Exploiting Temporal Correlation of Sparse Signals in Wireless Sensor Networks

Ahmed S. Alwakeel*, Mohamed F. Abdelkader†, Karim G. Seddik‡ and Atef Ghuniem†
*Department of Communication and Electronics, Sinai University, Egypt
†Department of Electrical Engineering, Port Said University, Egypt
‡Electronics Engineering Department, American University in Cairo, Egypt
Email: en.aei@su.edu.eg, mdfarouk@eng.psu.edu.eg, kseddik@aucegypt.edu, atghuniem@yahoo.com

***Abstract*—Collecting data continuously in Wireless Sensor Networks (WSNs) with limited power and bandwidth is still a challenging issue. Recently, the sparse nature of these data motivated the use of Compressive Sensing (CS) as an efficient data gathering technique. In this paper, several algorithms are proposed to effectively exploit the temporal correlation and the sparsity inherent in sensor network data over time. These algorithms combine recent advances in compressive sensing (CS) theory, data compression, and data gathering algorithms. Experimental analysis through simulation evinces that the proposed algorithms significantly reduce the power consumption by reducing the number of sent measurements for the same Normalized Mean Square Error (NMSE).**

I. Introduction

Wireless Sensor Networks consist of a large number of sensors distributed in a sensing area to serve different tasks, such as continuous environmental monitoring [1], [2]. These networks are intended to continuously sense an area of interest and transmit the sensed data to a sink node. Due to the power consumption constraints, it is inefficient to directly transmit the raw sensed data to the sink, as they often exhibit a high correlation in the spatial and temporal domains and can be efficiently compressed to reduce power and bandwidth requirements.

Spatial correlation in WSN refers to the correlation between the sensed data at spatially adjacent sensor nodes. On the other hand, temporal correlation usually refers to the slow varying nature of the sensed data. Researchers were motivated to explore this nature in designing efficient data gathering approaches. In [1], the sensor sends its reading only if it is changed from the last time. Spatial correlation between sensors is also utilized by suppressing the reading of the edges between nodes if the difference between the nodes values is unchanged. Goel and Imielinski in [3] propose a prediction model at the sink to predict the current readings of the sensors based on their past and surrounding readings. These predicted values are sent back to the sensors. The sensor ceases its transmission if difference between its reading and predicted one does not exceed a specific threshold.

Compressive sensing (CS) [4], [5] is a novel tool that provides a mean to process and transport correlated data in an efficient manner by exploring the sparsity of these data. Recently, CS was demonstrated effective in exploring data correlation in WSN. In [6], the authors introduce a compressive data gathering approach for WSN based on CS; they focus on designing an efficient measurement matrices and comparing between them experimentally. Meanwhile, they introduce several ideas for exploiting temporal correlation, spatial correlation, and measurement vector reshape. Unfortunately, they did not provide an experimental evaluation of these approaches. We will revisit some of their ideas in later parts of this paper.

On the other hand, many novel CS algorithms were introduced to further exploit signal structure in order to improve performance over traditional CS algorithms. The weighted $\ell 1$ [7] improves CS reconstruction by incorporating the possible location of non-zero entries as a prior information. From another perspective, temporal correlation can be modelled in the form of a multiple measurement vector (MMV) as in [8], where it models the source as an auto regressive (AR) process and then incorporates such information into the framework of sparse Bayesian learning for sparse signal recovery and converts MMV to block single measurement vector (SMV) model.

In this paper, we investigate different approaches to utilize temporal correlation for efficient CS data gathering in WSN. Some of these approaches utilize temporal correlation to process the sensed data in a way that increases its sparsity order. Which in turn reduces the required number of measurements in the sensing process. Another class of approaches utilizes the temporal correlation as a prior in the reconstruction step for CS. The last class of approaches combines several time instants into a single measurement vector, on the assumption that this MMV will be more sparse than distinct time instants measurements. We experimentally evaluate the performance of these approaches under different noise levels using synthetic and real WSN data.

