore, it rightly classifies the blocks with pure baseline variations as non-informative.

3.4. Stage II: Multivariate compression using PCA

The multivariate compression is invoked only when a particular anomaly is detected by a large number of sensors. For instance, a weld in the pipeline spans across all the sensors.

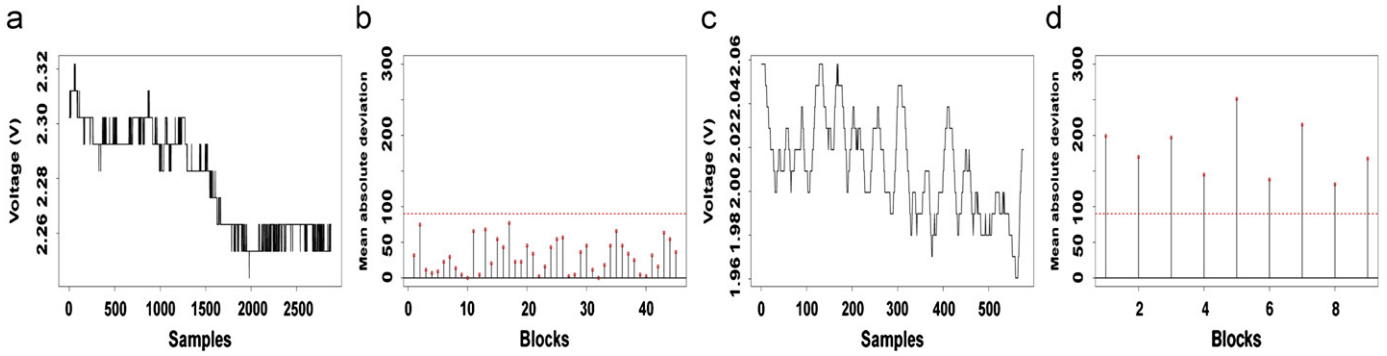

Fig. 3. Test for robustness. (a) Raw MFL signal, (b) Mean absolute deviation across blocks, (c) Raw MFL signal, (d) Mean absolute deviation across blocks.

Thus two or three blocks, depending upon the axial length of the weld, will be detected across all the sensors using the feature detection algorithm.

Principal Component Analysis is a linear orthogonal transform of measurements from a p-dimensional space to another p-dimensional space, so that the coordinates of the data in the new space are uncorrelated and greatest amount of variance of the original data is expressed by only few coordinates. Let p variables $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_p$ be transformed to another p variables $PC_1,PC_2,\ldots,PC_p$ called principal components. The principal components are arranged in the order of decreasing variance.

The variables $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_p$ are standardized to have zero mean and unity variance primarily to avoid ill-conditioning. The principal components can be calculated through an eigenvalue decomposition of the sample covariance matrix ($\mathbf{X}^T\mathbf{X}$) where

$$\mathbf{X}=[\mathbf{x}_1\ \ \mathbf{x}_2\ \ \ldots\ \ \mathbf{x}_p]$$

The ith principal component is calculated as a linear combination of the variables

$$PC_i = e_{i1}\mathbf{x}_1+e_{i2}\mathbf{x}_2+\cdots+e_{ip}\mathbf{x}_p,\quad i=1,2,3,\ldots,p \tag{5}$$

where the constants $e_{i1},e_{i2},\ldots,e_{ip}$ are the elements of the corresponding eigenvector (also called as loadings) of the covariance matrix [21].

The theoretical number of principal components is equal to number of sensors that have captured the anomaly. The useful information is however contained in a much fewer principal components as indicated by the significant eigenvalues of the covariance matrix. Retaining only the useful ones produces significant compression [22].

It was observed that first five principal components account for 98% of the variance. Thus, a compression ratio of 8.77 is achieved by storing only the first five principal components along with the corresponding loading matrix. In an offline mode, the signal can be reconstructed by taking the product of principal components with the transpose of the loading matrix. If the number of sensors capturing the anomaly such as a metal loss is very small, this stage of compression is skipped merely because the compression achieved is insignificant.

