CORRECTED INTERVAL MULTISCALE ANALYSIS (CIMSA) FOR THE DECOMPOSITION AND RECONSTRUCTION OF INTERVAL DATA

A Thesis

by

SHAMEEL ABDULLA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Mohamed Nounou
Committee Members,	Hazem Nounou
	Ahmed Abel-Wahab
Head of Department,	Arul Jayaraman

May 2021

Major Subject: Chemical Engineering

ABSTRACT

Multi-Scale Analysis (MSA) is a powerful tool used in process systems engineering and has been utilized in many applications such as fault detection and filtering. In this paper, the extension of MSA for interval data is studied. Unlike single-valued data, interval data use bounds to denote the uncertainties within data points. Data aggregation can be used to convert a set of single-valued data into a smaller set of interval data. The literature on MSA of interval data is sparse and its use in process engineering has not been documented. Therefore, in this paper, three methods of handling interval data are studied: an interval arithmetic (IA) method, a center and radii (CR) method, and an upper and lower (UL) bound method. The main drawback identified when working with intervals is interval inflation/over-estimation. In this paper, interval inflation caused when applying MSA on interval data is described in detail. New algorithms to correct for the over-estimations have been proposed. The overestimations in interval data were corrected, and all three methods performed equally well in decomposing and reconstructing the signals. The Interval MSA algorithms developed were utilized to filter noisy interval data. The CIMSA-CR (the center and radii method) performed the best amongst the three methods for the filtering application. The optimum depth of decomposition, the shape of features in the input signal were also studied to understand how it affects the filtering performance.

DEDICATION

I dedicate this thesis to my parents, my wife and my two daughters, for their inspiration and unconditional support. I would also like to dedicate this work to all my teachers and professors who mentored and guided me to what I am today.

ACKNOWLEDGEMENTS

I would like to thank my committee chair and co-chair, Dr. Mohamed Nounou, and Dr. Hazem Nounou and my committee members, Dr. Ahmed Abdel-Wahab, for their guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff at Texas A&M Univeristy at Qatar for facilitating a wonderful undergraduate and graduate experience.

Finally, thanks to my mother and father for their encouragement and to my wife for her patience and love.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of: the chair, Dr. Mohamed Nounou, from the Department of Chemical Engineering, and the committee members, Dr. Hazem Nounou, from the Department of Electrical Engineering and Dr. Abdel Wahab, from the Department of Chemical Engineering.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

The graduate study was self-funded with some assistance from the Employee Tuition Assistance program at TAMU.

The contents of this work are solely the responsibility of the authors and do not necessarily represent the official views of the TAMU.

NOMENCLATURE

MSA	Multi-Scale Analysis
IA	Interval Arithmetic
CR	Center and Radii approach
UL	Upper and Lower bounds approach
IMSA	Interval Multi-Scale Analysis
CIMSA	Corrected Interval Multi-Scale Analysis
DWT	Discrete Wavelet Transform

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

1.1. Overview

Data are generated from various systems at unprecedented levels. Advancements in sensor technologies, communication, and computational power have led to an influx in the amount of data that is generated. This trend is seen across various fields such as industrial, medical, financial, and educational. Therefore, there is a need to aggregate data to make them more manageable and make their utilization in various computational methods faster. Converting large datasets into intervals is a way to summarize or aggregate large data sets, and thus a smaller data set is used for analysis.

Multi-Scale Analysis (MSA) is a tool used to analyze and filter signals in the temporal and frequency domains. MSA has been effectively utilized in many applications, such as fault diagnosis of machines based on vibration data [1], image segmentation and extraction of details from images [2] [3], Fuzzy Kalman Filtering [4], and fault detection applications [5] [6]. MSA of single-valued data is well documented and has been studied extensively. However, there have been no comprehensive studies on the MSA of interval data to the best of our knowledge. Thus, the primary objective of this work is to investigate the issues that would arise when applying MSA on interval data and propose a novel algorithm that is capable of decomposing and reconstructing interval data.

Interval Arithmetic was developed around the 1960s to deal with computations using interval data [7]. However, the methods are conservative [8]. This was done to ensure that the solutions obtained always include all possible combinations from the input data. Although useful in some applications such as bracketing a solution to a problem with uncertain model parameters, or keeping track of round-off errors in a calculation, the conservative nature becomes detrimental

in applications such as filtering and state estimation. This drawback has not prevented people from exploring the possibilities of integrating interval data into classical methods such as the Kalman Filter. Techniques such as the Interval Kalman Filter (IKF) were developed [9] [10] [11] [12]. But most of these researches have adopted methods such as converting the interval to single-valued data when challenges arise, or to have algorithms that reset the interval bounds when they cross a certain threshold after multiple iterations to tackle issues with intervals [13] [14] [15]. The same conservative issues show up when trying to apply MSA on interval data using interval arithmetic. This work with describe these issues in detail and will propose ways to mitigate them.

The remainder of this paper is organized as follows. Section 2 provides a critical review of existing methods to deal with interval data. Section 3 introduces the multiscale decomposition and reconstruction of single-valued data. Section 5 discusses the challenges that arise when applying MSA on interval data. In Section 6, the proposed corrected interval multiscale analysis algorithms are described. In Section 7, illustrative examples that show how the proposed algorithm performs in decomposition, reconstruction, and filtering applications. Concluding remarks and future directions for work are presented in Section 9.

1.2. Research Contributions

The problem that arises with using intervals is inflation/over-estimation. This leads to diverging or large bounds that do not necessarily represent the data well. The main contribution of this work is the development of a corrected MSA algorithm that can mitigate the interval inflations. The research objectives are as follows:

- Study the effects of applying discrete MS decomposition and reconstruction (Naïve approach) on Interval Data using:
    - Interval Arithmetic (IA)
    - Center and Radii approach (CR)

    - Upper and Lower bounds approach (UL)
- Develop corrected algorithms for all three methods.
- Compare the results found with the Naïve approach and the corrected algorithms.
- Combine the developed corrected algorithms with an MS filtering algorithm and investigate its performance.
    - Study the effect of the depth of decomposition on the filtering performance.
    - Study the effect of the choice of interval size on the filtering performance.
    - Study the effect of using input signals with various input features on the filtering performance.

2. EXISTING METHODS OF DEALING WITH INTERVAL DATA

Interval data are represented using bounds, which may represent a set of aggregated measurements. The bounds can be computed in various ways. Common practices include the use of the largest and smallest values within the aggregated data as the upper and lower bounds. This method makes the bounds all-inclusive and neglects the distribution of the data. However, physical measurements are typically associated with some form of uncertainty, which is calculated from aggregated data by estimating their probability distribution functions. Therefore, a more useful approach would be to compute confidence intervals that give a better idea of where the center of the data is (mean, μ), and how the data are spread out (variance, σ) as shown in Figure 1. Various computational methods can be used to deal with interval data, which are summarized next.

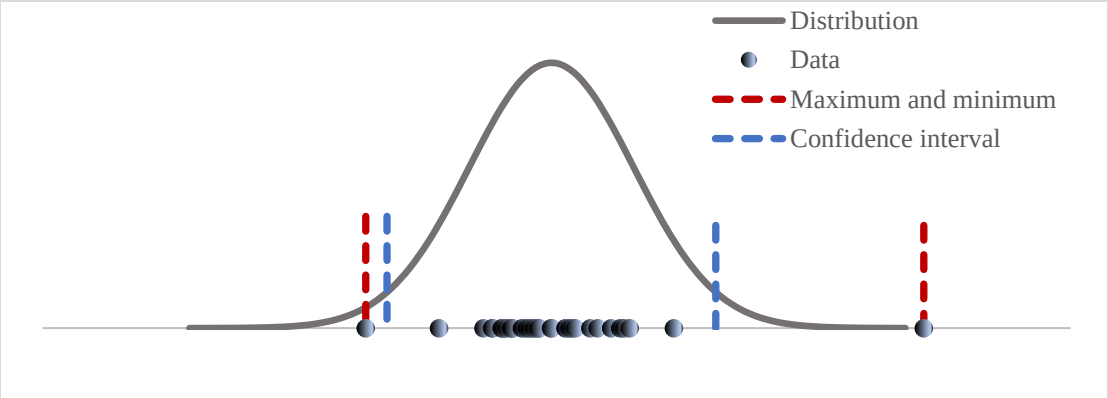


Figure 1. A sample data set is converted to interval data.

2.1. Interval Arithmetic (IA) Method

Interval Arithmetic is a recent branch of mathematics that deals with computations using intervals. Addition, subtraction, multiplication, and division of interval data are defined as shown

in Table 1. The final output is calculated such that it contains all the possible combinations of the operation on the interval data.

Table 1. The elementary arithmetic operations using intervals (underline and overline denote lower and upper bounds, respectively).

$[\underline{x},\bar{x}]+[\underline{y},\bar{y}]$	=	$[\,\underline{x}+\underline{y},\bar{x}+\bar{y}\,]$
$[\underline{x},\bar{x}]-[\underline{y},\bar{y}]$	=	$[\,\underline{x}-\bar{y},\bar{x}-\underline{y}\,]$
$[\underline{x},\bar{x}]*[\underline{y},\bar{y}]$	=	$[\min(\,\underline{x}\underline{y},\underline{x}\bar{y},\bar{x}\underline{y},\bar{x}\bar{y}\,),\max(\,\underline{x}\underline{y},\underline{x}\bar{y},\bar{x}\underline{y},\bar{x}\bar{y}\,)\,]$
$\frac{[\underline{x},\bar{x}]}{[\underline{y},\bar{y}]}$	=	$\left[\min\left(\frac{\underline{x}}{\underline{y}},\frac{\underline{x}}{\bar{y}},\frac{\bar{x}}{\underline{y}},\frac{\bar{x}}{\bar{y}}\right),\max\left(\frac{\underline{x}}{\underline{y}},\frac{\underline{x}}{\bar{y}},\frac{\bar{x}}{\underline{y}},\frac{\bar{x}}{\bar{y}}\right)\right] \Leftrightarrow 0\notin[\underline{y},\bar{y}]$

2.2. Centers and Radii (CR) Method

Intervals can also be represented using their centers and radii. Given an interval $[\underline{x},\bar{x}]$, it can be represented as $c(x)+r(x)[-1,1]$, where $c(x)$ is the center of the interval computed as $c(x)=\frac{\underline{x}+\bar{x}}{2}$, and $r(x)$ is the radius of the interval computed as $r(x)=\frac{\bar{x}-\underline{x}}{2}$. Thus, the interval data can be represented using two single-valued data points: $c(x)$ and $r(x)$. Computations using centers and radii avoid the pitfalls of interval arithmetic and its conservative nature. This procedure has been followed by other researches [6] in dealing with intervals.

By representing interval data as centers and radii, the computation can be done on the center and radii as single-valued data. The result can then be combined to retrieve the interval form as shown in Figure 2.

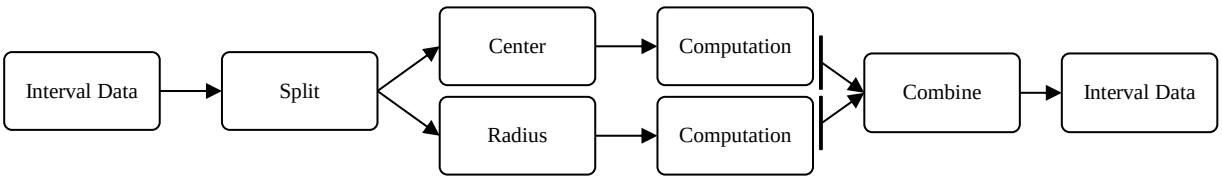


Figure 2. The interval data is split into centers and radii before computations.

2.3. Upper and Lower bounds (UL) Method

Since intervals are denoted by their bounds, $[\underline{x}, \bar{x}]$, the computation can also be performed on the lower $\underline{x}$, and upper $\bar{x}$ bounds separately as shown in Figure 3.

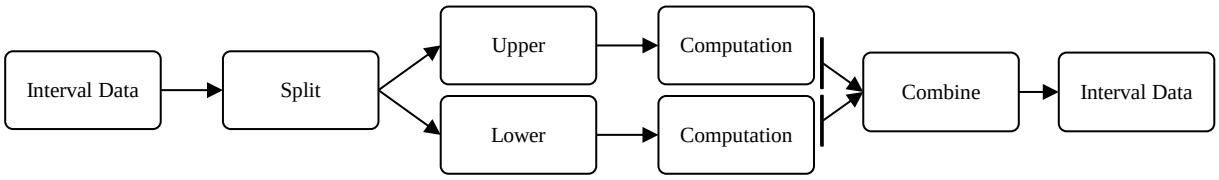


Figure 3. The interval data is split into upper and lower bounds before computations.

3. MULTI-SCALE ANALYSIS (MSA) OF SINGLE-VALUED DATA

Signals usually have temporal as well as frequency information. Process data collected from industrial plants are riddled with noise from sensor interference, measurement uncertainty from analog to digital conversions, and other factors. Trying to filter a signal indiscriminately will result in the loss of important features. Hence, it becomes beneficial to look at the signal at different scales where each scale highlights the features at a different frequency. This is helpful for fault detection applications [16]. In multiscale analysis, a signal is first decomposed into two parts: a low-frequency component called the approximate or scaled signal and a high-frequency component called the detail signal. The approximate signal is found by applying a low pass filter and the detail signal is found by applying a high pass filter on the original data. The approximate signal captures the trends in the data, while the detail signal captures the noise and some important features (such as sharp high-frequency changes). Subsequent levels of decomposition are obtained by repeating the decomposition steps described above on the newly found approximate signal. A schematic of a 3-level decomposition is shown in Figure 4.

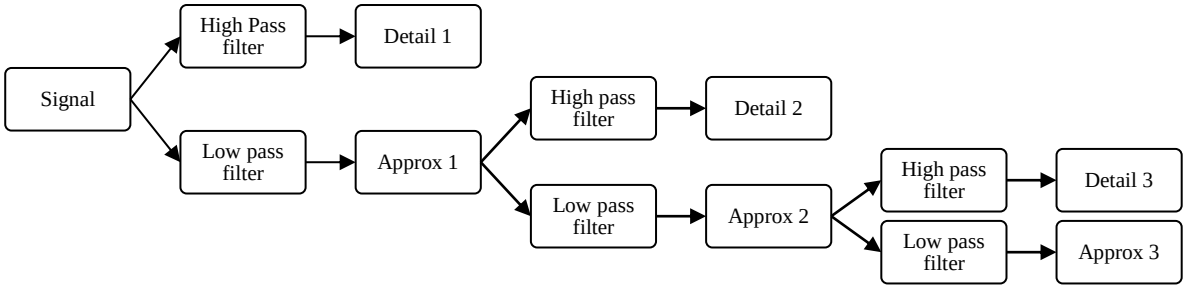


Figure 4. Schematic showing the MS decomposition of a signal using high and low pass filters.

3.1. Discrete Wavelet Transforms (DWT)

In discrete wavelet decomposition, data are down-sampled at every level of decomposition (Figure 5) such that the length of the approximate and the detail signals halves at every subsequent decomposition level. This is also known as Decimated MS decomposition.