The rest of the paper is organized as follows. We provide an overview of the mathematical background of CS in Section II. In Section III, the system model is described. We explain different algorithms that exploit temporal correlation using different approaches in Section IV. In Section V, the experimental results are presented. We conclude with the conclusion and future work in Section VI.

II. A Mathematical Background on Compressive Sensing

Compressive sensing theory [4], [5] provides an elegant mathematical framework to compress and recover signals using a small number of linear measurements. It shows that under certain conditions on the measurement matrix, the acquired signal can be perfectly reconstructed from these measurements. Consider a real-valued signal $\boldsymbol{x}_{N\times 1}$, which is sparse in some domain $\boldsymbol{\Psi}$.

$$\boldsymbol{x} = \sum_{i=1}^{N} s_i \boldsymbol{\psi_i} \quad \text{or} \quad \boldsymbol{x} = \boldsymbol{\Psi s}. \tag{1}$$

The signal $\boldsymbol{x}$ is K-sparse if it can be represented as a linear combination of only K basis vectors; that is, only K of the $\boldsymbol{s_i}$ coefficients in equation (1) are non zero.

In CS, we do not acquire $\boldsymbol{x}$ directly but rather acquire $M < N$ linear measurements $\boldsymbol{y}$, using an $M \times N$ measurement matrix $\boldsymbol{\Phi}$, as shown in the following equation.

$$\boldsymbol{y} = \boldsymbol{\Phi x} = \boldsymbol{\Phi \Psi s} = \boldsymbol{\Theta s}. \tag{2}$$

If the measurement matrix $\boldsymbol{\Phi}$ satisfies a condition called the restricted isometric property condition (RIP) [4], then the signal reconstruction problem can be framed as an ℓ_1-norm minimization problem and perfect reconstruction is guaranteed.

$$\underset{\boldsymbol{s}\in\mathbb{R}^N}{min} ||\boldsymbol{s}||_{\ell 1} \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{\Phi \Psi s}. \tag{3}$$

III. System Model

We consider a WSN composed of N sensors, randomly disseminated to continuously monitor a given area. We denote the signal collected by sensors as $\boldsymbol{x_{N\times 1}}$. We assume this signal is sparse in some transform domain with a number of non zero elements $K << N$. Sensor reading data are forwarded to a central sink node.

We use the single hub compressive data gathering (CDG) model proposed in [9] as shown in Figure 1. We assume that each node knows its local routing structure. Initially, the sink sends an $M \times N$ measurement matrix to all nodes. Each node multiplies its reading with the corresponding column in the measurement matrix, adds it to the vector received from the previous node, and then sends the resulting $N \times 1$ vector to the next hop node. This vector is propagated through the network until it reaches the sink as shown in Figure 1. Thus, each node sends M messages and the overall complexity is $O(MN)$. The sink collects the M measurements and needs to reconstruct N readings.

The received signal at the sink is represented as

$$\boldsymbol{y_{M\times 1}} = \boldsymbol{\Phi_{M\times N}} \times \boldsymbol{x_{N\times 1}} + \boldsymbol{w_{M\times 1}} \tag{4}$$

where $\boldsymbol{y_{M\times 1}}$ is the measurement collected at the sink, $\boldsymbol{w_{M\times 1}}$ is the additive noise between the sensors and the sink, whose components are independent with zero mean and a variance of $\boldsymbol{\sigma^2}$, and $\boldsymbol{\Phi_{M\times N}}$ is a measurement matrix designed according to the data gathering algorithm used.

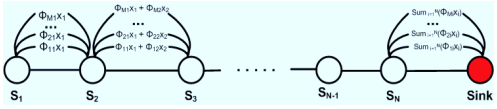


Fig. 1: Sub-tree of WSN.

Our goal is to reconstruct the original signal $\boldsymbol{x}_{N\times 1}$ from the compressed received signal at sink $\boldsymbol{y}_{M\times 1}$, where $K < M << N$. We note that all of the approaches in this paper can be applied to different CS data gathering approaches [10] with any different network organization.