3.5. Stage III: univariate compression using DWT

In the third and final stage of compression, principal components obtained from the second stage (if stage two was activated) or the screened raw MFL data obtained from the first stage (if the second stage was skipped) are transformed to wavelet domain to achieve further compression. The wavelet transform consists of projecting the signal onto a set of wave-like basis functions [15],

$$\psi_{s,\tau}(t)=\frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \tag{6}$$

generated from a small wave-like function called the "mother wavelet", $\psi(t)$, satisfying

$$\int_{-\infty}^{\infty}\psi(t)\,dt=0$$

The quantities s and τ denote the scale and translation respectively. The transform of a signal with $\psi_{s,\tau}(t)$ captures the details present in a signal at that scale (resolution) and location. Naturally these details are accompanied by a corresponding approximation, obtained by the transform of the signal with a scaling function $\phi(t)$ counterpart of $\psi(t)$.

The distinguishing feature of wavelet transform is that it provides a multiresolution (multiscale) representation (MR) of the signal similar to the representation of a geographical map at different scales (resolutions). A compact representation, desirable in compression applications, is achieved by requiring the basis functions to be orthonormal, which is achieved by choosing $s=2^j$ and $\tau=n2^j$, $j\in Z$. It can be shown that this MR is identical to the repeated filtering of signal through a set of (closed-to-ideal) low- and high-pass filters combined with downsampling (by a factor of two) at every stage of filtering [15]. The resulting coefficients are the approximation and detail coefficients respectively. Due to the nature of the wavelet basis, several signals attain sparse representations (small number of non-zero coefficients) in the wavelet domain.

For signals corrupted with noise, sparse representation is only achieved after thresholding of the wavelet coefficients, also termed as wavelet de-noising. Wavelet de-noising involves forcing the insignificant wavelet coefficients to zero with and without shrinking the significant ones [23,24]. A good deliberation on the different methods and important influencing parameters in wavelet de-noising is found in Cai and Harrington [25]. The sparseness inherently leads to data compression [26].

A three level wavelet decomposition of every data block received from the previous stage is implemented. The universal thresholding is used to identify the significant wavelet coefficients [24]

$$t=\sigma\sqrt{2\ln(N)};\sigma\approx\frac{median\|d_i\|}{0.6745} \tag{7}$$

where N is the length of data array and σ is the standard deviation of the noise. For real-time data σ is unknown, but can be estimated using the robust MAD estimator where $\{d_i\}$ is the set of first level detail coefficients. Thresholding of coefficients can be carried out in two different ways - hard and soft thresholding.

Hard thresholding simply shrinks all the coefficients below the threshold to zero without affecting the significant ones. On the other hand, soft thresholding additionally shrinks the significant coefficients, thereby reducing the amplitude of signal. The amplitude of MFL signals plays an important role in determining the percentage wall loss estimation in regions containing anomalies [27]. Hence soft thresholding can lead to poor characterization of pipeline anomalies. Thus, hard thresholding is better suited for the purpose.

The significant wavelet coefficients along with their index values (time stamps) are retained at the end of this stage. If stage two was activated, the loading matrix (weighings to reconstruct the principal components) is also retained.

4. Results and discussion

The data presented in this paper has been collected by an IPIG from an actual buried oil pipeline. The instrument contains strong permanent magnets that magnetize the ferromagnetic pipe-wall axially to near saturation. An array of circumferentially disposed Hall-effect sensors, located at the magnetic neutral plane of the permanent magnet assembly senses the radial and/or axial component of leakage flux. The measured Hall voltage is proportional to the leakage flux density. The paper works with the radial component of MFL signal, calibrated in terms of measured voltage. Four real-time MFL datasets are selected for evaluating the performance of the proposed three-stage compression algorithm. Each dataset contains 5000×64 samples (*i.e.*, 5000 samples from each 64 sensor). A grayscale image of the first dataset from a region near a weld is shown in Fig. 4(a). The second dataset carries information about a weld and few metal losses as displayed in Fig. 5(a). The third dataset consisting of the MFL signal readings near a single region encompassing welds, sleeve and valves along with few metal losses is shown in Fig. 6(a). Finally, the fourth dataset contains the MFL signals from the region where there are large number of small defects due to corrosion patch in the pipeline as shown in Fig. 7(a). The metal loss defects that are discussed in this paper are circumferential defects as discussed in Section 1. However the same technique can be extended to longitudinal defects as well.

