

# A TWO-STAGE DPCM SCHEME FOR WIRELESS SENSOR NETWORKS

Huiyu Luo, Yu-Ching Tong, and Gregory Pottie

University of California, Los Angeles

## ABSTRACT

We implement a two-stage DPCM coding scheme for wireless sensor networks. The scheme consists of temporal and spatial stages that compress data by making predictions based on samples from the past and helping sensors. It continuously monitors the additional gain provided by samples from other sensors, and therefore can be combined with data-centric routing algorithms for joint compression/routing optimization. Backward  $\epsilon$ -NLMS adaptation is used to better track changing environments and avoid coefficient transmissions. Several simulations are conducted to demonstrate the effectiveness of this coding scheme.

## 1. INTRODUCTION

Wireless sensor networks often operate under tight energy budgets. Communication power accounts for a substantial portion of sensor's energy consumption, and so the data rate should be aggressively reduced to achieve conservation [1].

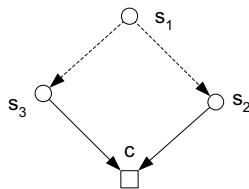


Fig. 1. A simple joint compression/routing problem.

Sensor networks differ from traditional communication networks in that data generated at different sensors, especially proximate ones, have high correlation since they are observations of closely related physical phenomena. A lot of research has been focusing on judiciously routing packets through sensors with highly correlated data such that the overall transmission is minimized [2][3]. As a simple illustration, consider Fig. 1 where three sensors transmit their observations to the fusion center. Sensor  $s_1$  may route its packets through  $s_2$  or  $s_3$ . The relay can read  $s_1$ 's packets to

further compress its own data. To determine the better routing strategy, we need to know how much *additional* rate reduction, which may vary with time,  $s_1$ 's data can produce for  $s_2$  and  $s_3$ . Most data-centric routing algorithms assume this rate-reduction information. Here, we discuss a two-stage DPCM scheme that processes first local side information, which is available without cost, then samples from other sensors. Additional coding gain provided by distant helping samples can be continuously monitored such that spatial side information is used only when the gain outweighs the cost. (This information is generally not available in traditional coding schemes.) Our method can be combined with data-centric routing strategies for use in joint compression/routing optimization.

We use closed loop backward adaptation, which does not require coefficient transmission, and tracks the changing statistics. In contrast, a forward adaptive scheme computes prediction coefficients from a block of samples in advance. It offers slightly higher gain for a slowly evolving field, but requires data buffering and additional bandwidth for coefficient transmission. Along with the adaptive prediction, adaptive quantization is used to make the most of the coding gain.

DPCM has been widely used in speech and video codings [4]. Two-stage DPCM schemes in speech coding base their predictions on previous samples and samples that are about one pitch period away. In video coding, the two-stage scheme is applied when both adjacent and inter-frame samples are used. However, in these methods, the distant side information is as readily available as adjacent samples. Hence, their coder design has more flexibilities.

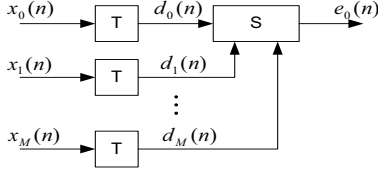
The paper is organized as follows. The two-stage DPCM scheme is presented in section 2. Simulation results are discussed in section 3. Section 4 concludes the paper.

## 2. TWO-STAGE ADAPTIVE DPCM SCHEME

### 2.1. Two-stage suboptimal approach

The two-stage suboptimal approach is described in Fig. 2. Sequences  $x_j(n)$ ,  $j = 0, 1, \dots, M$  are the measurements of sensor  $j$  at time  $n$ . Assuming that all the sequences have had their mean removed, a temporal DPCM stage is first

This material is based upon work supported by the National Science Foundation (NSF) under the Cooperative Agreement #CCR-0121778.