IV. Different algorithms exploiting temporal correlation in WSNs

In this section, we discuss the different algorithms that are studied to exploit temporal correlation of sensor data. This work can be categorized into three main approaches:

1) An approach that increases the sparsity order by taking the difference between the current reading and the previous one. This can be performed either at the sensor (denoted by "CS-diff") [1], [11] or at the sink (denoted by "Measurement diff") [6].
2) An approach that increases the sparsity order by rearranging the sensor readings based on temporal correlation. Two algorithms are presented under this approach, denoted by "Reshape" [6] and "MMV" [8].
3) An approach that takes previous reading as a prior information and utilizes weighted CS [7] to enhance the reconstruction with fewer number of measurements (denoted by "Weighted") [12].

A. Calculate the difference at sensors: CS-diff

Sensor readings usually exhibit a slow varying pattern in time. In many scenarios, only a limited number of sensors change their values between two consecutive time instants. This was used in [1], where the sensors send their reading only if it has changed from the previous one. CS provides an efficient framework to benefit from this assumption, as the current reading and the previous reading would be highly correlated and the difference will be sparse in the wavelet domain and in various situations in the original time domain. The "CS-diff" algorithm, listed below, uses CS to compress and transmit the time difference signal instead of the original sensor readings. One benefit of this algorithm is its high noise immunity, but it needs more energy for extra-processing at the sensors. We note that the same approach was used in [11] for compressive background subtraction in images.

B. Calculate the difference at sink: Measurement diff

This algorithm is built on subtracting the current measurements from the previous one. The idea of this algorithm was proposed in [6] to explore what was denoted as "temporal-frequency sparsity". However, it was not part of the experimental results performed in that work. While based on the same idea of CS-diff, this algorithm reduces the computation load at the sensors by moving the difference computation to

Algorithm 1 CS-diff.

Input:

$\boldsymbol{x}_{t_1}$ (data at time t_1), $\boldsymbol{x}_{t_2}$ (data at time t_2), $\boldsymbol{\Phi}_{M\times N}$ (measurement matrix)

Steps:

At each sensor calculate the difference : $\boldsymbol{x}_{dif} = \boldsymbol{x}_{t_2} - \boldsymbol{x}_{t_1}$

Collect measurements at time t_2 : $\boldsymbol{y}_{t_2} = \boldsymbol{\Phi}_{t_2}\boldsymbol{x}_{dif} + \boldsymbol{w}_{t_2}$

Reconstruct the difference signal at sink :

$\underset{\boldsymbol{x}_{dif}\in\mathbb{R}^N}{min} ||\hat{\boldsymbol{x}}_{dif}||_{\ell 1}$ s.t. $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}_{dif}$

Output:

Recover the signal at sink : $\hat{\boldsymbol{x}}_{t_2} = \boldsymbol{x}_{t_1} + \hat{\boldsymbol{x}}_{dif}$

the sink node. However, this is at the expense of less noise immunity as we will show in the results section. We will list the algorithm here for completeness and refer the reader to [6] for more details.

Algorithm 2 Measurement diff.

Input:

$\boldsymbol{x}_{t_1}$ (data at time t_1), $\boldsymbol{x}_{t_2}$ (data at time t_2), $\boldsymbol{\Phi}_{M\times N}$ (measurement matrix)

Steps:

Collect measurements at time t_2 : $\boldsymbol{y}_{t_2} = \boldsymbol{\Phi}_{t_2}\boldsymbol{x}_{t_2} + \boldsymbol{w}_{t_2}$

At sink calculate the difference between measurements : $\boldsymbol{y}_{dif} = \boldsymbol{y}_{t_2} - \boldsymbol{y}_{t_1} = \boldsymbol{y}_{t_2} - \boldsymbol{\Phi}_{t_2}\boldsymbol{x}_{t_1} = \boldsymbol{\Phi}_{t_2}\boldsymbol{x}_{dif}$

Reconstruct the difference signal at sink :