A grayscale image of the first dataset post screening (Stage 1) is shown in Fig. 4(b). The gray blocks represent the presence of features carrying useful information about that section of the pipe. The blocks whose μAD values fall below the threshold are indicated by white regions; in real-time screening these data blocks are rejected. It can be observed that the screening algorithm has retained few additional blocks which are known not to contain any useful information. This is due to the fact that these spurious signals have similar variability as that of MFL signals with some features. The detection algorithm is only based on variability (based on μAD) present in the signal. Thus the proposed MFL feature detection algorithm, per se, will not able to discriminate spurious signals of similar variability. Hence these signals are considered as informative blocks and processed further. In offline feature characterization, the spurious signals are discarded using their signatures. The compression achieved can be computed by noting that only 9152 samples (143 data blocks $\times$ 64 samples per block) are required to be stored as against 320 000 samples (5000×64) of the raw record. The calculation yields a compression ratio of 34 in the first stage without any loss of useful information. Similarly, Figs. 5(b), 6(b) and 7(b) show the performance of screening algorithm in detecting the features and compressing the data in the other three datasets. These results clearly indicate the effectiveness of the proposed data

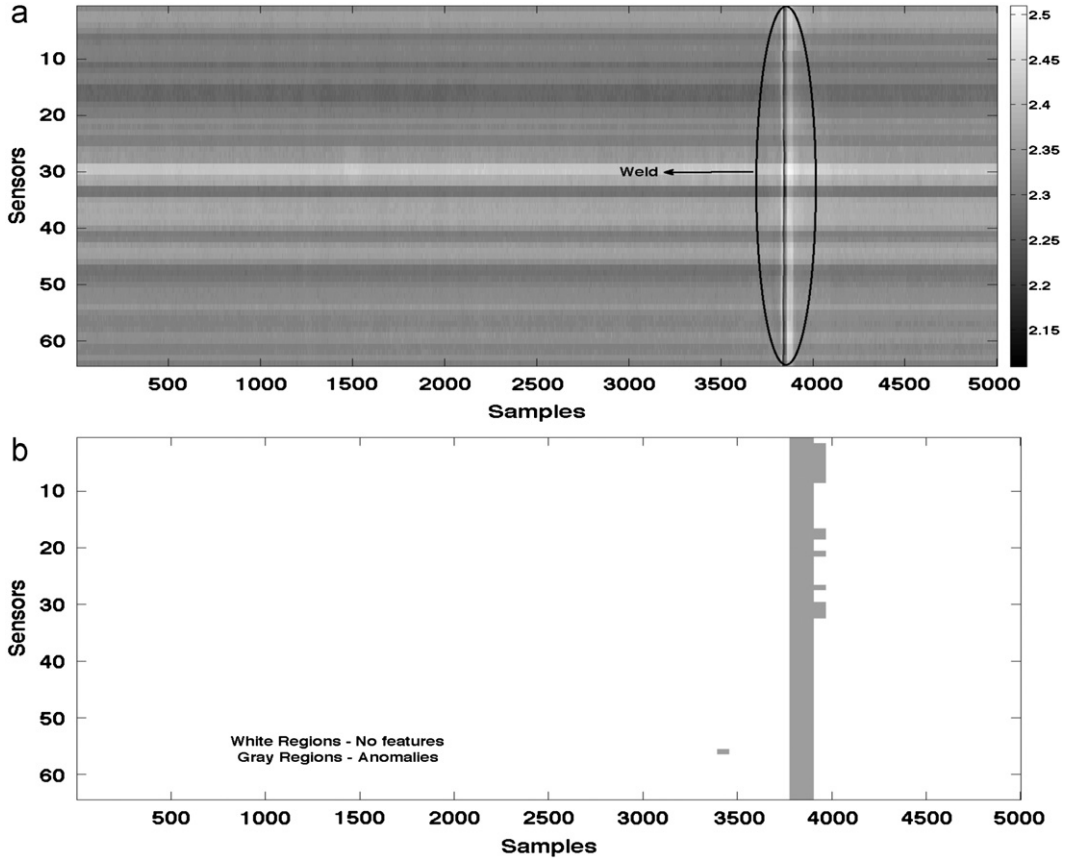

Fig. 4. Data screening results for first dataset. (a) Raw MFL data, (b) Dataset after screening algorithm.