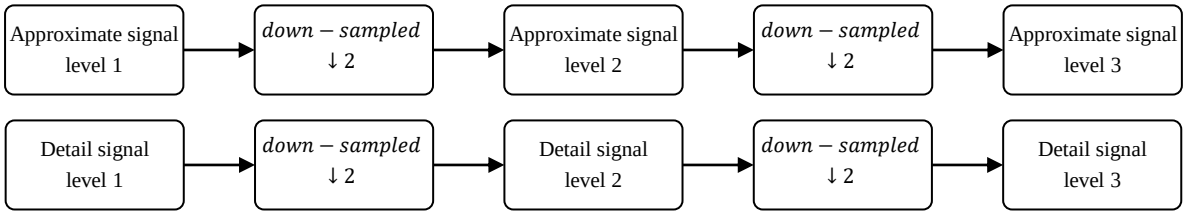
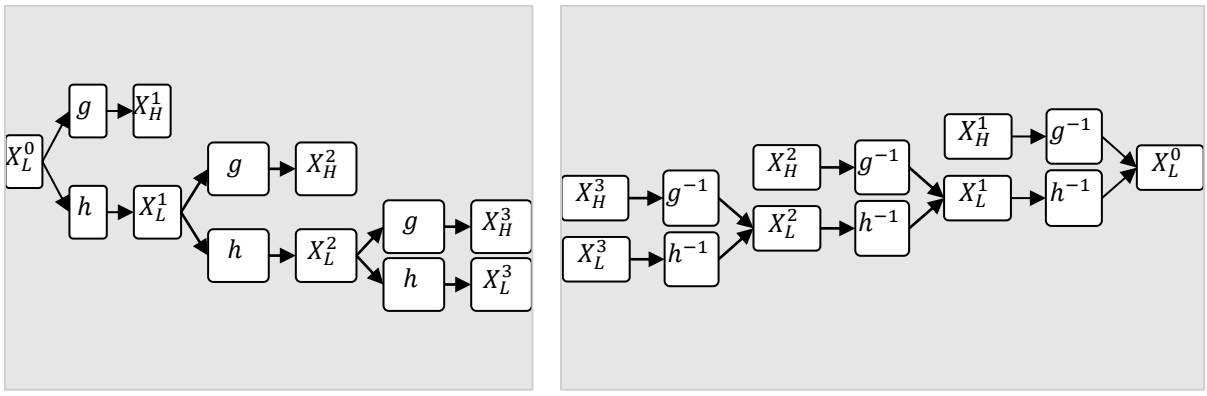


Figure 5. The filtered signals are down-sampled at every level in discrete wavelet decomposition.

Consider an input signal denoted as X_L^0. The signal is then decomposed into an approximate signal, X_L^1 and a detail signal, X_H^1. The subscripts L and H denote low and high frequencies, respectively, and the superscript indicates the level of decomposition, with 0 being the original signal. The decomposition is done using the appropriate low-pass filter coefficients, h and high-pass filter coefficients, g, which depend on the chosen mother wavelet function. A 3-level decomposition is illustrated using the schematic shown in Figure 6 (a). Also, the DWT decomposition of a data set that follows a standard normal distribution, $N(0,1)$ with a step change of 0.5 starting from the 1000th observation is shown in Figure 7. Figure 7 shows the potential benefits of MSA for fault detection at coarser (larger decomposition levels), where the peaks of the distribution of the two distinct parts of the signal (representing a fault-free and faulty regions) become more prominent and distinguishable. This is shown by the two clear peaks in the

distribution of the approximate signal at scale 3, as compared to almost a single peak in the distribution of the original signal on the top right in Figure 7.



(a) (b)

Figure 6. Schematic showing (a) 3 levels of discrete wavelet decomposition of a signal using high and low pass filters. (b) 3 levels of discrete wavelet reconstruction of a signal using high and low pass filters.

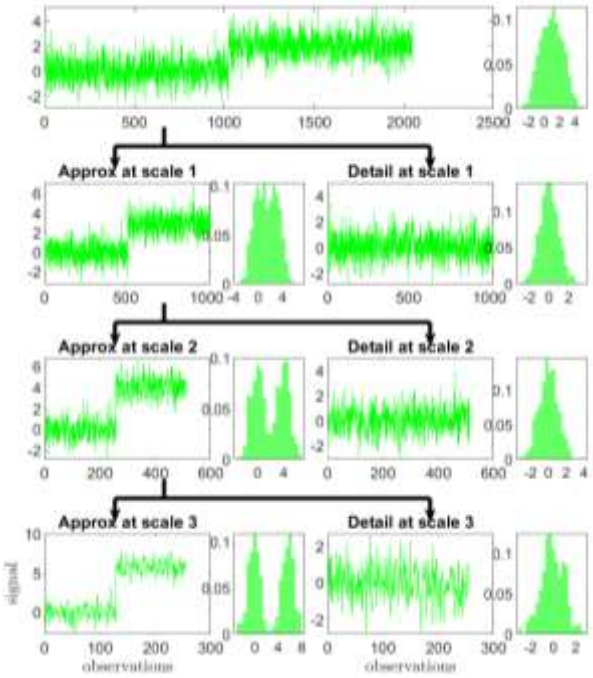
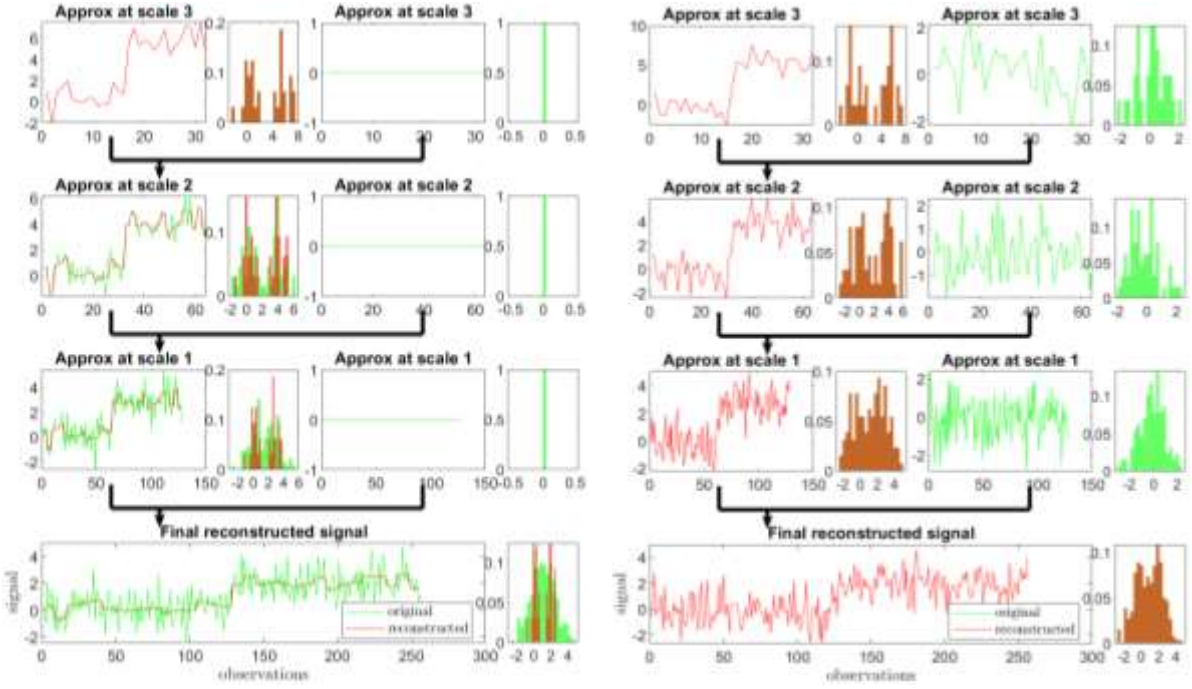


Figure 7. The multiscale discrete decomposition of a signal is shown.

The reconstruction of the data starts with the approximate and detail signal at the deepest (coarsest) scale and continues upward to the finer scales as shown in Figure 6 (b). Thresholding can also be applied to the detail signal at different scales (before reconstruction) to filter the data. Figure 8 (a) shows the reconstructed signal if all the details signals are discarded, and Figure 8 (b) shows the reconstructed signal if all the details signals are retained.



(a) (b)

Figure 8. The approximate and detailed signals are then combined to get a reconstructed signal: (a) if all details are discarded, and (b) if all details are retained.

4. INTRODUCTION TO INTERVAL MULTISCALE ANALYSIS (IMSA) METHODS

This section will introduce various ways in which multiscale analysis can be performed on interval data. This field is relatively new and literature review on it is sparse. One of the recent applications published in this area dealt with watermarking images using Interval Arithmetic and Dyadic Wavelet Transforms [3] where the authors replace the traditional compression algorithm of wavelet decomposition of digital images with an interval wavelet decomposition. The introduction of interval arithmetic causes redundancies in the computation, which allows the preservation of the watermarks even after the images undergo tampering. The research only uses it as a mathematical tool, and its properties have not been thoroughly explored. Also, the advantages of multiscale of interval data in the area of process engineering have not been fully explored. This section will introduce three methods to perform multiscale analysis on interval data.

4.1. Interval Arithmetic Method (IMSA-IA)

The idea behind this method is to apply an MSA algorithm similar to the one described in Section 3.1 for single-valued data, but on the interval data using interval arithmetic. The challenge with the use of Interval Arithmetic is well documented in the literature. When used with other classical methods such as Kalman Filtering, the interval variation tends to over-estimate the final states. Techniques that resets the interval bounds when certain thresholds are crossed have been proposed to avoid over-estimation [10]. Other proposed methods recalculate the interval boundaries based on the propagation of variance in the state error covariance [17]. The challenges encountered when dealing with interval data will be covered in more detail in Section 5.1.

4.2. Centers and Radii Method (IMSA-CR)

To avoid over-estimations in Interval Arithmetic, another proposed method relies on transforming the interval data into two single-valued data sets, namely, the centers and radii. A center is found by averaging the interval bounds, and the radius is estimated as half of the difference between these bounds. MSA is then performed on the two single-valued data sets traditionally as shown in Figure 9. For brevity, only the MS decomposition is shown in Figure 9. This approach has been utilized to improve fault detection using Interval Principal Component Analysis (IPCA) [6].

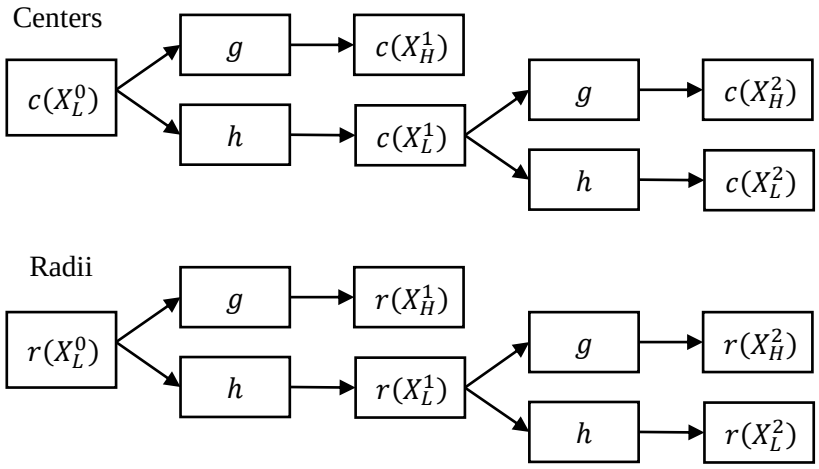


Figure 9. Schematic showing how MS decomposition is applied to the centers and radii separately.

4.3. Upper and Lower Method (IMSA-UL)

Similar to the centers and radii method, another method is to consider the two bounds of the interval (the upper and lower bounds) as two single-valued data sets. MSA is then performed on the two single-valued data sets separately in the traditional manner as shown in Figure 10.

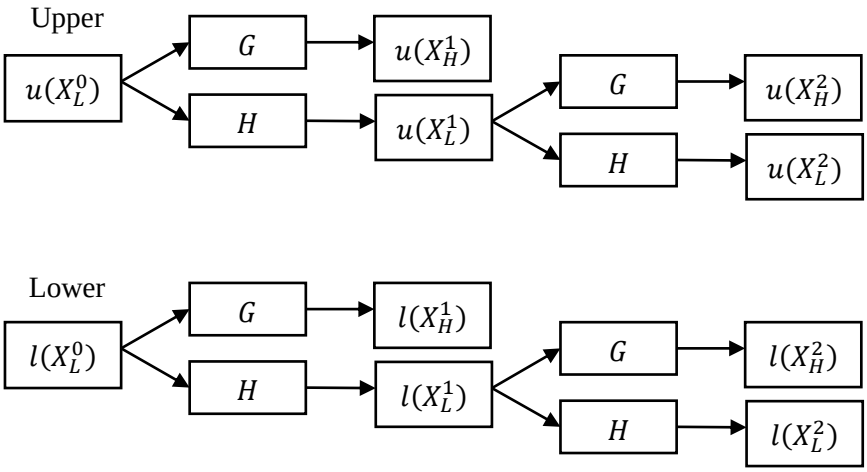


Figure 10. Schematic showing how MS decomposition is applied to the upper and lower bounds separately.

5. CHALLENGES IN INTERVAL MSA (IMSA)

The three methods covered in Section 4 for performing MSA on interval data were: 1) to use interval arithmetic, 2) to use ordinary arithmetic on the interval data converted into centers and radii, and 3) to use ordinary arithmetic on the interval data converted into upper and lower bounds. This section will present the challenges arising when applying the discrete MSA algorithm using all three methods. The interval data shown in the rest of the paper is generated by aggregating original data sets, which follow a normal distribution. The aggregation is done by computing the mean and standard deviation of the aggregated data as explained in Section 2.

5.1. Challenges in IMSA-IA

To understand the challenges with Interval arithmetic, MSA is performed on a simulated block data consisting of 2^{10} (1024) samples. The single-valued data are converted to intervals by aggregating every 2^3 samples into an interval point with its center at the mean and bounds at $\pm 2\sigma$ (to represent 95% of the aggregated data). The interval data points are then plotted such that they match the length of the original signal and envelope it comparably. The decomposition is shown in Figure 11. As mentioned before, interval arithmetic is conservative, which becomes more apparent at coarser scales where the interval bounds are larger than expected.