$\underset{\boldsymbol{x}_{dif}\in\mathbb{R}^N}{min} ||\hat{\boldsymbol{x}}_{dif}||_{\ell 1}$ s.t. $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}_{dif}$

Output:

Recover the signal at sink : $\hat{\boldsymbol{x}}_{t_2} = \boldsymbol{x}_{t_1} + \hat{\boldsymbol{x}}_{dif}$

Two main drawbacks can be easily noticed in the CS-diff and Measurement diff approaches; first, they depend heavily on the rate of change of the data, as the sparsity order and hence the number of measurements needed depends on the sparsity of the difference signal. The second problem is the accumulation of error if the algorithm is reiterated for a long period of time. This can be readily fixed by periodically sending the original sensor reading to correct drift error.

C. Signal reshape using measurement matrix: Reshape

The idea of shuffling or reshaping the sensor readings to increase the sparsity was also originally proposed in [6], where the authors used the information from the last recovered signal to design a measurement matrix that can reshape the measurement vector into a more sparse vector in a desired domain.

A reshaping matrix $\boldsymbol{Q}_{N\times N}$ is designed to optimally rearrange the signal at time t. Under the assumption of a slowly varying signal, the same matrix will approximately rearrange the signal captured at time $t+1$, which in turn increases its sparsity order.

The reshape algorithm effectiveness depends mainly on how much the most sparsifying order is preserved across successive time instants. Also, the optimal order depends on the domain in which the signal is considered sparse.

D. Multiple measurement reshape: MMV

A completely different approach to explore temporal correlation is based on the stacking of several time measurements into a multiple measurement vector (MMV). We will use the formulation presented in [8] to transfer the MMV into a sparse single measurement vector using the Kronecker product. However, we will utilize the standard ℓ_1-norm minimization for reconstruction instead of the sparse Bayesian learning (SBL) approach proposed in that paper.

At each time t, we collect the measurement $\boldsymbol{y}_{M\times 1}$ using the relation

$$\boldsymbol{y}_{\boldsymbol{M}\times\boldsymbol{1}} = \boldsymbol{\Phi}_{\boldsymbol{M}\times\boldsymbol{N}} \times \boldsymbol{x}_{\boldsymbol{N}\times\boldsymbol{1}} + \boldsymbol{w}_{\boldsymbol{M}\times\boldsymbol{1}} \tag{5}$$

We arrange the measurements from L time instants as follows.

$$\boldsymbol{\Upsilon} = \boldsymbol{\Phi}\boldsymbol{d} + \boldsymbol{\upsilon} \tag{6}$$

where $\boldsymbol{\Upsilon} \in \mathbb{R}^{M\times L}$ is the measurement matrix consisting of L measurement vectors, $\boldsymbol{d} \in \mathbb{R}^{N\times L}$ is the source matrix with each row representing single source readings, and $\boldsymbol{\upsilon} \in \mathbb{R}^{M\times L}$ is a white Gaussian noise matrix with each column representing the noise at a different time instant.

The MMV can be transformed into SMV by letting $Y = \text{vec}(\boldsymbol{\Upsilon}^T) \in \mathbb{R}^{ML\times 1}$, $X = \text{vec}(\boldsymbol{d}^T) \in \mathbb{R}^{NL\times 1}$, $V = \text{vec}(\boldsymbol{\upsilon}^T) \in \mathbb{R}^{ML\times 1}$, and $\boldsymbol{D} = \boldsymbol{\Phi} \otimes \boldsymbol{I}_L$. Where $\otimes$ is the Kronecker product. Then equation 6 transforms to

$$\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X} + \boldsymbol{V}. \tag{7}$$

As proposed in [8], to improve the sparsity of this signal, we can rewrite it as $Y = [\phi_1 \otimes \boldsymbol{I}_L, \cdots, \phi_M \otimes \boldsymbol{I}_L] \left[x_1^T, \cdots, x_N^T\right]^T + V$, where ϕ_i is the i^{th} column in $\boldsymbol{\Phi}$ and $x_i \in \mathbb{R}^{L\times 1}$ is the i^{th} row in X.