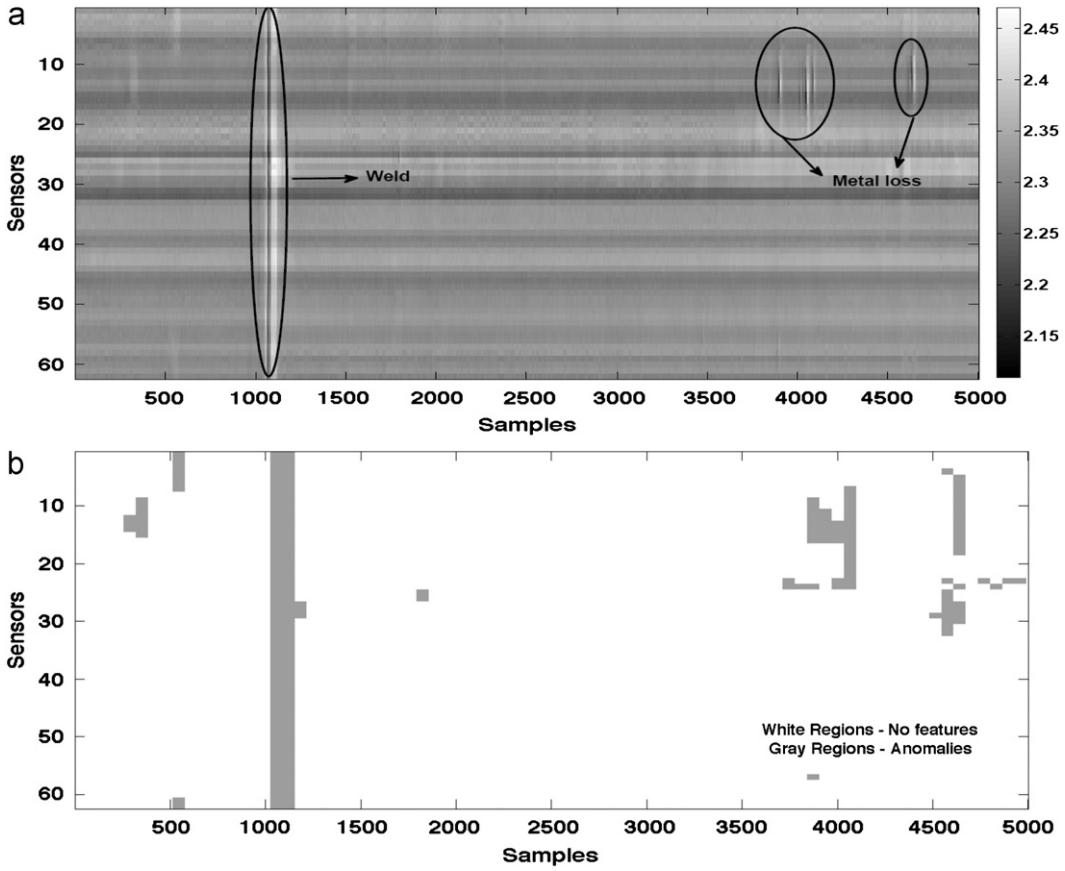

Fig. 5. Data screening results for second dataset. (a) Raw MFL data, (b) Dataset after screening algorithm.