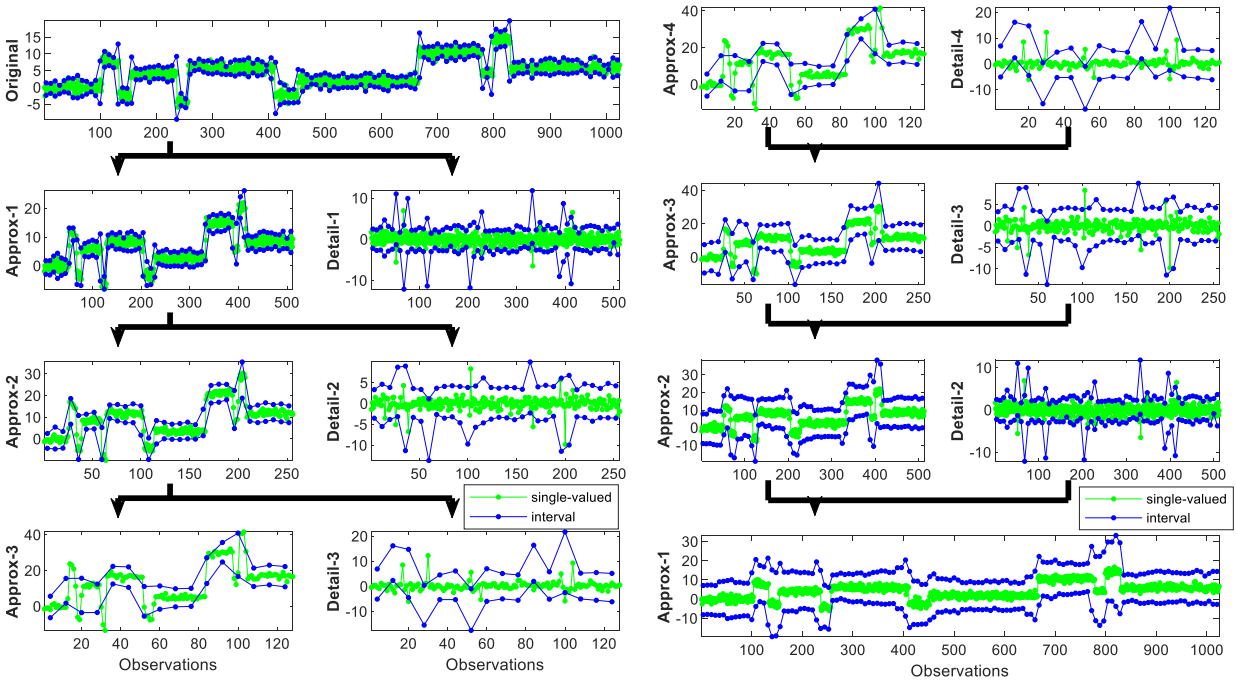


Figure 11. (a)Discrete wavelet decomposition of interval data using interval arithmetic, (b) Discrete wavelet reconstruction of interval data using interval arithmetic.

The decomposed signals in Figure 11 (a) are used to reconstruct the original signal. However, from Figure 11 (b), it is clear that the reconstructed signal overestimates the original signal, i.e., its bounds are larger than necessary. There are multiple reasons for this over-estimation, which are discussed in the following sub Sections.

5.1.1. Problem 1: Lack of Additive Inverse in Interval Arithmetic

One of the main limitations of interval arithmetic is the non-existence of an additive inverse. The result of subtracting an interval from itself does not result in zero. Instead, it returns a symmetric interval (symmetric about zero) with twice the width of the original interval. This becomes an issue in multiscale analysis, especially during reconstruction. Also, when interval data are decomposed and reconstructed, these non-zero terms resulting from the subtraction operations build up from one level to the next. The residual obtained as a result of the decomposition and

reconstruction of an interval signal to one decomposition scale using Haar is derived in Section A.1 of the Appendix. This paper will provide a solution to correct this issue in Section 6.1.

5.1.2. Problem 2: Effect of Multiple Iterations

Methods that require the repetitive application of functions (such as MSA, simulation of Discrete models, etc.) tend to overestimate the solution when performed on intervals. To illustrate this phenomenon, consider the linear transformation operation of the interval vector, $\boldsymbol{A} = \begin{Bmatrix} [\underline{x}, \bar{x}] \\ [\underline{y}, \bar{y}] \end{Bmatrix} = \begin{Bmatrix} [-1, 1] \\ [-1, 1] \end{Bmatrix}$, which rotated 60° counter-clockwise using the transformation matrix $R = \begin{bmatrix} cos\left(\frac{\pi}{3}\right) & -sin\left(\frac{\pi}{3}\right) \\ sin\left(\frac{\pi}{3}\right) & cos\left(\frac{\pi}{3}\right) \end{bmatrix}$. Consider a point $(x, y) = (1,1)$ in the interval vector $\boldsymbol{A}$ (shown as a green circle inside blue square representing the interval vector, and monitor how it moves as the linear transformation is repeated (see Figure 12). The linear transformation operation causes the interval to expand after every iteration. The results of 9 iterations on the x-coordinate of the single point, (x, y), and the interval of x-coordinate of the interval box, $\left([\underline{x}, \bar{x}], [\underline{y}, \bar{y}]\right)$, are shown in Figure 13.

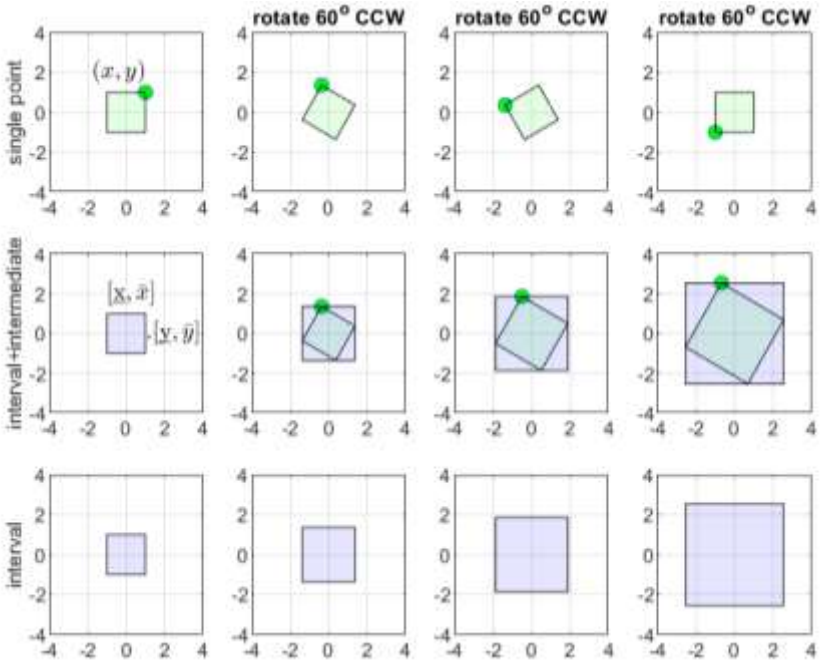


Figure 12. Applying a function such as a rotation repeatedly demonstrates how the intervals (blue) over-estimates as compared to the single values in a set (green).

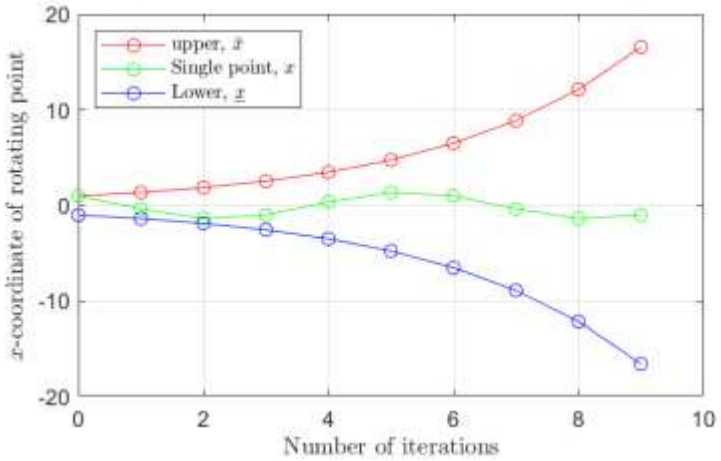


Figure 13. The evolution of the interval x-coordinate of the interval box and the x-coordinate of a single point is shown as it proceeds through multiple iterations.

Note that not all operations cause over-estimation. For example, if the linear transformation matrix is multiplied by 0.5 to scale down the transformed vectors to 50%, i.e., $R = 0.5 *$

$\begin{bmatrix} \cos\left(\frac{\pi}{3}\right) & -\sin\left(\frac{\pi}{3}\right) \\ \sin\left(\frac{\pi}{3}\right) & \cos\left(\frac{\pi}{3}\right) \end{bmatrix}$, then the intervals would converge (along with the true value) as shown in Figure 14. Over-estimation and convergence have been studied in detail in previous efforts [18] [19]. According to [19], the operation will converge if the spectral radius of the absolute values of the matrix is less than 1, i.e. $\rho(|A|) < 1$, where the spectral radius of a matrix is defined as the largest absolute value of its eigenvalues, i.e., $\rho(A) = \max\{|\lambda_1|, |\lambda_2|, \ldots |\lambda_n|\}$. In the case shown in Figure 14, the spectral radius of the transformation matrix, R is more than 1. A counter example has been shown in Figure 14 where the spectral radius is less than 1. The spectral radius of the Haar filter matrix for decomposition is 1.414 and thus will over-estimate the results. However, during reconstruction, the same filter is halved, and the spectral radius drops to 0.707 and will thus converge. Although this problem has been briefly studied in this paper, a specific solution has not been developed.

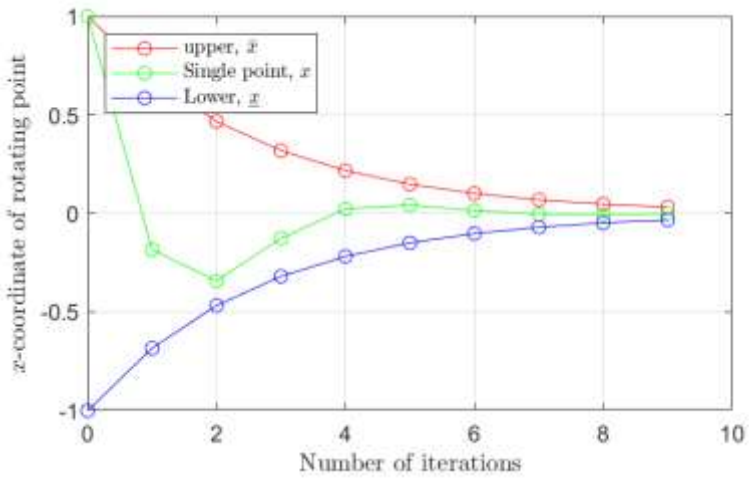


Figure 14. The convergence of the interval bounds over multiple iterations is shown.

5.2. Challenges common to IMSA-IA, IMSA-CR, and IMSA-UL

Some challenges are common to all the interval methods. In process engineering, the intervals are meant to represent uncertainties or variance in the measurements or models. However, from the mathematical perspective, intervals represent the boundaries of a set of values. Hence, the arithmetic operations performed with intervals conform to the operations between sets, and in most cases, does not reflect well on the uncertainties.

5.2.1. Effect of Convolution

A drawback of using interval arithmetic is that it tends to overestimate the spread or radius of the bounds when applying a function on interval data. As an example, consider two data sets, x_1 and x_2 each consisting of 2000 single-valued samples that follow a standard normal distribution, $X \sim N(0,1)$. The data are converted into intervals, $\left[\underline{x_1}, \overline{x_1}\right]$ and $\left[\underline{x_2}, \overline{x_2}\right]$ by aggregating every 100 samples. Then, the linear function $f(x_1,\ x_2) = \alpha x_1 + \beta x_2$ is applied to the single-valued data, x_1 and x_2, and also on the interval data, $\left[\underline{x_1}, \overline{x_1}\right]$ and $\left[\underline{x_2}, \overline{x_2}\right]$. α and β are chosen as $\frac{1}{2}$ for simplicity. Figure 15 shows the single-valued x_1 (green) and its interval $\left[\underline{x_1}, \overline{x_1}\right]$ (blue) in the top graph, and x_2 and its interval $\left[\underline{x_2}, \overline{x_2}\right]$ in the middle graph. The single-valued result $y = f(x_1, x_2)$ (in green) along with its interval $\left[\underline{y}, \bar{y}\right]$ (in blue) are shown in the bottom graph. The blue interval is what is expected of the result. The bottom graph in Figure 15 also shows the result of applying the function on the intervals (in red). The radius of the interval in red can be shown to be $r\left(\left[\underline{y}, \bar{y}\right]\right) = \frac{1}{2} r(x_1) + \frac{1}{2} r(x_2)$ for all the methods as detailed in the Appendix in Section A.2. The bounds estimated by the interval operation (red), however, is larger than the true expected interval (blue). This simple example demonstrates how the result of convoluting interval data is prone to overestimation. Such

overestimations are detrimental when convoluting a signal with low pass and high pass filters in MSA. This paper will provide a method to correct for this overestimation in Section 6.

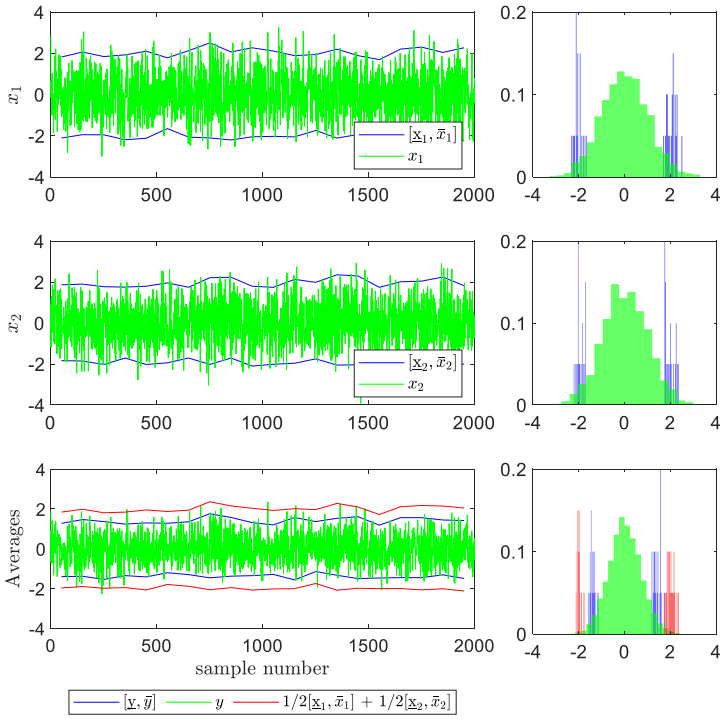


Figure 15. Comparing the convolution of single-valued data and interval data.

In the IMSA-CR method, the intervals are converted to centers and radii as described in Section 2.2. The centers and radii are then treated as single-valued data. In Figure 16 (a) and (b), the decomposition and reconstruction are shown using the centers and radii approach. As shown in the top graph in Figure 16 (a), and the bottom graph in Figure 16 (b), the final reconstructed signal looks exactly like the original signal without any over-estimation. Thus, conversion of the interval data into centers and radii form and applying normal arithmetic helps avoid the issues that arise when using interval arithmetic. So, it has inherently solved problem 1 that occurs in Interval

Arithmetic, which was discussed in Section 5.1.1. However, the issue related to convolution, which was discussed in Section 5.2.1, is still present. The bounds of the decomposed approximate interval (blue) over-estimate the single-valued decomposed approximate signal (green) in level 3 as shown in the bottom left graph in Figure 16 (a). Additionally, there are some other anomalies in how the intervals of the detail signals are computed. The bounds of the decomposed detail interval (blue) have collapsed closer to zero and do not represent the actual decomposed detail signal well as shown in the graphs on the right in Figure 16 (a). A further issue to note during the IMSA-CR decomposition is that the radius will be computed as negative in some instances. This can be remedied by finding the absolute value and then recomputing the bounds of the interval.