Note that this algorithm does not reconstruct the signal at each time instant, but it collects signals at different time instants and reconstructs all of them at once. This gives some limitation of using this algorithm in real time sensing applications.

E. Exploit weight at reconstruction: Weighted

The idea of using weighted $\ell 1$ in CS reconstruction to incorporate prior information about the signal was first introduced in [7]. Weighted CS was used in WSN in [12] to detect certain events using the event signature in the transform domain to construct the weighting matrix. In this algorithm, we propose using the previous time reading as a prior information in designing the weight matrix. This is done by replacing the CS $\ell 1$ minimization problem in equation 3 by the following

$$\underset{\boldsymbol{s}\in\mathbb{R}^N}{min} ||\boldsymbol{W}\boldsymbol{s}||_{\ell 1} \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{s}, \tag{8}$$

where $\boldsymbol{W}$ is a diagonal matrix $\boldsymbol{W} = diag(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_N)$. At the first time instant, the values of the weight matrix are initialized with ones, then at any time $t+1$ the values are inversely proportional to the corresponding signal transform components at previous time t, which is given by

$$w_i^{t+1} = \frac{1}{|s_i^t| + \epsilon}, \tag{9}$$

where $\epsilon > 0$ is a parameter introduced to provide stability and to ensure that a zero-valued component in s_i^t does not completely prohibit a nonzero estimate at the next time instant. ϵ should be set slightly smaller than the minimum nonzero magnitudes of $\boldsymbol{s}$. The weights in equation 9 focus the energy in the reconstruction process to the same nonzero locations at the previous time instant. Generally speaking, large weights are used to discourage nonzero entries in the recovered signal, while small weights are utilized to encourage nonzero entries.

V. Simulation Results and Analysis

In order to compare the different proposed approaches, we run experimental analysis through simulation. The normalized mean square error (NMSE) between the original and reconstructed signal is used as a performance measure in all of these experiments. The compression ratio (C.R.) in our experiment refers to the ratio between the number of measurements and the total number of sensors.

We use two types of data sets in our simulation. The first one is a synthetic data set generated by "peaks" function in Matlab [13]; this data is sparse in the discrete cosine transform (DCT) domain. We dynamically change the data over time by changing a percentage β of the sensors reading at each new time instant. The value of the signal at these sensors is changed within α of previous time values according to the following equation

$$x_{i,t+1} = (1 \pm \alpha) x_{i,t}. \tag{10}$$

In all of our experiments we randomly chose $\alpha \in [0.05, 0.25]$, and $\beta = \{10, 30\}$. The signal dimension corresponding to the number of sensors is N=400. We average the results over 50 trials to avoid fluctuations.

The second data set used is the real data set obtained by a WSN deployed at Intel Labs Berkeley [14]. This data set contains temperature, humidity, light, and voltage value periodically collected every 31 seconds from 54 distributed sensor nodes between February 28th and April 5th, 2004. Figure 2 shows the distribution of sensors in Berkeley Lab and the sink node is near sensor number 54. We apply the CDG technique shown in Figure 1, where each sensor sends to the next hop sensor and so on until reaching to the sink. The temperature sensor readings at different time instants are shown in Figure 3.

In Figure 4, we individually evaluate the performance of each of the different algorithms at different time instants, SNRs, and compression ratios. This figure clearly shows the improvement in performance after first time instant. This improvement decreases as time progress due to error accumulation. The Reshape algorithm is the most affected algorithm

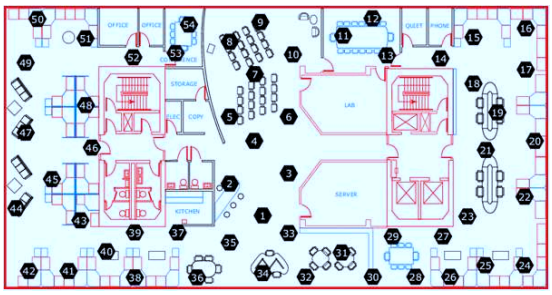


Fig. 2: Intel Labs Berkeley WSN [14].