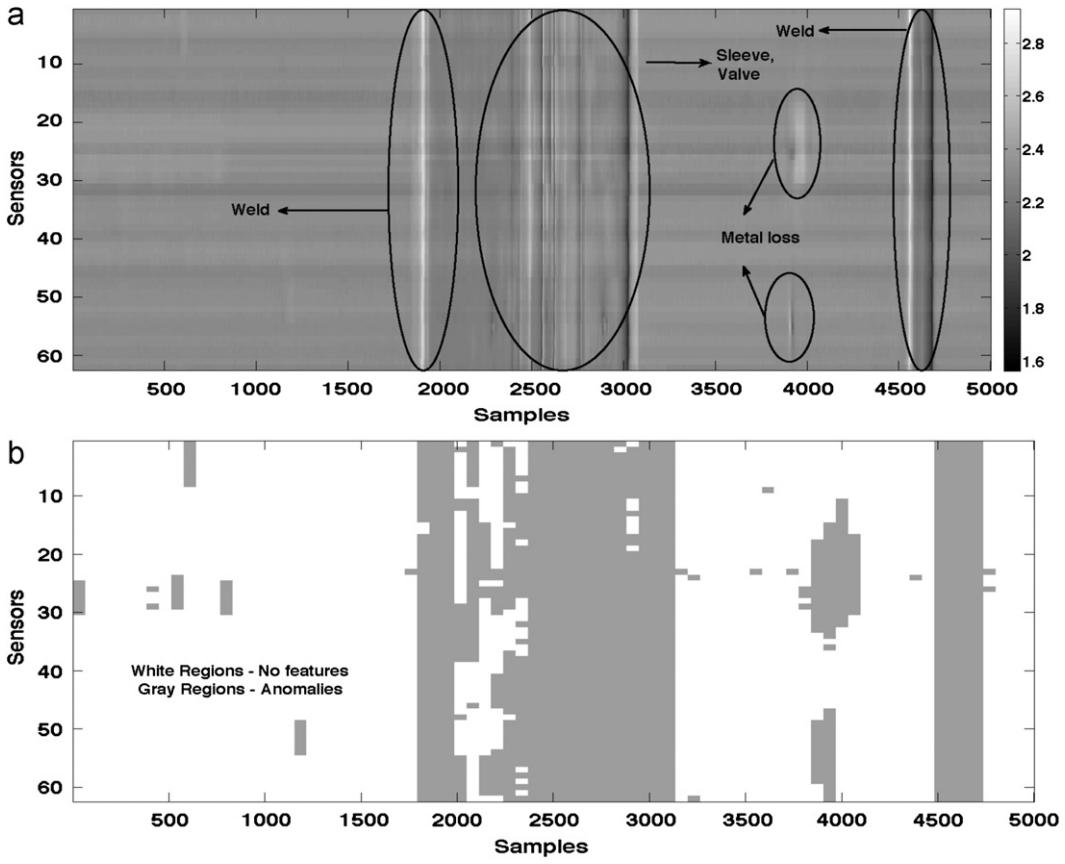

Fig. 6. Data screening results for third dataset. (a) Raw MFL data, (b) Dataset after screening algorithm.