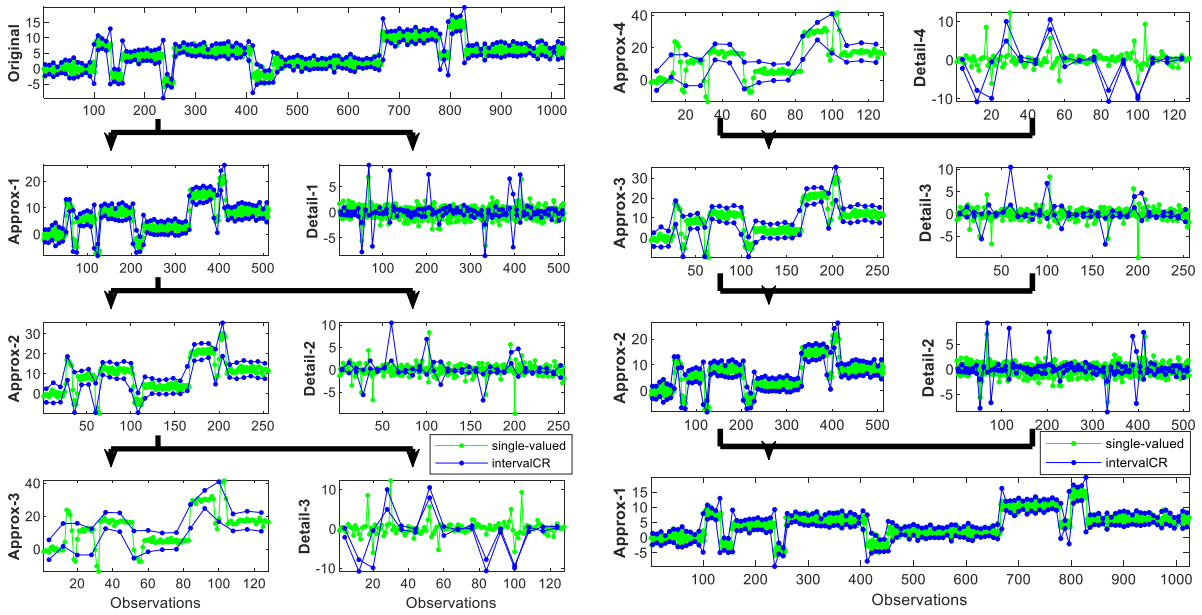
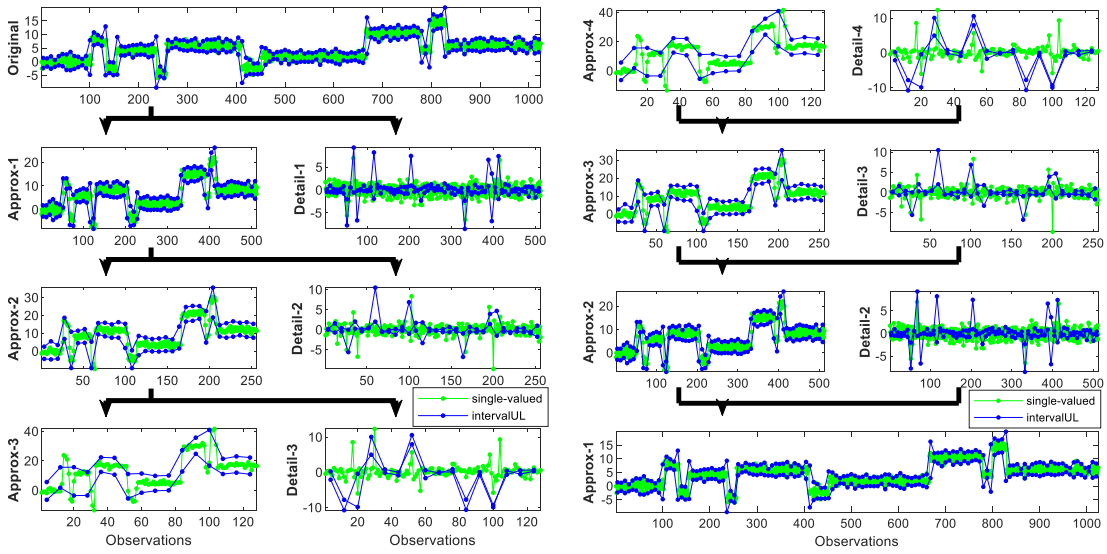


Figure 16. (a)Discrete wavelet decomposition of interval data using IMSA-CR, (b) Discrete wavelet reconstruction of interval data using IMSA-CR.

5.3. Challenges in IMSA when Applying Single-Valued Arithmetic on the Upper and Lower Bounds (IMSA-UL)

In the IMSA-UL method, the intervals are converted to their upper and lower bounds, and MSA is applied to these bounds as discussed in Section 2.3. The resulting decomposition and reconstruction are shown in Figure 17 (a) and (b), respectively. The results are similar to the ones obtained in the previous section using the IMSA-CR method. The bounds on the approximate interval signals (blue) are overestimated and the bounds on the decomposed detail interval signals (blue) are once again, too tight and not useful. Another issue to note during the IMSA-UL decomposition is that there are intervals whose lower bound is found to be larger than the upper bound. In these cases, the bounds then have to be switched around to maintain the proper definition of an interval, where the upper bound is always larger than the lower bound.



(a) (b)

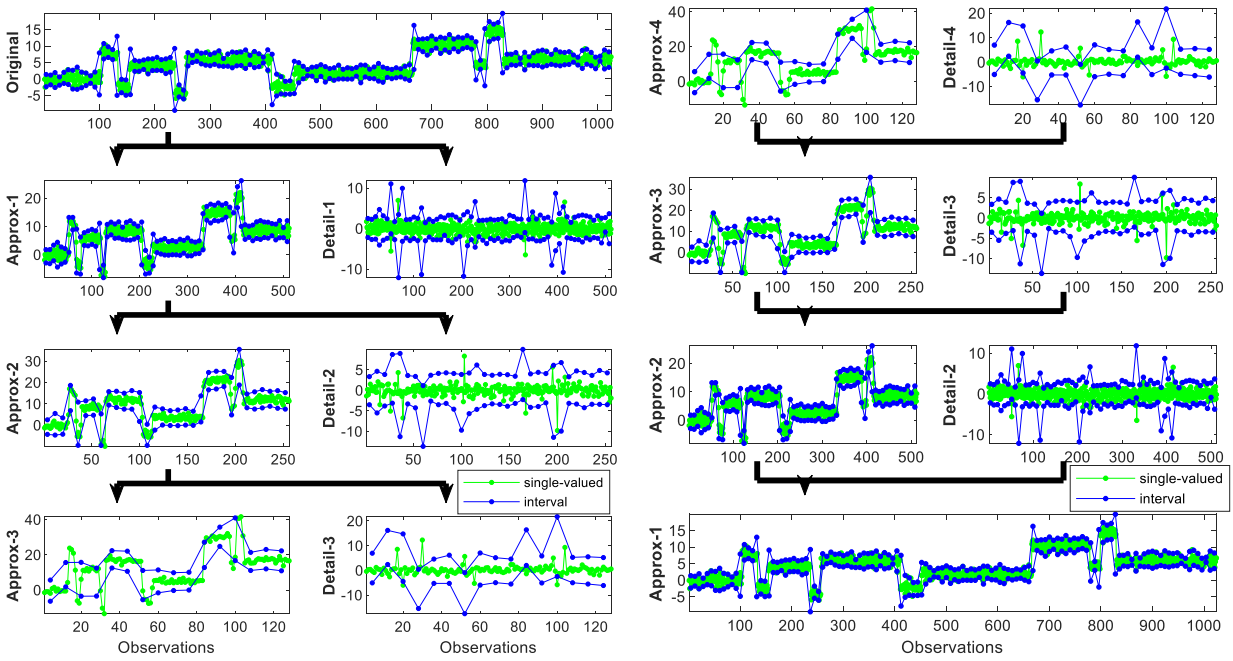
Figure 17. (a)Discrete wavelet decomposition of interval data using IMSA-UL, (b) Discrete wavelet reconstruction of interval data using IMSA-UL.

6. CORRECTED INTERVAL MULTISCALE ANALYSIS (CIMSA)

In this section, corrected interval multiscale analysis methods that deal with the challenges encountered in multiscale interval analysis described in Section 5 are developed.

6.1. Correct Interval Multiscale Analysis – Interval Arithmetic (CIMSA-IA)

As explained in Section 5.1.1, the subtraction operations on interval arithmetic introduce a residual that builds up as the interval data go through the MS reconstruction phase. During reconstruction, and because of the problem of additive inverse, terms need to cancel out to recover the original signal don't. As a result, there is a difference between the reconstructed interval signal and the true interval signal by an interval value of $r\left(x_{k_L}^{j}\right)[-1,1]$ when reconstructing the k^{th} sample from scale $j+1$ to j, which ultimately results in over-estimation. To deal with this over-estimation, the reconstructed intervals are corrected as detailed in the Appendix in Section A.1 and schematically illustrated in Section 6.1.1. The effect of applying the correction is shown in Figure 18 (b). When applying the correction, the signal decomposed in Figure 18 (a) is perfectly reconstructed. The original signal (top graph in Figure 18 (a)) and the reconstructed signal (bottom graph in Figure 18 (b)) are identical. To see the advantages of this correction, compare the results obtained in Figure 18 with those in Figure 11.



(a) (b)

Figure 18. (a) Shows the decomposition and (b) shows the reconstruction with the correction applied.

6.1.1. CIMSA-IA Algorithm

The steps taken to correct for the additive inverse issue in IA during MS reconstruction is shown in Figure 19. The top flowchart (read from right to left) shows a 2-level decomposition of a signal X_L^0. The bottom flowchart (read from left to right) then shows the reconstruction of the signal for the level 2 decomposed signals. The decomposed level 2 detailed and approximate interval signals X_H^2 and X_L^2 are used to reconstruct the approximate level 1 interval signal, ${X_R}_L^1$. The reconstructed signal is not the same as the actual approximate level 1 interval signal, X_L^1 due to the deviations caused by interval arithmetic. Hence the reconstructed signal is denoted with a subscript "*R*". The reconstructed signal is then corrected (shown in green dashed lines in Figure 19) by applying the function $P({X_R}_L^j, X_L^j)$ where $P\left({X_R}_L^j, X_L^j\right) = \left[\underline{x_R}_L^j + r\left(x_L^j\right),\ \overline{x_R}_L^j - r\left(x_L^j\right)\right]$ for each

interval point in the reconstructed approximate interval signal. The correction step tightens the interval by shifting the lower bound up and the upper bound down by a certain amount. The correction derived in this paper is exclusive for a Haar filter. For other filters, the correction methods would have to be rederived with the appropriate coefficients or a general framework needs to be developed. The derivation of the correction value is presented in the Appendix in section A.3.

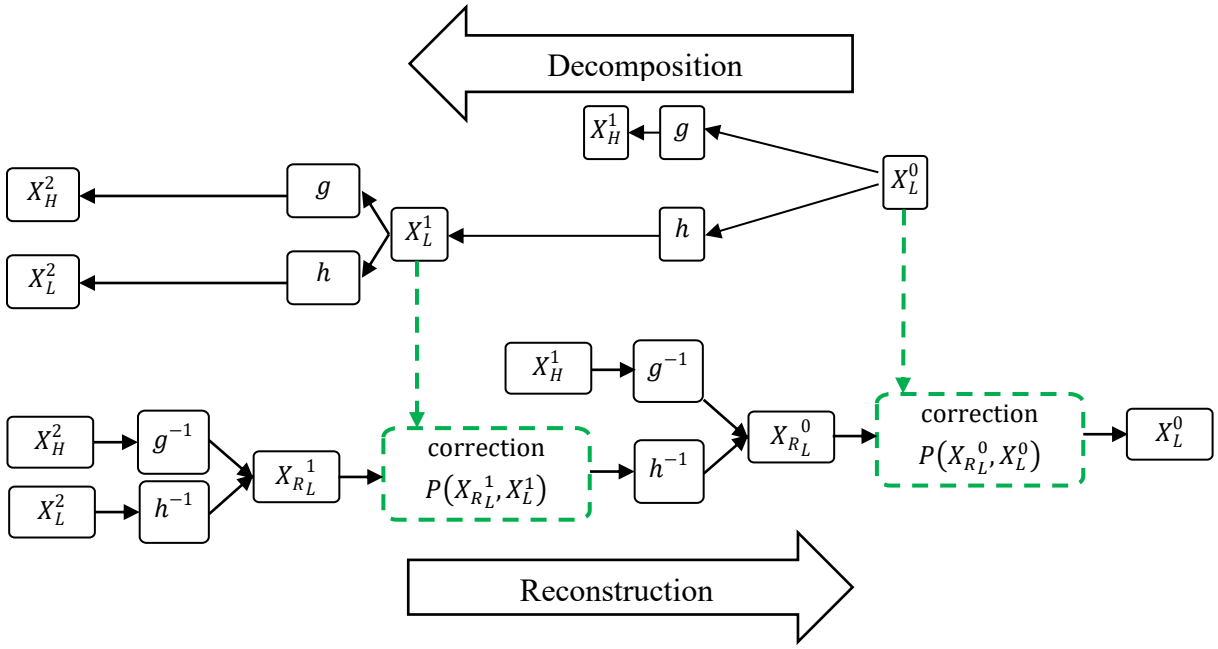


Figure 19. Schematic showing the correction being applied for 2 levels of discrete reconstruction of a signal

6.2. Corrected Interval Multiscale Analysis – Centers and Radii (CIMSA-CR)

The conclusion drawn from Section 5.2.1 is that the convolution of interval data will always overestimate the bounds of the resulting signal. A more reasonable estimate of the bound can be found using the propagation of uncertainty [20]. When doing calculations with variables whose uncertainties are known (such as interval data), there will be uncertainties associated with

the results. The propagation of uncertainty allows for a better estimation of the uncertainties in the results. Therefore, to correct for the over-estimation in Interval MSA, the radii of the intervals are estimated from the principle of propagation of uncertainty as described in the Appendix in Section A.2 and schematically illustrated in Section 6.2.1.

6.2.1. CIMSA-CR Algorithm

The effect of convolution on interval data is seen in MSA during decomposition and reconstruction using Centers and radii. The steps to correct the signal's radii are shown in Figure 20. The top flowchart shows the decomposition of the centers, where correction is not required. The bottom flowchart shows the decomposition of the radii. The radii are not decomposed by convoluting with the filters in the traditional sense. The radii are instead computed using the equation derived from the propagation of uncertainty. This is denoted in the flowchart as Q_g and Q_h, where $Q_g = \sqrt{\sum\left(\frac{\partial(X_L * g)}{\partial x_i} r(x_i)\right)^2}$ and $Q_h = \sqrt{\sum\left(\frac{\partial(X_L * h)}{\partial x_i} r(x_i)\right)^2}$, where $X_L * h$ and $X_L * g$ is the convolution of the data with the appropriate filter coefficients.