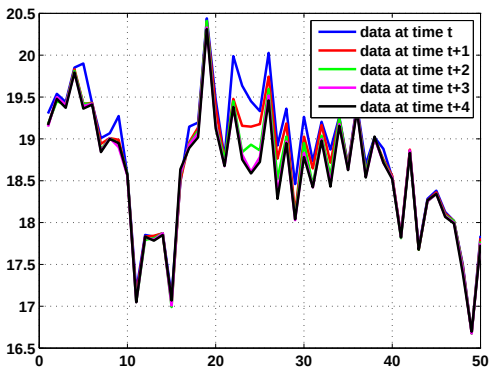


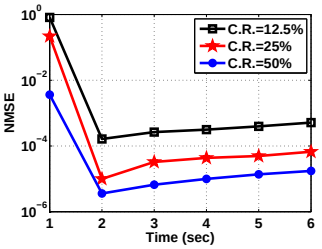
Fig. 3: Real data at different times.

with accumulator error. Also we need to note that in subfigure 4i, we do not plot NMSE against time as all previous sub-figures since the MMV algorithm collects the signals from different time instants and reconstructs all of them at once.

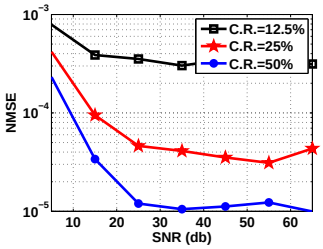
In Figure 5, we compare between the various approaches and the default CS approach in terms of the NMSE performance at different noise levels. The four subfigures show the effect of changing the time and the compression ratio on the different algorithms. It is clearly evident the superior performance of the CS-diff algorithm, especially at low SNRs. We also note from Figure 5d that by decreasing C.R. to 25 % the performance of the default and MMV algorithms are the most affected.

In Figure 6, we show the effect of changing the percentage of modified data at each time instant β on the different algorithms. As expected the "CS-diff" and "Measurement diff" algorithms are the most affected algorithms with changing β. The algorithms compress the difference signal which is sparse in time domain. This sparsity rate is highly affected by the amount of change in the data.

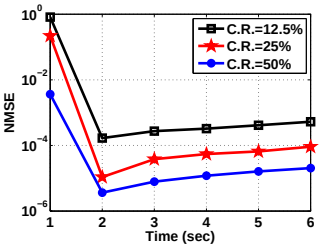
In Figure 7, the effect of the different algorithms in real data is shown. We note that all algorithms have a similar behavior to the peaks signal, but the behavior of the weighted algorithm is changed, because the sensors values have high deviation IV-E. The "CS-diff" algorithm outperforms all other algorithms at lower SNRs as it is considered the least effected algorithm with noise. However, at higher SNRs "Measurement diff" algorithm has better performance.