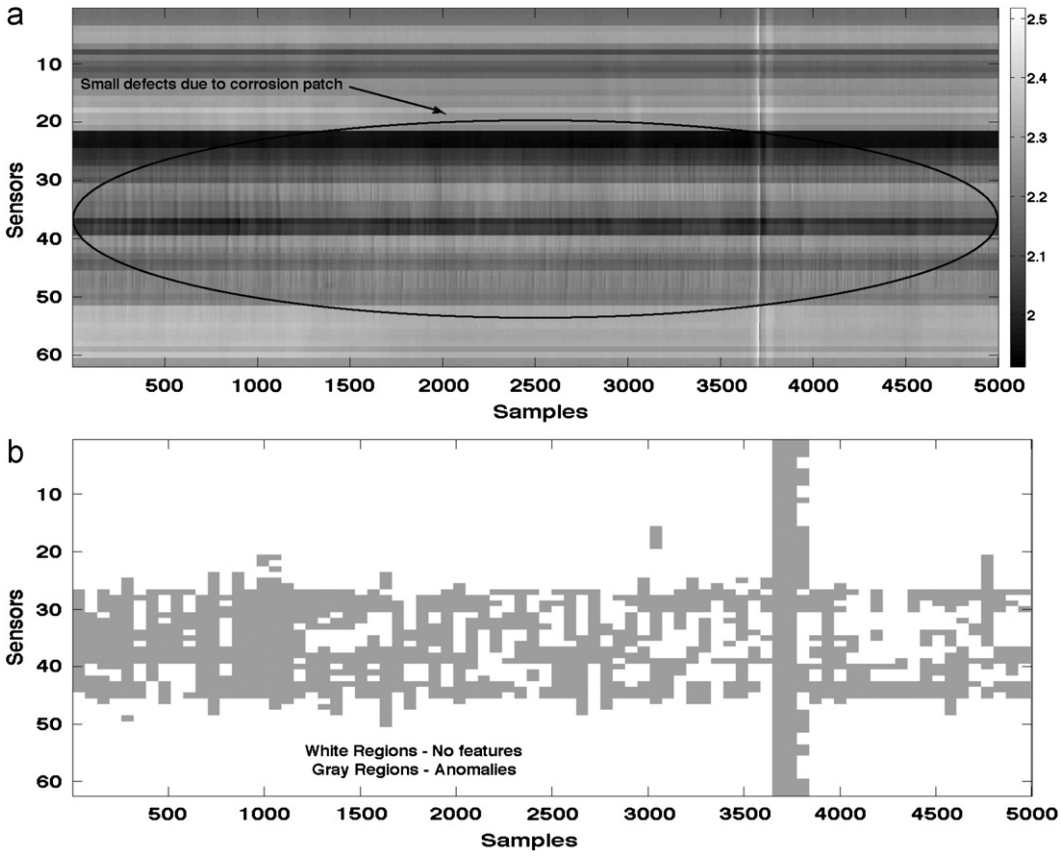

Fig. 7. Data screening results for fourth dataset. (a) Raw MFL data, (b) Dataset after screening algorithm.

screening to detect the regions of anomalies. The compression ratios achieved in stage one for the three datasets are tabulated in Table 3.

Moving on to the second stage, the compression is invoked depending on the circumferential span of the feature. In the weld (first) dataset, the sensor span near the weld region is maximum since the weld spans across entire periphery of the pipeline. The axial length of this weld is roughly equal to the length of two blocks. Thus totally 128 blocks (*i.e.*, 2 blocks per sensor $\times 64$ sensors) are considered for stage two compression. Subjecting this data to PCA reveals that merely five principal components are sufficient to explain 98% variance in the signal. The remaining 15 blocks (that have little circumferential span) are retained without any transformation.

The second stage compression yields a CR of 2.06 taking into account the fact that five PCs, the loading matrix for these five PCs and the untransformed 15 blocks from stage 1 are stored as against the 9152 samples at stage 1. The total compression ratio for the weld dataset at the end of stage 2 is 70.04 (34×2.06). The compression ratios achieved from the first two stages for the three datasets are listed in Table 4.

The third stage of compression algorithm is implemented either on the principal components or on the un-transformed raw data (retained from stage 1), depending upon the type of feature being detected. A three level wavelet decomposition using a Daubechies filter (D4) is carried out on the data followed by a hard thresholding of the wavelet coefficients. At the end of the three-stage compression algorithm, only the significant wavelet coefficients and the loading matrix of the principal components are stored. A compression ratio of 2.752 is achieved for the first dataset after employing wavelet compression. The overall compression ratio for all the datasets using the three-stage algorithm is listed in Table 5. The compression ratio that can be achieved decreases as the number of anomalies present in the dataset increases, which increase the number of blocks have to be retained at the screening stage. The comparison of original and reconstructed MFL signal of two blocks from a single sensor is shown in Fig. 8. The loss due to compression (reconstruction error) is negligible. In particular, the peaks have been preserved quite accurately.

Table 3
Stage I compression results (μAD).

No	Nature of dataset	CR
1	Weld	34
2	Metal loss & weld	18.4
3	Flange, valve & Sleeve	3.07
4	Large metal losses	3.57

Table 4
Stage II compression results (μAD and PCA).