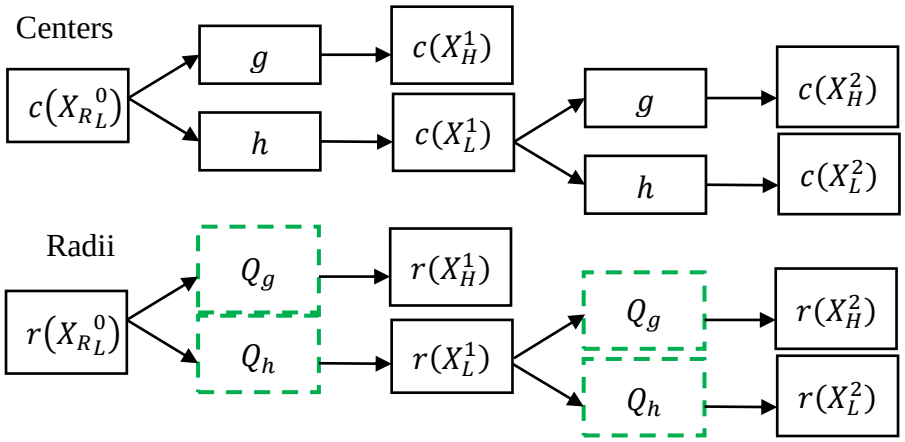


Figure 20. Schematic showing the radius being estimated using the equation derived from the propagation of uncertainty.

The reason this method works well is that the radius signifies the uncertainty in the data, and the principle of propagation of uncertainty gives a better estimate of the radius than merely doing computations with the bounds, or the radii themselves. The same process is then repeated during the reconstruction stage. However, the radii would be then computed using $Q_R = \sqrt{\sum \left(\frac{\partial(X_L * h^{-1} + X_H * g^{-1})}{\partial x_i} r(x_i)\right)^2}$ during reconstruction.

6.3. Corrected Interval Multiscale Analysis – Upper and Lower (CIMSA-UL)

The correction applied in the above section can also be applied to the IMSA-UL method. The radius is first calculated from the upper and lower bounds. The radius is then corrected, and then the upper and lower bounds are recomputed from the adjusted radius. The rest of the process is the same.

6.3.1. CIMSA-UL Algorithm

The effect of convolution on interval data is seen in MSA during decomposition and reconstruction using the upper and lower bounds. The steps to correct the effect of this problem is shown in Figure 21 for a signal using only one level decomposition for brevity. The idea is to convert the upper and lower bounds to centers and radii, and then apply the same propagation of uncertainty correction described earlier Since the propagation of uncertainty deals with errors in radii, this conversion makes it much easier to be applied. Once the new radii are computed, the signal is reverted to the upper and lower bound form. A similar process is performed during reconstruction.

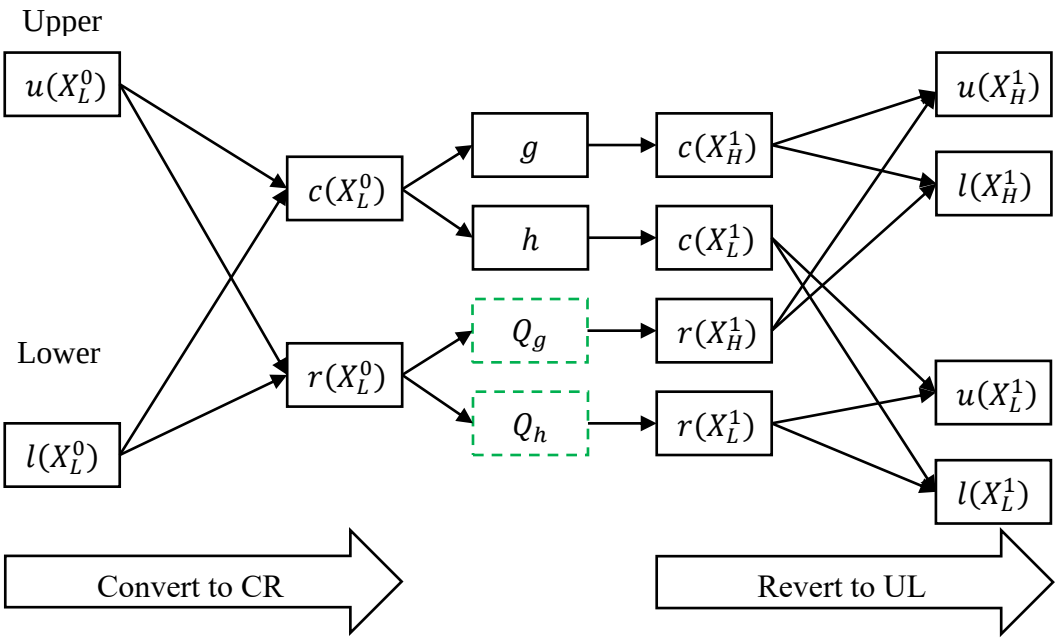


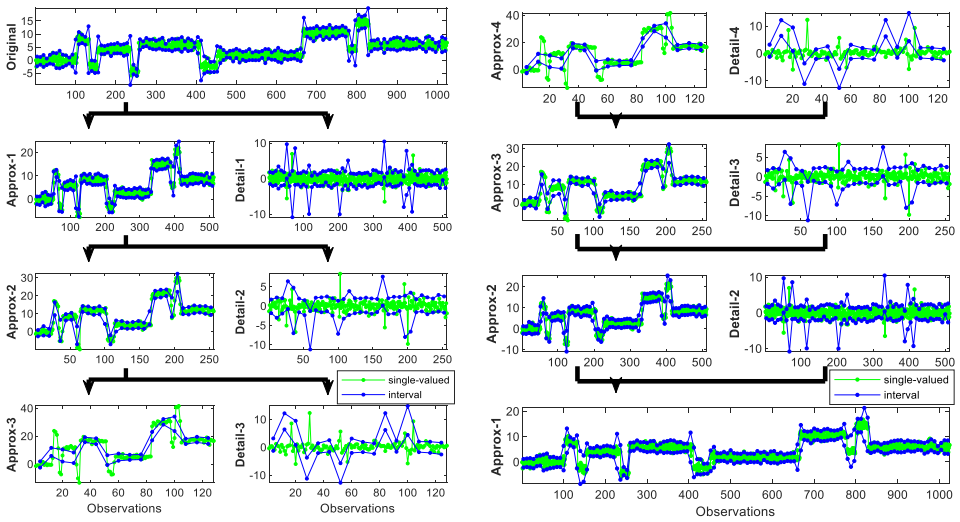
Figure 21. Schematic showing the correction of the radius by re-estimating it using the equation derived from the propagation of uncertainty.

7. ILLUSTRATIVE EXAMPLES

In this section, the capabilities of the proposed CIMSA algorithms are illustrated using various synthetic examples.

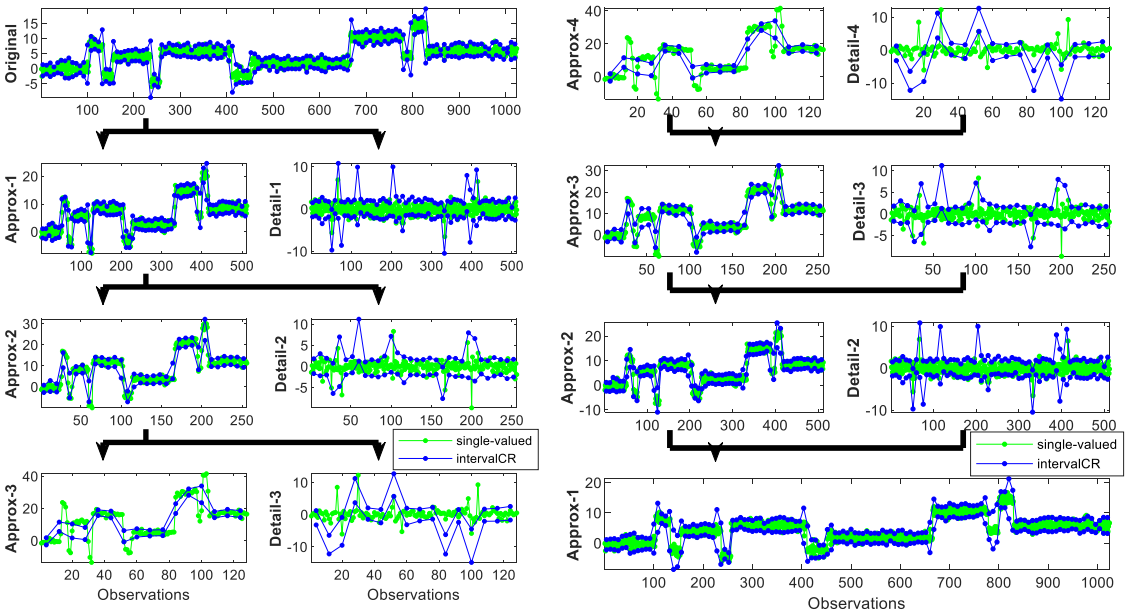
7.1. Comparison of the Three Corrected IMSA (CIMSA) Methods for Decomposition and Reconstruction

The developed CIMSA methods have been used to decompose and reconstruct the same signal used in Section 5, and the results for the various methods are shown in the following Figures: CIMSA-IA (Figure 22), CIMSA-CR (Figure 23), and CIMSA-UL (Figure 24). When compared to the methods without corrections applied, such as the IMSA-IA (Figure 16), IMSA-CR (Figure 17), and the IMSA-UL (Figure 18), the bounds computed using the corrected methods estimate the true single-valued data (no overestimation or underestimation).



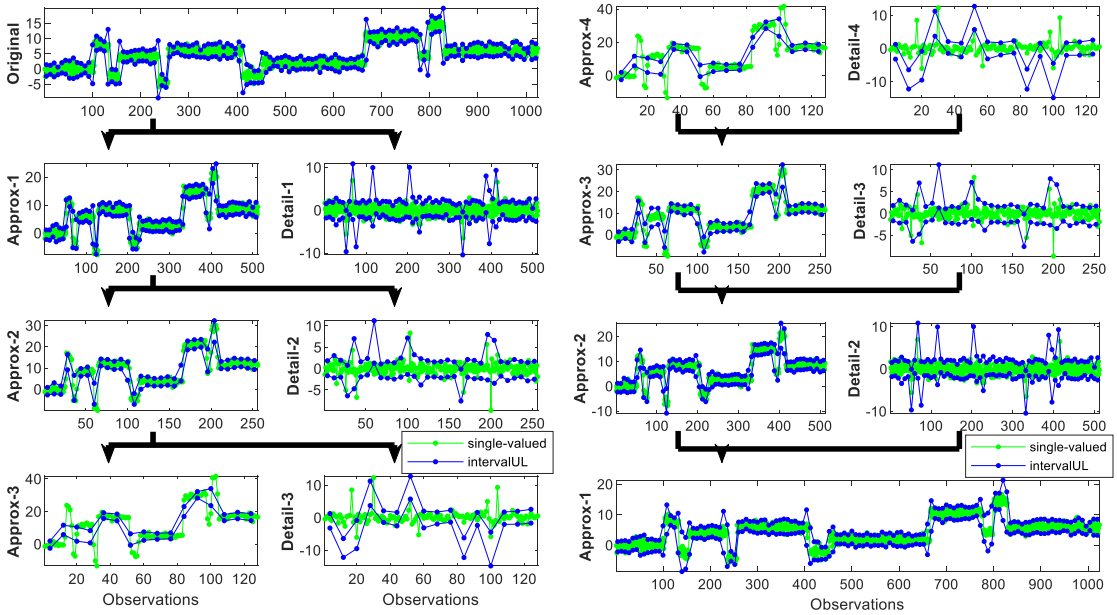
(a) (b)

Figure 22. (a) Shows the decomposition and (b) shows the reconstruction with the CIMSA-IA



(a) (b)

Figure 23. (a) Shows the decomposition and (b) shows the reconstruction with the CIMSA-CR



(a) (b)

Figure 24. (a) Shows the decomposition and (b) shows the reconstruction with the CIMSA-UL

7.2. Application of MS filtering

MS Filtering is an effective way of filtering process signals. Since MS decomposition decomposes the signal based on its frequencies (approximate and detail signals at various scales), targeted filtering can be applied to the signal based on ranges of frequencies and ranges of time (observations). MS decomposition allows the separation of trends and features into approximate and detail signals. Since noise is typically high-frequency fluctuations, the detail signals will mostly contain the noisy components of the signals along with few features such as sharp or abrupt changes. Discarding the detail signal during the reconstruction will provide a cleaner signal. However, it could result in the loss of important features. To avoid this, thresholding for filtering is used [21]. During reconstruction, a threshold is applied to the detail signals to discard unimportant features (noise fluctuations). This is shown as blue dashed boxes in Figure 25, which results in a reconstructed signal being different from the decomposed signal. Therefore, subscript "R" is used to denote the reconstructed signals. If the magnitude of a value in the detail signal exceeds a predetermined threshold, it is retained. If it falls below, then the value is replaced with a 0. This is referred to as hard thresholding and is illustrated in Figure 26. The other variation of thresholding is called soft thresholding. In soft thresholding, the points below the threshold are set to 0. However, the points that exceed the threshold are reduced by the amount of the threshold. In this paper, only hard thresholding is used.

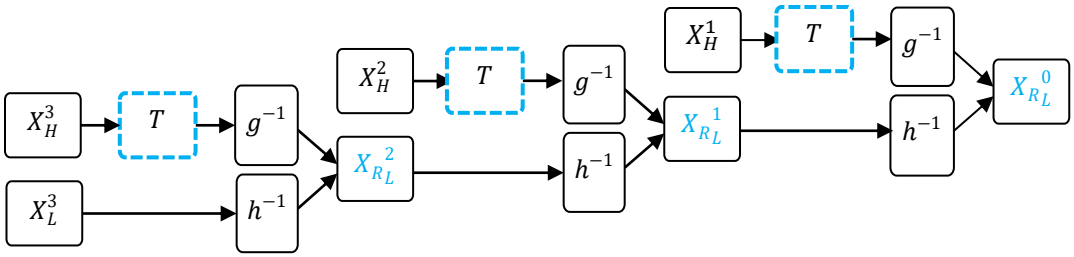


Figure 25. The threshold is applied to the detail signals during reconstruction.