**Fig. 2.** The block diagram of a two-stage DPCM encoder

run at all sensors, then further compression is achieved using other sensors' temporal DPCM output as the side information. Define  $\tau_j$ ,  $j = 1, \dots, M$  as the delay at  $d_j(n)$  that yields the highest correlation with  $d_0(n)$ . It can be estimated using cross-correlation methods. Denote by  $a^*$  the complex conjugate of  $a$ , and  $A^H$  the complex conjugate and transpose of  $A$ . We have the following:

$$\begin{aligned} d_0(n) &= x_0(n) - \sum_{i=1}^N a_i^* \tilde{x}_0(n-i) \\ &= x_0(n) - \mathbf{w}_t^H \mathbf{y}_t(n) \end{aligned}$$

$$\begin{aligned} e_0(n) &= d_0(n) - \sum_{j=1}^M \sum_{k=-K}^K b_{j,k}^* \tilde{d}_j(n + \tau_j + k) \\ &= d_0(n) - \mathbf{w}_s^H \mathbf{y}_s(n) \end{aligned}$$

where

$$\mathbf{w}_t = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix}, \mathbf{y}_t(n) = \begin{bmatrix} \tilde{x}_0(n-1) \\ \vdots \\ \tilde{x}_0(n-N) \end{bmatrix},$$

$$\mathbf{w}_s = \begin{bmatrix} b_{1,K} \\ \vdots \\ b_{1,-K} \\ \vdots \\ b_{M,K} \\ \vdots \\ b_{M,-K} \end{bmatrix}, \mathbf{y}_s(n) = \begin{bmatrix} \tilde{d}_1(n + \tau_1 + K) \\ \vdots \\ \tilde{d}_1(n + \tau_1 - K) \\ \vdots \\ \tilde{d}_M(n + \tau_M + K) \\ \vdots \\ \tilde{d}_M(n + \tau_M - K) \end{bmatrix}$$

In a closed loop implementation,  $\tilde{x}_0(n)$  and  $\tilde{d}_j(n)$  are samples that are available at the decoder. Here, we assume they are the same as  $x_0(n)$  and  $d_j(n)$  with sufficient quantization bits. MMSE criteria on separate stages yield:

$$\mathbf{w}_t^{\text{opt}} = \mathbf{R}_{tt}^{-1} \mathbf{r}_t, \mathbf{w}_s^{\text{opt}} = \mathbf{R}_{ss}^{-1} \mathbf{r}_s,$$

and the minimum mean square error

$$J_{\min}^s = \sigma_x^2 - \mathbf{r}_t^H \mathbf{R}_{tt}^{-1} \mathbf{r}_t - \mathbf{r}_s^H \mathbf{R}_{ss}^{-1} \mathbf{r}_s.$$

in which

$$\mathbf{R}_{tt} = \text{E} \mathbf{y}_t \mathbf{y}_t^H, \mathbf{R}_{ss} = \text{E} \mathbf{y}_s \mathbf{y}_s^H, \mathbf{r}_t = \text{E} \mathbf{y}_t x_0^*, \mathbf{r}_s = \text{E} \mathbf{y}_s d_0^*$$

Define the overall and spatial coding gains

$$G = \text{E} x_0^2(n) / \text{E} e_0^2(n) = \sigma_x^2 / J_{\min}^s$$

$$G_s = \text{E} d_0^2(n) / \text{E} e_0^2(n) = (\sigma_x^2 - \mathbf{r}_t^H \mathbf{R}_{tt}^{-1} \mathbf{r}_t) / J_{\min}^s$$

In contrast, a one-stage scheme using the same set of side information would yield:

$$J_{\min} = \sigma_x^2 - \mathbf{r}_y^H \mathbf{R}_{yy}^{-1} \mathbf{r}_y$$

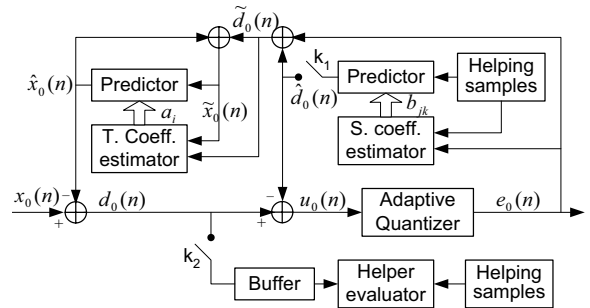
in which

$$\mathbf{R}_{ts} = \text{E} \mathbf{y}_t \mathbf{y}_s^H, \mathbf{R}_{yy} = \begin{bmatrix} \mathbf{R}_{tt} & \mathbf{R}_{ts} \\ \mathbf{R}_{ts}^H & \mathbf{R}_{ss} \end{bmatrix}, \mathbf{r}_y = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{r}_s + \mathbf{R}_{ts}^H \mathbf{w}_t^{\text{opt}} \end{bmatrix}$$