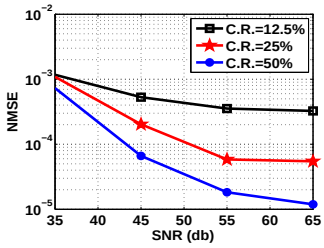
(a) CS-diff., SNR = 65 db



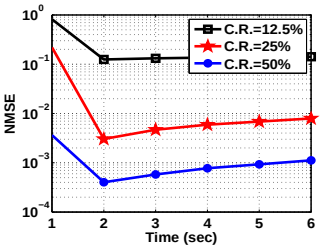
(b) CS-diff., $t = 3$ sec



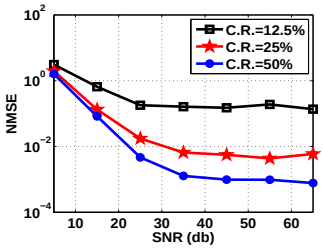
(c) Measurement diff., SNR = 65 db



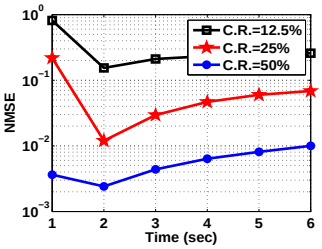
(d) Measurement diff., $t = 3$ sec



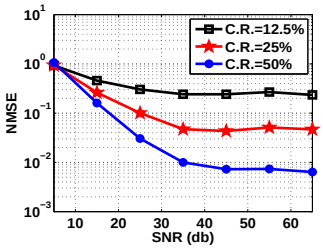
(e) Weighted, SNR = 65 db



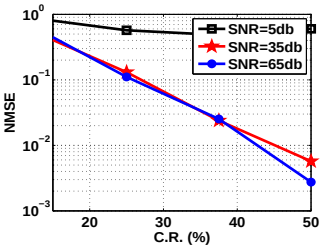
(f) Weighted, $t = 3$ sec



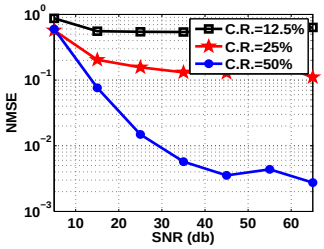
(g) Reshape, SNR = 65 db



(h) Reshape, $t = 3$ sec

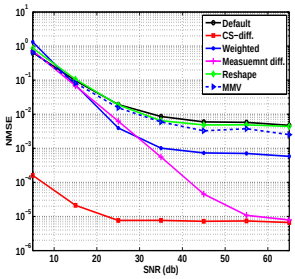


(i) MMV, $t = 3$ sec

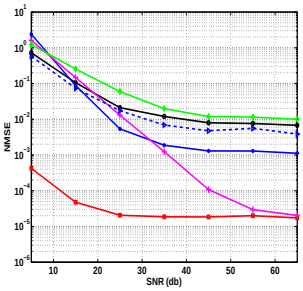


(j) MMV, $t = 3$ sec

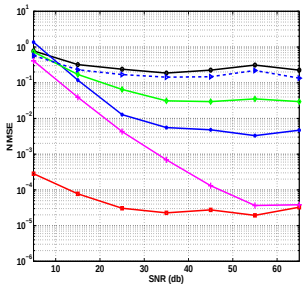
Fig. 4: Normalized mean square error Vs. time or C.R. or SNR for signal dimension N = 400 and β = 10 %.



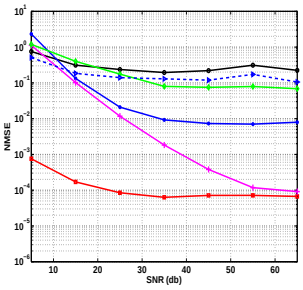
(a) $t = 2$ sec, C.R. = 50 %



(b) $t = 5$ sec, C.R. = 50 %

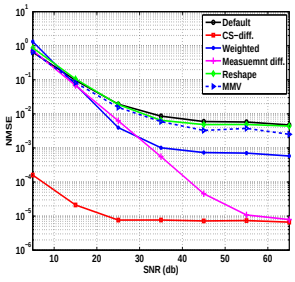


(c) $t = 2$ sec, C.R. = 25 %

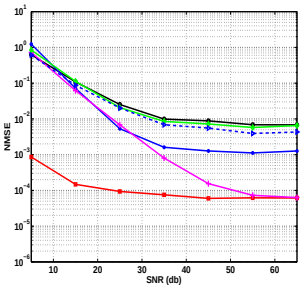


(d) $t = 5$ sec , C.R. = 25 %

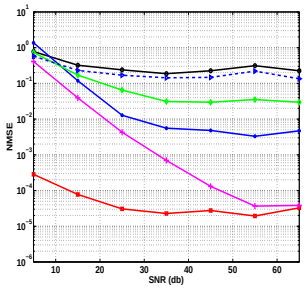
Fig. 5: Peaks signal, Normalized mean square error Vs. SNR for signal dimension N = 400 and and β = 10 %, for different times.