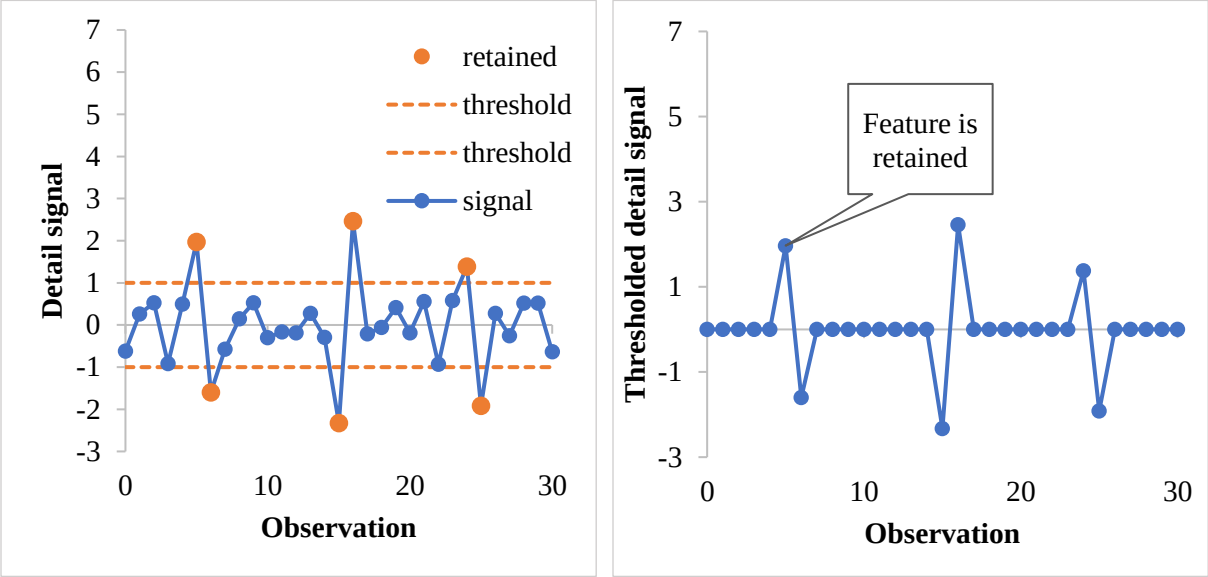


Figure 26. The hard thresholding of a sample signal is shown.

The threshold, t_m can be computed using the following equation as proposed in [21].

$$t_m = \sigma_m \sqrt{2 \log n}$$

where n is the signal length and σ_m is the standard deviation of the detail signal estimated as

$$\sigma_m = \frac{1}{0.6745} median\{X_H^j\}$$

In this paper, the MS filtering algorithm is modified to work with interval data along with the various CIMSA algorithms developed. The effectiveness of the Interval MS filter will then be

studied using synthetic examples. Thresholding is done separately on each of the center, radii, upper and lower bounds for the CIMSA-CR and CIMSA-UL as the idea is to treat them separately. For CIMSA-IA, the interval is treated as a single unit. Hence if either the upper or lower bound falls below the threshold, the interval is set to 0.

To gauge the performance of the various methods when utilized in data filtering, the Mean Square Error (MSE) between the original signal and the filtered signal is used. The MSE of the reconstructed signal, X_R is the average of the MSE of upper and lower bounds, $MSE(X_R) = \frac{1}{2}\left(MSE(\overline{X_R}) + MSE\left(\underline{X_R}\right)\right)$. The same metric is used for all three methods. The original single-valued signal is first aggregated by taking the mean (or centers) of a consecutive subset of samples to form a signal of the same length as the interval signals. The MSE of upper and lower bounds for the original signal is calculated as $MSE(\overline{X_R}) = \frac{1}{n}\sum_{i=1}^{n}\left(\overline{x_{R_i}} - \overline{x_i}\right)^2$ and $MSE\left(\underline{X_R}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\underline{x_{R_i}} - \underline{x_i}\right)^2$, where $\left[\overline{x_{R_i}}, \underline{x_{R_i}}\right]$ is the filtered and reconstructed interval signal and x_i is the center of the aggregated original signal.

A synthetic data set consisting of 2^{11} samples are generated with some added noise such that the Signal to Noise ratio is 8. The data is then converted into interval data by aggregating every 2^2 samples as shown in the top chart in Figure 27. The MSE computed from the noisy data is 1.039 and the one computed from the interval data is 5.565. Just for comparison, the single-valued data have been filtered as per the algorithm covered in [21] and are shown in the bottom chart in Figure 27, which results in an MSE of 0.184.

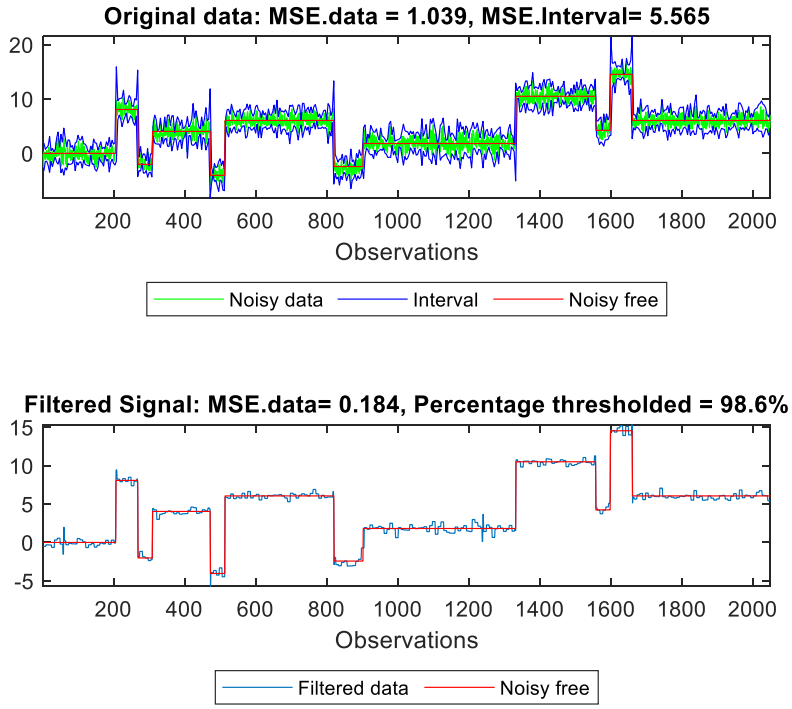


Figure 27. Synthetic data containing step changes are filtered using MS filtering.

The three IMSA methods are first applied without any corrections to filter the data. The filtering results using a decomposition depth of 3 using IMSA-IA, IMSA-CR, and IMSA-UL are shown in Figure 28 (a). IMSA-UL and IMSA-CR show minimal improvements with a lower MSE (Original MSE = 5.565). The filtering done with IMSA-IA was counterproductive and has increased the MSE. The variations in the MSE for the different methods as a function of the depth of decomposition is shown in Figure 29 (a). IMSA-CR seems to be the better method while IMSA-IA is the worst. And the performance of all methods degrades at a higher depth of decomposition.

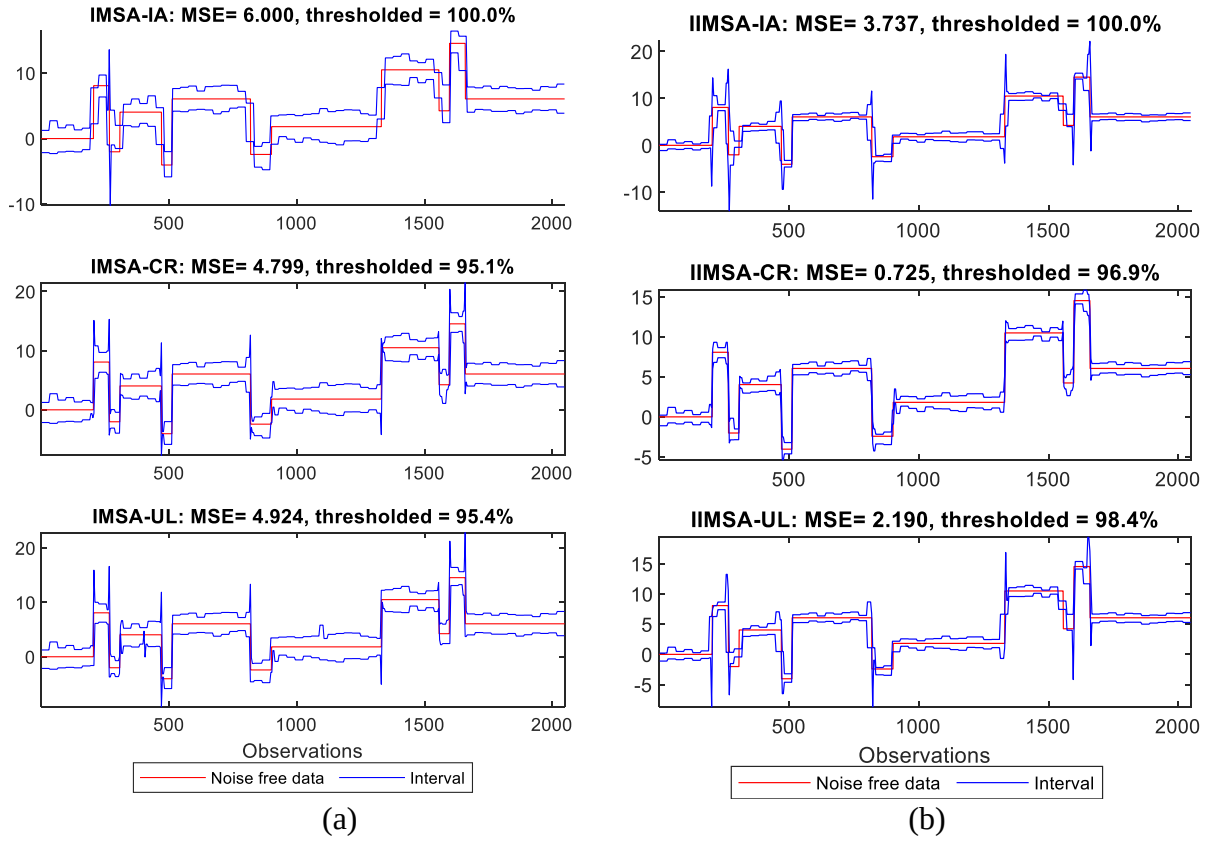


Figure 28. The data is filtered by decomposing to a depth of 3 with the different methods (a) without corrections (IMSA), (b) with corrections (CIMSA).

The corrections discussed in Section 6 are then applied and the filtering step is repeated. This time, a significant improvement in the performance of filtering can be observed as shown in Figure 28 (b). The MSE as a function of the depth of decomposition shown in Figure 29 (b) reveals that the best option among the three methods is the CIMSA-CR. For this particular data set, a decomposition depth of 6 gave the lowest MSE value for CIMSA-CR. The interval filtered data using CIMSA-CR at a depth of 6 is shown in Figure 30.

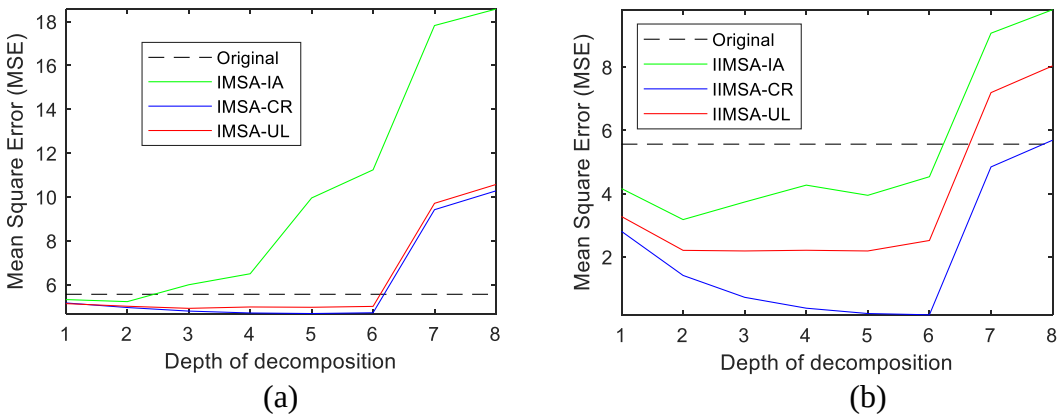


Figure 29. The MSE computed at various depths of decomposition for the different methods (a) without corrections, (b) with corrections.

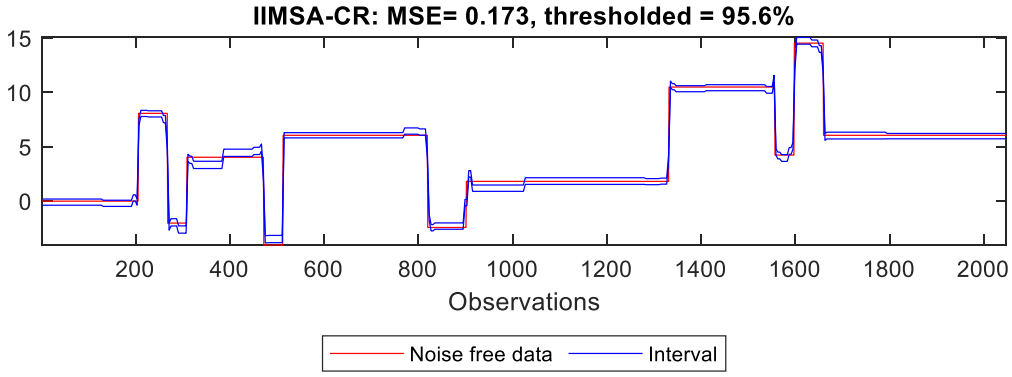
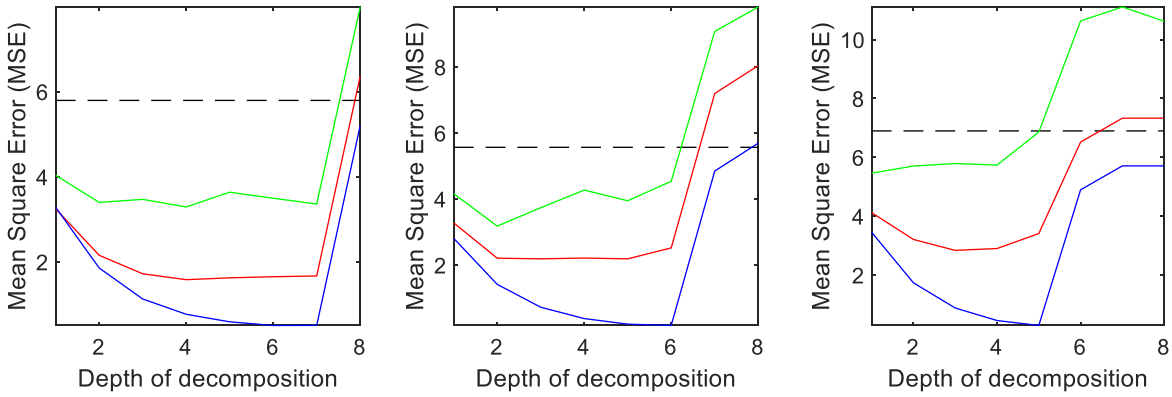


Figure 30. The data filtered at a depth of 6 with CIMSA-CR gave the lowest MSE.