No	Nature of dataset	Compression results		
		Stage I	Stage II	Stage I & II
1	Weld	34	2.06	70.04
2	Metal losses and weld	19.23	1.76	33.84
3	Flange, valve & sleeve	3.07	2.07	6.35
4	Large metal losses	3.57	1.56	5.56

Implementation aspects

An important aspect of online implementation is the computational complexity of the algorithm, *i.e.*, the number of additions and multiplications involved. The FLoating point OPeration (FLOP)

Table 5
Stage III results (μAD, PCA & DWT).

Nature of dataset	Compression results			
	Stage I	Stage II	Stage III	Final
Weld	34	2.06	2.752	192.75
Metal losses and weld	19.23	1.76	2.51	84.95
Flange, valve & sleeve	3.07	2.07	2.81	17.85
Large metal losses	3.57	1.56	2.61	14.53

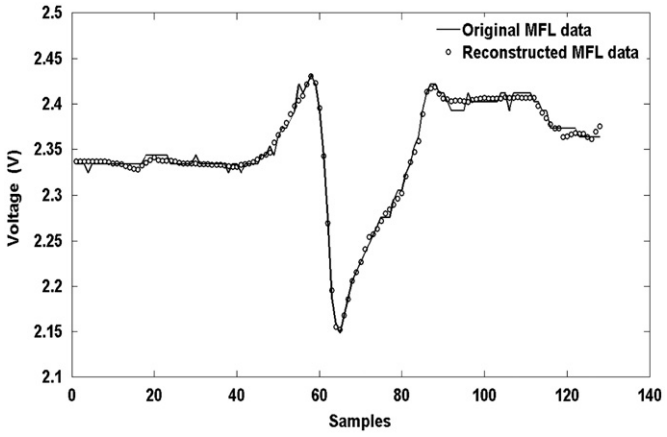

Fig. 8. Comparison of original vs. reconstructed MFL signal.

Table 6
Computational complexity (no. of flops) at each stage.

Method	μAD (Stage I)	PCA (Stage II)	3 Level DWT (Stage III)
Theoretical	$3N*N_s$	$N*p^2$	$7N*N_p$
Test data	12 288	262 144	2240

is a commonly used measure to compute the complexity order of any algorithm. The approximate number of FLOPS required for each stage calculated for the worst-case scenario, *i.e.*, when a feature is detected in a particular axial location across all sensors is reported in Table 6. For the algorithm in use and the field data, the block size is 64 for a 64-array data. In Table 6, N refers to block size, N_s is equal to the number of sensors, p indicates the number of principal components and N_p refers to number of principal components retained.

The three stages collectively involve approximately 276 672 FLOPs for the worst case scenario. The DSPs of today operate at a speed of 100 to 120 MegaFLOPs per second. Computationally the algorithm only imposes 0.2% of the maximum load that the processor can handle. In terms of the throughput, approximately 2.5 ms are required for the calculations on one block of all-sensor data, which is negligible compared with 64 ms, the time available until the next block of data is acquired, assuming a sampling rate of 1 kHz. Thus, the proposed algorithm is practically feasible and fast. In order to reduce the time consumed by the process, the algorithm is also being implemented on a parallel DSP hardware.

5. Conclusions

A three-stage univariate–multivariate algorithm for the compression of MFL signals that can be implemented in real-time has been developed. The development stems from identifying the mathematical tools that rightly meet the requirements of the problem. Applications to field data clearly demonstrate the effectiveness of the algorithm. High compression ratios with minimal loss of information are achieved. It is best to run the screening algorithm at the first stage for maximum compression. The screening algorithm based on μAD significantly contributes to the overall compression ratio. A noteworthy point is that the screening algorithm serves as a preliminary picture of the pipeline condition thereby also presenting a first-level discrimination of the anomalies, *i.e.*, between a metal loss and one of weld, flange or sleeve. Naturally, the screening algorithm can be tailored to incorporate a higher-order statistical measure that can provide better discrimination. Similarly a non-linear multivariate compression algorithm can be used instead of PCA. However, in making any such refinements, the primary goal of compression should not be forgotten since sophisticated algorithms carry a price tag of increased computational complexity that can render them unsuitable for online implementation. Preliminary tests on real-time implementation of the proposed algorithm have been encouraging to foresee its implementation in the next generation IPIGs.