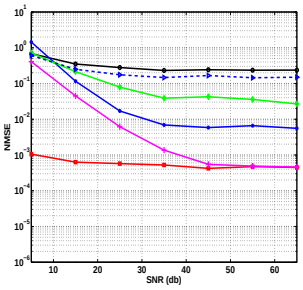
(a) β = 10% , C.R. = 50 %



(b) β = 30% , C.R. = 50 %



(c) β = 10% , C.R. = 25 %



(d) β = 30% , C.R. = 25 %

Fig. 6: Peaks signal, Normalized mean square error Vs. SNR for signal dimension N = 400 and time = 2 sec, for different β.

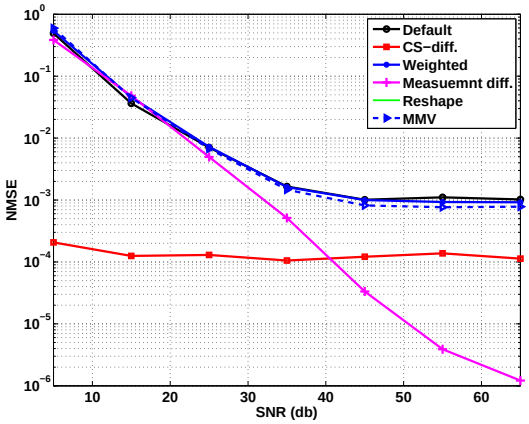


Fig. 7: Real data, Normalized mean square error Vs. SNR for signal dimension N = 50, for different times and C.R. = 50 % and time = 5 sec.

VI. Conclusion

In this work, we examine the behavior of the different algorithms that utilize temporal correlation to enhance compressive data gathering in WSN. The proposed algorithms cover different approaches to incorporate temporal correlation into measurement matrix design and reconstruction algorithms. We evaluated the performance of these approaches at different noise levels using synthetic and real data sets. In the future work of this research, we plan to design new approaches to further exploit the combined spatial and temporal correlation in WSN data. Other aspects of these approaches such as the computational complexity and latency will be further studied.

References

[1] A. Silberstein, R. Braynard, and J. Yang, "Constraint chaining: on energy-efficient continuous monitoring in sensor networks," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, 2006, pp. 157–168.

[2] X. Yang, K. G. Ong, W. R. Dreschel, K. Zeng, C. S. Mungle, and C. A. Grimes, "Design of a wireless sensor network for long-term, in-situ monitoring of an aqueous environment," *Sensors*, vol. 2, no. 11, pp. 455–472, 2002.

[3] S. Goel and T. Imielinski, "Prediction-based monitoring in sensor networks: taking lessons from mpeg," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 5, pp. 82–98, 2001.

[4] D. L. Donoho, "Compressed sensing," *IEEE Transactions on, Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[5] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE, Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[6] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Efficient measurement generation and pervasive sparsity for compressive data gathering," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 12, pp. 3728–3738, 2010.

[7] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l 1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[8] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 912–926, 2011.

[9] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," *ACM, Proceedings of the 15th annual international conference on Mobile computing and networking*, pp. 145–156, 2009.

[10] H. Zheng, S. Xiao, X. Wang, and X. Tian, "Energy and latency analysis for in-network computation with compressive sensing in wireless sensor networks," *INFOCOM*, pp. 2811–2815, 2012.

[11] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 155–168.

[12] Z. Charbiwala, Y. Kim, S. Zahedi, J. Friedman, and M. B. Srivastava, "Energy efficient sampling for event detection in wireless sensor networks," in *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*. ACM, 2009, pp. 419–424.

[13] L. Xiang, J. Luo, C. Deng, A. V. Vasilakos, and W. Lin, "Dual-level compressed aggregation: Recovering fields of physical quantities from incomplete sensory data," *arXiv preprint arXiv:1107.4873*, 2011.

[14] "Intel labs berkeley data,," *http://www.select.cs.cmu.edu/data/labapp3/*.