7.2.1. Effect of the Number of Aggregated Samples

The choice of the number of samples to aggregate when converting single-valued data to interval data also plays a role in determining the filtering performance. The CIMSA-CR remains the best performing method. However, the optimum depth of decomposition is reduced as we aggregate more samples. This is the case because as more data are aggregated, the length of the interval data reduces, which reduces the noise content, and thus a smaller decomposition depth is needed. Starting with a data set of 2^{11} and aggregating every 2^1 samples result in an interval data

of size 2^{10}, whereas aggregating every 2^3 samples result in an interval data of size 2^9. Hence, the optimum depth of decomposition in both cases is close to half of the maximum decomposition depth.

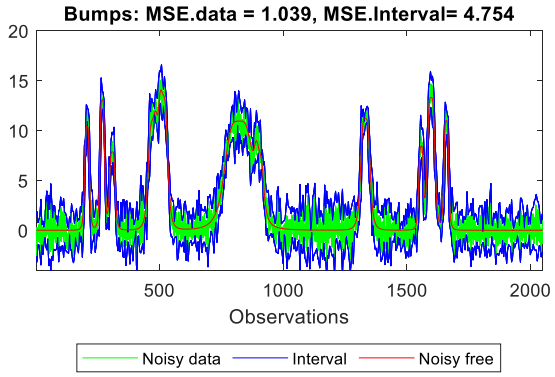


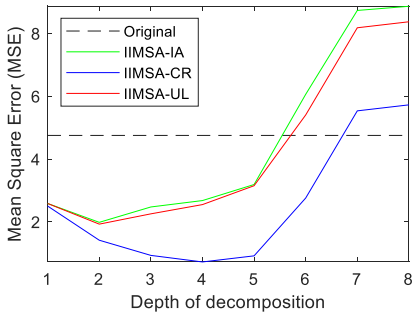
(a) (b) (c)

Figure 31. The MSE computed at various depths of decomposition for the different methods where the number of aggregated samples are (a) 2^1 samples, (b) 2^2 samples, and (c) 2^3 samples.

7.2.2. Effect of Feature Shapes

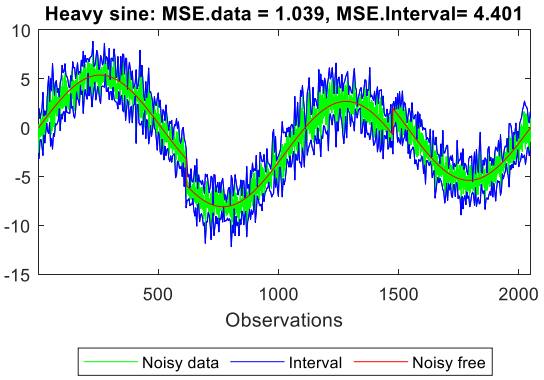
The shape of the features in the input data is important to consider when judging the performance of the filtering method. Hence, different types of signals were filtered and their MSE at various depths were computed. The different signals are Blocks (Figure 28 (b) and Figure 29 (b)), Bumps (Figure 32), Heavy Sine (Figure 33), Doppler (Figure 34). In all cases, the CIMSA-CR performs better with a lower MSE at all depths. However, for the Heavy sine, all the methods perform equally well. This could be due to the choice of the mother wavelet (Haar in this example), which is not an appropriate wavelet as it does not match the shape of the signal, resulting in all three methods being indistinguishable.

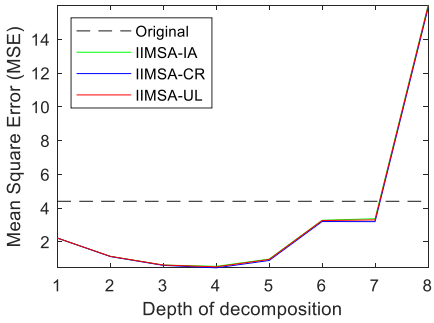




(a) (b)

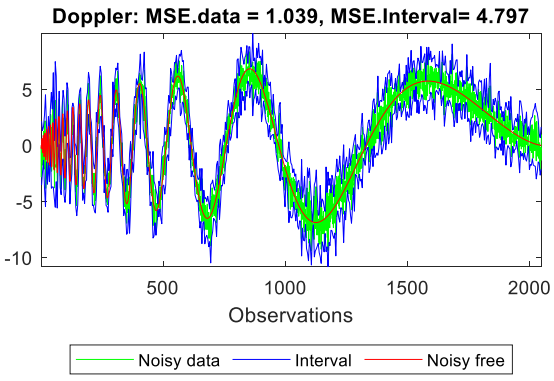
Figure 32. (a) A bumps signal. (b) The performance of different methods on the signal.

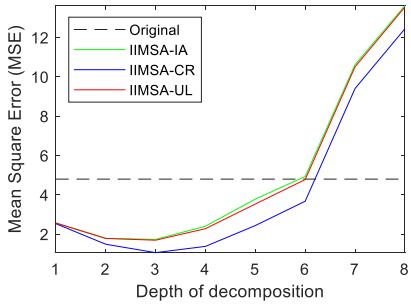




(a) (b)

Figure 33. (a) A heavy sine signal. (b) The performance of different methods on the signal.





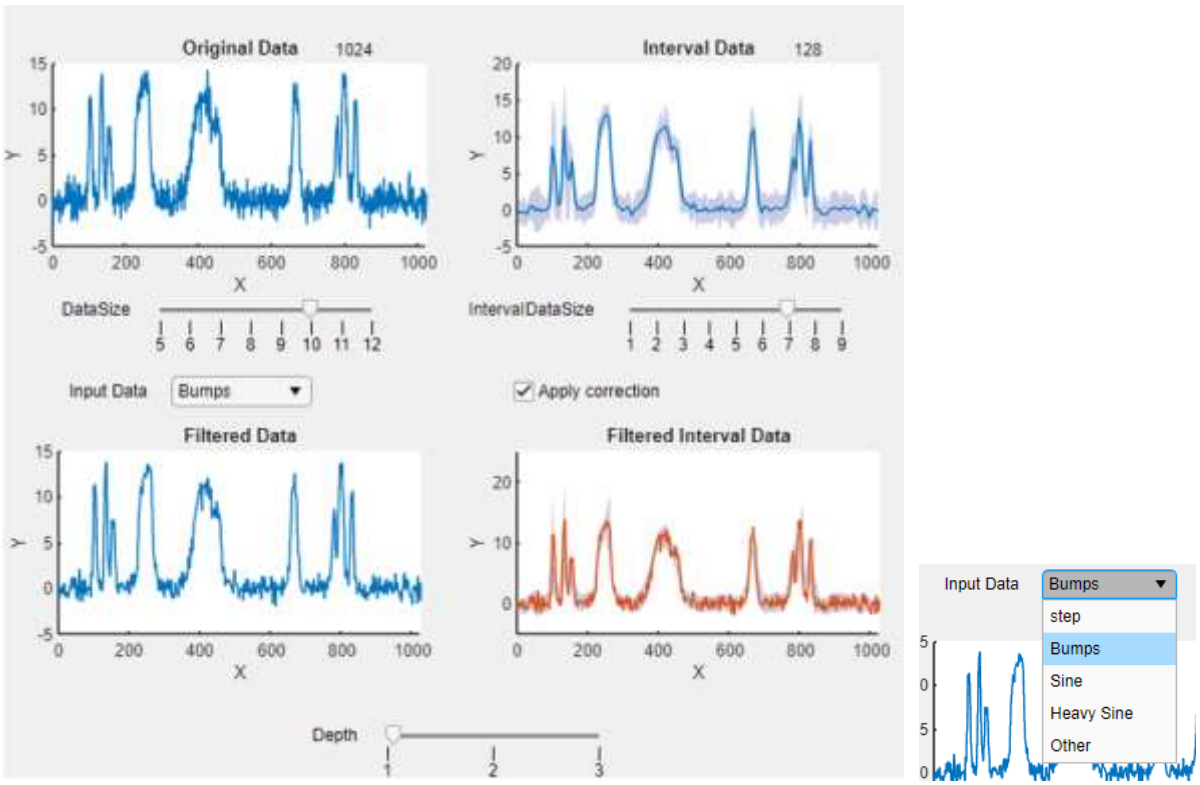
(a) (b)

Figure 34. (a) A Doppler signal. (b) The performance of different methods on the signal.

8. CIMSA-CR APP CREATED WITH MATLAB

As detailed in the previous section, the CIMSA-CR has demonstrated a better performance in most test cases. Therefore, an application was built in MATLAB based on the CIMSA-CR algorithm. The application is a Graphical User Interface (GUI) built-in MATLAB's App developer. It allows users to easily process their data, try different parameters, and see the results immediately. Figure 35 shows the interface of the application. The left quadrant shows the original single-valued-data, and the right quadrant shows the interval representation of the same data. The top right graph is the interval generated from the original single-valued data. The slider under the original data is used to set the size of the generated data in powers of 2 (length of data = 2^{10}). It is important to keep the size as powers of 2 since DWT requires a dyadic data set. The dropdown under the top-left graph is used to select the shape of the generated data. The slider under the top right graph is used to set the size of the final interval data in powers of 2 (length of interval data = 2^7). Based on this choice, the appropriate interval size is calculated and implemented. In the particular case shown in Figure 35(a), there is a size reduction of 2^3.

In Figure 35(a), the graphs at the bottom are the data filtered using MSA. The bottom left graph shows the original single-valued data filtered using MSA, while the bottom right graph shows the interval data filtered using the CIMSA-CR along with the filtered single-valued data for comparison. The depth of decomposition can be controlled using the slider at the bottom. To load external data, the user can choose the "Other" option from the dropdown which will prompt the user to select a data file. The data file must be a single value data vector. The data set is then trimmed to a length of the nearest power of 2.



(a) (b)

Figure 35. (a) The graphical program created using the CIMSA-CR algorithm is shown here. (b) the options available in the dropdown menu.

9. CONCLUSION

In this paper, the challenges arising from applying MSA on interval data were studies and methods for dealing with these challenges were proposed. One of the key issues when working with intervals is interval inflation/over-estimation. This was mitigated by developing three corrected algorithms that were presented in Section 6. The results obtained using the corrected Interval MS decomposition and reconstruction algorithms matched the actual single-valued data at multiple scales. The overestimations in the approximate signals and the underestimations in the details signals were corrected. The algorithm developed was also utilized to perform multiscale filtering of noisy interval data. The CIMSA-CR which is the center and radii approach performed the best amongst the three methods for the filtering application. This can be attributed to the additional filtering effect that averaging has when the centers of the intervals are calculated. It has been shown that the optimum depth of decomposition in filtering depends on the size of the aggregated samples and the shape and features in the data. Prior knowledge about the expected shapes will allow one to choose the best level of decomposition for filtering purposes.

This work can be extended to include other wavelets such as the Daubechies wavelets and other multiscale analysis methods, such as stationary wavelet transforms. Several future directions can be followed. For example, optimization can be used to determine the right choice of interval size and the depth of decomposition. Also, although the results for CIMSA-IA are poor compared to the other methods, IA offers additional tools, that could be used to optimize the algorithm further.

REFERENCES

[1] P. Liang, C. Deng, J. Wu, Z. Yang, J. Zhu and Z. Zhang, "Compound Fault Diagnosis of Gearboxes via Multi-label Convolutional Neural Network and Wavelet Transform," *Computers in Industry,* vol. 113, p. 103132, 12 2019.

[2] J. Gao, B. Wang, Z. Wang, Y. Wang and F. Kong, "A Wavelet Transform-Based Image Segmentation Method," *Optik,* p. 164123, 12 2019.

[3] M. Teruya and O. Ryuji, "A Blind Digital Image Watermarking Method Based on the Dyadic Wavelet Transform And Interval Arithmetic," *Applied Mathematics and Computation,* vol. 226, no. January, pp. 306-319, 2014.

[4] H. N. Nounou and M. N. Nounou, "Multiscale Fuzzy Kalman Filtering," *Engineering Applications of Artificial Intelligence,* vol. 19, pp. 439-450, 8 2006.

[5] M. Z. Sheriff and M. N. Nounou, "Enhanced Performance of Shewhart Charts Using Multiscale Representation," in *2016 American Control Conference (ACC)*, 2016.

[6] N. Basha, M. Nounou and H. Nounou, "Multivariate Fault Detection and Classification Using Interval Principal Component Analysis," *Journal of Computational Science,* vol. 27, pp. 1-9, 7 2018.

[7] R. Moore, Introduction to Interval Analysis, Philadelphia, PA: Society for Industrial and Applied Mathematics, 2009.

[8] L. Jaulin, M. Kieffer, O. Didrit and É. Walter, Applied Interval Analysis, Springer London, 2001.

[9] G. Chen, J. Wang and L. S. Shieh, "Interval Kalman Filtering," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 33, pp. 250-259, 1 1997.

[10] J. Xiong, C. Jauberthie, L. Trave-Massuyes and F. L. Gall, "Fault Detection Using Interval Kalman Filtering Enhanced by Constraint Propagation," in *52nd IEEE Conference on Decision and Control*, 2013.

[11] B. Li, C. Li, J. Si and G. P. Abousleman, "Interval Least-Squares Filtering With Applications to Robust Video Target Tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, 2008.

[12] A. Motwani, "Adaptive And Interval Kalman Filtering Techniques in Autonomous Surface Vehicle Navigation: A Survey," Marine and Industrial Dynamic Analysis, School of Marine Science and Engineering, Plymouth University, Plymouth, 2012.

[13] C. Fu, Y. Liu and Z. Xiao, "Interval Differential Evolution With Dimension-Reduction Interval Analysis Method for Uncertain Optimization Problems," *Applied Mathematical Modelling,* vol. 69, pp. 441-452, 5 2019.

[14] A. Meng, H. Wang, S. Aziz, J. Peng and H. Jiang, "Kalman Filtering Based Interval State Estimation For Attack Detection," *Energy Procedia,* vol. 158, pp. 6589-6594, 2 2019.

[15] M. S. Grewal, "Kalman Filtering," in *International Encyclopedia of Statistical Science*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2011, pp. 705-708.

[16] B. Malluhi, "Improved Fault Detection And Isolation Using Enhanced Multiscale Principal Component Analysis: Algorithms And Applications," in *M.S. Thesis*, College Station, Texas A&M University, 2019.

[17] A. Hyo-Sung, K. Young-Soo and C. YangQuan, "An Interval Kalman Filtering With Minimal Conservatism," *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9563-9570, 2012.

[18] L. Rall, "A Theory of Interval Iteration," *Proceedings of the American Mathematical Society*, pp. 625-631, December 1982.

[19] G. Mayer, "On the Asymptotic Convergence Factor of the Total Step Method in Interval Computation," *Linear Algebra and its Applications*, pp. 153-164, 1987.

[20] S. Kline and F. A. McClintock, "Describing Uncertainties in Single-Sample Experiments," *Mechanical Engineering*, vol. Vol. 75, pp. 3-8, 1953.

[21] B. R. B. Mohamed N. Nounou, "On-Line Multiscale Filtering of Random and Gross Errors Without Process Models," *AIChE Journal*, vol. 45, no. 5, pp. 1041-1058, 1999.

[22] T. T. Anh, F. L. Gall, C. Jauberthie and L. Travé-Massuyès, "Two Stochastic Filters and Their Interval Extensions," *IFAC-PapersOnLine*, vol. 49, pp. 49-54, 2016.