References

[1] Srivastava GP. Developments in instrumentation and automation for NDE applications: in-house experience in the Department of Atomic Energy. Insight 2003(1):73–86.
[2] Joshi A, Udpa L, Udpa S, Tamburrino A. Adaptive wavelets for characterizing magnetic flux leakage signals from pipeline inspection. IEEE Trans Magnet 2006;42(10):3168–70.
[3] Mandache C, Clapham L. A model for magnetic flux leakage signal predictions. J Phys D: Appl Phys 2003;36:2427–31.
[4] Altschuler E, Pignotti A. Nonlinear model of flaw detection in steel pipes by magnetic flux leakage. NDT & E Int 1995;28(1):35–40.
[5] Saha S, Mukhopadhyay S. Empirical structure for characterizing metal loss defects from radial magnetic flux leakage signal. NDT & E Int 2010(43):507–12.
[6] Bahuguna SK, Mukhopadhyay S, Bhattacharya S, Patil MB, Das S, Biswas BB. Development of a DSP based Data Acquisition System for IPIG Project. BARC Newsletter 2006;264:26–9.
[7] Ziv J, Lempel A. A universal algorithm for sequential data compression. IEEE Transact Inf Theory 1977;IT-23(3):337–43.
[8] Tipping ME, Bishop MC. Probabilistic principal component analysis. J R Stat Soc B 1999;61(3):611–22.
[9] Tai SC, Sun CC, Yan WC. A 2D ECG compression method based on wavelet transform and modified SPIHT. IEEE Trans Biomed Engg 2005;52(6):999–1008.
[10] Lu Z, Kim DY, Pearlman WA. Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm. IEEE Trans Biomed Engg 2000;47(7):849–56.
[11] Cox JR, Nulle FM, Fozzard HA, Oliver GC. AZTEC—a preprocessing program for real-time ECG rhythm analysis. IEEE Trans Biomed Engg 1968;15:128–9.
[12] Cohen A, Poluta M, Scott-Millar R. Compression of ECG signals using vector quantization. Proc IEEE 1990;90:49–54.
[13] Fink E, Gandhi HS. Compression of time series by extracting major extrema. Exp Theoret Artif Int 2011;23:255–70.
[14] Al-Nashash HAM. ECG data compression using adaptive Fourier coefficients estimation. Med Eng Phys 1994;16:62–6.
[15] Mallat S. A wavelet tour of signal processing. Academic Press; 2009.
[16] Cardoso G, Saniie J. Performance evaluation of DWT, DCT, and WHT for compression of ultrasonic signal. In: IEEE conference, vol. 3;2004. p. 2324–17.
[17] Pearson K. On lines and planes of closest fit to systems of points in space. Phil Mag, Ser B 1901;2:559–72.
[18] Jackson JE. Principal components and factor analysis: I. Principal components. J Qual Technol 1979;21(4):341–9.
[19] Gerbrands JJ. On the relations between SVD, KLT and PCA. Pattern Recogn 1981;14:375–81.
[20] Pham-Gia T. The mean and median absolute deviations. Math Comput Model 2001;34:921–36.
[21] Jolliffe II. Principal component analysis. Springer; 2002.
[22] Chau FT, Liang YZ, Gao J, Shao XG. Chemometrics: from basics to wavelet transform; vol. 164. John Wiley & Sons, Inc.;2004.
[23] Donoho DL, Johnstone IM, Kerkyacharian G, Picard D. Wavelet shrinkage: asymptopia? J R Stat Soc B 1995;57:301–9.
[24] Donoho DL. De-noising by soft-thresholding. IEEE Transact Inf Theory 1995;41(3):613–27.
[25] Cai C, Harrington PB. Different discrete wavelet transforms applied to denoising analytical data. J Chem Inf Comput Sci 1998;38:1161–70.
[26] Santoso S, Powers EJ, Grady WM. Power quality disturbance data compression using wavelet transform methods. IEEE Trans Power Delivery 1997;12(3):1250–7.
[27] Zyoying H, Peiwen Q, Liang C. 3D FEM analysis in magnetic flux leakage method. NDT & E Int 2006;39:61–6.