[23] L. Jaulin and E. Walter, "Set Inversion via Interval Analysis for Nonlinear Bounded-Error Estimation," *Automatica,* vol. 29, pp. 1053-1064, 7 1993.

[24] B. R. B. Mohamed N. Nounou, "On-Line Multiscale Filtering of Random and Gross Errors without Process Models," *AIChE Journal,* vol. 45, no. 5, pp. 1041-1058, 1999.

[25] G. Mayer, "On the Asymptotic Convergence Factor of the Total Step Method in Interval Computation," *Linear Algebra and its Applications,* pp. 153-164, 1987.

[26] B. Li, C. Li, J. Si and G. P. Abousleman, "Interval Least-Squares Filtering With Applications to Robust Video Target Tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, 2008.

APPENDIX A

A.1 The Issue With Lack of Additive Inverse When Performing Discrete MS Reconstruction

Consider the following signal consisting of 4 observations. $X_L^0 = [x_1, x_2, x_3, x_4]^T$. The signal will be decomposed using Discrete Wavelet Decomposition with a Haar wavelet. Therefore, the filter coefficients are: $h = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$ and $g = \left[-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$. The decomposition is shown in Figure A-1.

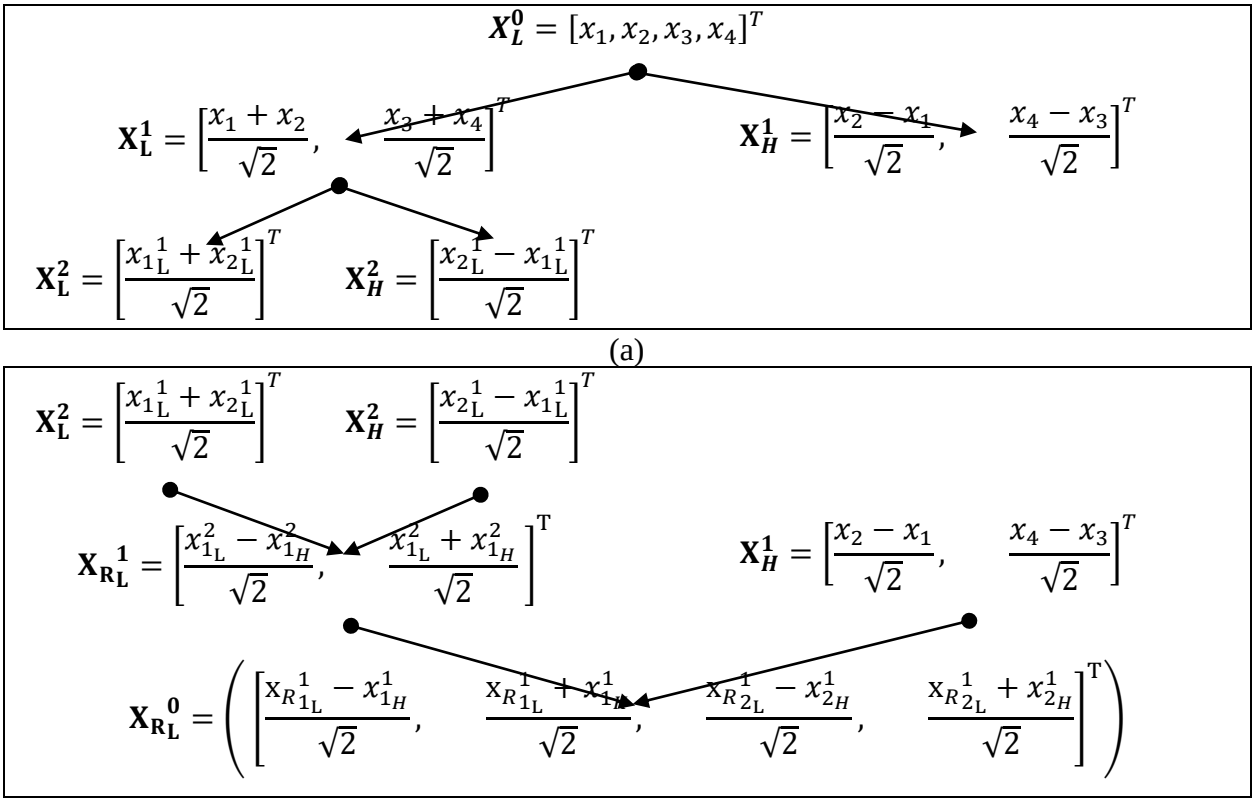


Figure A-1. Example of discrete wavelet decomposition and reconstruction.

Consider the first element in the signal reconstructed from level 2 to level 1, $\mathrm{x}_{R_1}{}^1_{\mathrm{L}} = \frac{2x_1{}^1_{\mathrm{L}} + (x_2{}^1_{\mathrm{L}} - x_2{}^1_{\mathrm{L}})}{2}$. With ordinary arithmetic, this would be $\mathrm{x}_{R_1}{}^1_{\mathrm{L}} = \frac{2x_1{}^1_{\mathrm{L}} + (0)}{2}$ resulting in $\mathrm{x}_{\mathrm{R}1}{}^1_{\mathrm{L}} = x_1{}^1_{\mathrm{L}}$, a perfect reconstruction of the level 1 approximate signal. But in the case of interval arithmetic $\left[\underline{\mathrm{x}_{R_1}}, \overline{\mathrm{x}_{R_1}}\right]^1_{\mathrm{L}} = \frac{2\left[\underline{x_1}, \overline{x_1}\right]^1_{\mathrm{L}} + \left(\left[\underline{x_2}, \overline{x_2}\right]^1_{\mathrm{L}} - \left[\underline{x_2}, \overline{x_2}\right]^1_{\mathrm{L}}\right)}{2}$ resulting in $\left[\underline{\mathrm{x}_{R_1}}, \overline{\mathrm{x}_{R_1}}\right]^1_{\mathrm{L}} = \left[\underline{x_1}, \overline{x_1}\right] + r\left(x_2{}^1_{\mathrm{L}}\right)[-1,1]$. The terms marked in red would have canceled, but it remains as a residue of the interval arithmetic subtraction operation.

Similarly, reconstruction from level 1 to level 0 should take us back to the original signal. Again, considering the first element in the reconstructed signal, $\mathrm{x}_{R_1}{}^0_{L} = \frac{\mathrm{x}_{R_1}{}^1_{L} - x_1{}^1_{H}}{\sqrt{2}}$ should yield a similar result as before. Consider the second term, $\mathrm{x}^0_{2\mathrm{L}} = \frac{\mathrm{x}_{R_1}{}^1_{\mathrm{L}} + x_1{}^1_{H}}{\sqrt{2}}$. Expanding $\mathrm{x}_{R_1}{}^1_{\mathrm{L}}$ and $x_1{}^1_{H}$ based on the original values, $\mathrm{x}^0_{2\mathrm{L}} = \frac{\frac{x_1 + x_2}{\sqrt{2}} + \frac{x_2 - x_1}{\sqrt{2}}}{\sqrt{2}}$. With ordinary arithmetic, this would be $\mathrm{x}^0_{2\mathrm{L}} = \frac{2x_2 + (x_1 - x_1)}{2}$. But for interval arithmetic, $\left[\underline{\mathrm{x}_{R_2}}, \overline{\mathrm{x}_{R_2}}\right]^0_{\mathrm{L}} = \left[\underline{x_2}, \overline{x_2}\right]^0_{\mathrm{L}} + r\left(x_1{}^1_{\mathrm{L}}\right)[-1,1]$**.** There is some residual left from the interval arithmetic subtraction operation. In a more general way, this can be formulated as

$$\left[\underline{x_{R_i}}, \overline{x_{R_i}}\right]^j_L = \left[\underline{x_i}, \overline{x_i}\right]^j_L + r\left(x^j_{k_L}\right)[-1,1], \qquad (1)$$

where k is $i + 1$ when i is odd, and $i - 1$ when i is even. Hence, the original signal cannot be recovered while doing IMSA with interval arithmetic.

To correct for this over-estimation during reconstruction, the radius is added to the reconstructed lower bound and subtracted from the reconstructed upper bounds. In a general way, the lower bound and the upper bound of the corrected reconstructed signal can be expressed as,

$$\underline{x_{R_i}}^{j}{}_{L} = \frac{1}{2}\left(\frac{\underline{x_i}^{j+1}{}_{L}}{\sqrt{2}} + \frac{\underline{x_i}^{j+1}{}_{H}}{\sqrt{2}}\right) + r\left(x_{k_L}^{j}\right), \qquad \overline{x_{R_i}^{j}} = \frac{1}{2}\left(\frac{\overline{x}_i^{j+1}{}_{L}}{\sqrt{2}} + \frac{\overline{x}_i^{j+1}{}_{H}}{\sqrt{2}}\right) - r\left(x_{k_L}^{j}\right),$$

where k is $i+1$ when i is odd, and $i-1$ when i is even.

A.2 Overestimation During Convolution of Intervals

To understand the issue of overestimation in intervals, consider the following example. Consider two interval data points, $\left[\underline{x_1}, \overline{x_1}\right]$ and $\left[\underline{x_2}, \overline{x_2}\right]$. Now consider the following linear operation on the intervals. $\left[\underline{y}, \bar{y}\right] = \alpha\left[\underline{x_1}, \overline{x_1}\right] + \beta\left[\underline{x_2}, \overline{x_2}\right] = \left[\alpha\underline{x_1} + \beta\underline{x_2}, \alpha\overline{x_1} + \beta\overline{x_2}\right]$. And the resulting interval $\left[\underline{y}, \bar{y}\right]$ will have a radius of $r\left(\left[\underline{y}, \bar{y}\right]\right) = \frac{\bar{y}-\underline{y}}{2}$ which comes out to be $\alpha r(x_1) + \beta r(x_2)$. The same result can be obtained when dealing with the interval as upper and lower bounds. For the Centers and Radii approach, the same result is obtained, since we will calculate the new radius as $r\left(\left[\underline{y}, \bar{y}\right]\right) = \alpha r(x_1) + \beta r(x_2)$.

The true interval radius can be better estimated with the principle of propagation of uncertainty [20]. Using the propagation of uncertainties, the expected radius of the result $\left[\underline{y}, \bar{y}\right]$, with the same confidence level as the input variables should be, $r(y) = \sqrt{\left(\frac{\partial y}{\partial x_1} r(x_1)\right)^2 + \left(\frac{\partial y}{\partial x_2} r(x_2)\right)^2}$. Therefore for $f(x_1, x_2)$, the expected radius would be $\sqrt{(\alpha\, r(x_1))^2 + (\beta\, r(x_2))^2}$. This expected radius will always be tighter and a better estimate than the one derived through interval arithmetic. That is

$$\sqrt{(\alpha\, r(x_1))^2 + (\beta\, r(x_2))^2} \leq \alpha(r(x_1)) + \beta(r(x_2)). \qquad (2)$$

In more general terms, the convolution with the filter coefficients, $X_L * g$ can be seen as a linear operation. The variance (or radius) in the result found from this operation is calculated using the propagation of uncertainty as $r(X_L * g) = \sqrt{\sum\left(\frac{\partial (X_L * g)}{\partial x_i} r(x_i)\right)^2}$

Therefore, during the decomposition stages, the radii are estimated $Q_g = \sqrt{\sum\left(\frac{\partial(X_L * g)}{\partial x_i} r(x_i)\right)^2}$ and $Q_h = \sqrt{\sum\left(\frac{\partial(X_L * h)}{\partial x_i} r(x_i)\right)^2}$

A.3 The Detailed Algorithm for Solving the Lack of Additive Inverse in Interval Arithmetic

1. Reconstruct the signal $\left[\underline{x_R}, \overline{x_R}\right]_{\mathrm{L}}^{j}$ from the approximate $\left[\underline{x}, \bar{x}\right]_{\mathrm{L}}^{j+1}$ and the detail $\left[\underline{x}, \bar{x}\right]_{\mathrm{H}}^{j+1}$.
2. Compute the radius of the previously decomposed approximate signal $x_i^{R\,j}{}_L = \frac{\overline{x_i^J{}_L} - \underline{x_i^{\mathrm{j}}{}_{\mathrm{L}}}}{2}$. If the signal being reconstructed is the original ($j = 0$), then the radius of the original signal is computed.
3. The radius found in step 2 is then added to the lower bound and subtracted from the upper bound calculated in step 1. i.e., $\underline{x_R}{}_L^{\,j} = \underline{x_R}{}_L^{\,j} + x_i^{R\,j}{}_L$, and $\overline{x_R}{}_L^{\,j} = \overline{x_R}{}_L^{\,j} - x_i^{R\,j}{}_L$

Note: The subscript $_R$ stands for reconstructed signal, and the superscript ^{R} stands for the radius of the interval data.

A.4 The Detailed Algorithm for Solving for the Effect of Convolution

1. Decompose the interval signal, $\left[\underline{x}, \bar{x}\right]_{\mathrm{L}}^{j-1}$ to the approximate $\left[\underline{x}, \bar{x}\right]_{\mathrm{L}}^{j}$ and detail $\left[\underline{x}, \bar{x}\right]_{\mathrm{H}}^{j}$.
2. Each interval point in the approximate signal is converted to its center and radius as $x_i^{C\,j}{}_L = \frac{\underline{x_i^{\mathrm{j}}{}_{\mathrm{L}}} + \overline{x_i^J{}_L}}{2}$, and $x_i^{R\,j}{}_L = \frac{\overline{x_i^J{}_L} - \underline{x_i^{\mathrm{j}}{}_{\mathrm{L}}}}{2}$ respectively.
3. The corrected estimate of the radius of the approximate signal is calculated as $x_i^{R\,j}{}_L = \sqrt{\left(\frac{1}{\sqrt{2}} x_i^{R\,j-1}{}_L\right)^2 + \left(\frac{1}{\sqrt{2}} x_{i+1}^{R\;\;j-1}{}_L\right)^2}$.
4. The radius in step 2 is replaced with the one found in step 3. The bounds are then recalculated as $\underline{x_i^{\mathrm{j}}{}_{\mathrm{L}}} = x_i^{C\,j}{}_L - x_i^{R\,j}{}_L$ and $\overline{x_i^J{}_L} = x_i^{C\,j}{}_L + x_i^{R\,j}{}_L$.

5. This procedure is then repeated for all interval points in the approximate signals at every stage of decomposition.
6. The process is similar for the detail signal. The corrected estimate of the radius in step 3 will be $x_{i\ H}^{R\ j} = \sqrt{\left(\frac{1}{\sqrt{2}} x_{i\ H}^{R\ j-1}\right)^2 + \left(\frac{1}{\sqrt{2}} x_{i+1\ H}^{R\ j-1}\right)^2}$.

For reconstruction, the process is the same. However, the corrected estimate of the radius is calculated in step 3 is $x_{i\ L}^{R\ j} = \sqrt{\left(\frac{1}{\sqrt{2}} x_{i\ L}^{R\ j+1}\right)^2 + \left(\frac{1}{\sqrt{2}} x_{i\ H}^{R\ j+1}\right)^2}$.