In general,  $J_{\min}^s > J_{\min}$ , but a two-stage implementation offers several other advantages over a one-stage approach. It improves stability. For highly correlated  $x_0(n)$  and  $x_j(n)$ , matrix  $\mathbf{R}_{yy}$  becomes near-singular. Separately designing temporal and spatial stages can help ensure that the temporal stage is minimum phase, thus stable. The spatial coding gain  $G_s$  sheds light on how much additional gain is provided by distant samples. At the spatial stage,  $d_j(n)$  instead of  $x_j(n)$  is used, so less decoding effort is required. In addition to compression, the temporal stage serves as a pre-whitening process, and the resulting  $\mathbf{R}_{ss}$  has better eigenvalue structures. This enhances the adaptive performances of the second stage [5].

## 2.2. $\epsilon$ -NLMS adaptation

The detailed block diagram of the closed loop DPCM encoder is given in Fig. 3. Switch  $k_1$  controls whether the spatial stage is used. The decoder has a similar structure, and hence is not shown here.



**Fig. 3.** The detailed block diagram of the encoder

The weight iteration uses  $\epsilon$ -NLMS with power update. The algorithm starts with  $\mathbf{w}_t(-1)$ ,  $p_t(-1)$ ,  $\mathbf{w}_s(-1)$ , and  $p_s(-1)$ , iterate for  $n = 0, 1, 2, \dots$

$$\hat{x}_0(n) = \mathbf{w}_t^H(n-1) \mathbf{y}_t(n), d_0(n) = x_0(n) - \hat{x}_0(n)$$

$$\begin{aligned}
\hat{d}_0(n) &= \mathbf{w}_s^H(n-1)\mathbf{y}_s(n), \quad u_0(n) = d_0(n) - \hat{d}_0(n) \\
e_0(n) &= \mathbf{Q}[u_0(n)] \\
\tilde{d}_0(n) &= \hat{d}_0(n) + e_0(n), \quad \tilde{x}_0(n) = \hat{x}_0(n) + \tilde{d}_0(n) \\
p_s(n) &= \beta_s p_s(n-1) + (1 - \beta_s) |\tilde{d}_0(n)|^2 \\
\mathbf{w}_s(n) &= \mathbf{w}_s(n-1) + \frac{\mu_s}{\epsilon_s + p_s(n)} e_0^*(n) \mathbf{y}_s(n) \\
p_t(n) &= \beta_t p_t(n-1) + (1 - \beta_t) |\tilde{x}_0(n)|^2 \\
\mathbf{w}_t(n) &= \mathbf{w}_t(n-1) + \frac{\mu_t}{\epsilon_t + p_t(n)} \tilde{d}_0^*(n) \mathbf{y}_t(n)
\end{aligned}$$

### 2.3. Helper evaluator

The helper evaluator, controlled by  $k_2$  has two functions: delay estimation and helper selection. Delay estimator determines the  $\tau_j$ ,  $j = 1, \dots, M$  resulting in the highest cross-correlation

$$\phi_{0j}(\tau_j) = \frac{\sum_{i=n}^{n+L-1-\tau_j} d_0(i) d_j^*(i + \tau_j)}{\sqrt{\sum_{i=n}^{n+L-1} |d_0(n)|^2 \sum_{i=n}^{n+L-1} |d_j(n)|^2}}$$

Directly computing  $\phi_{0j}(\tau_j)$  requires  $O(L)$  operations ( $L$  is the block size). The cost is reduced by using coarsely quantized samples [6]. The helper selector comes into play when a decision needs to be made on using which set of sensors' data as side information. The autocorrelation method [7] is applied to applicable sets of sensors and the one with the best gain/cost tradeoff is selected assuming the cost of distant side information is known. Note that the correlation matrix has Toeplitz-like structures, and efficient algorithms exist for solving such systems [8]. The overall cost of the evaluator can be kept within  $O(L)$ .

As the prediction error  $u_0(n)$  varies, an adaptive quantizer is essential to maximize the coding gain and limit quantization error. Readers are directed to [4] for more details.

## 3. SIMULATIONS

### 3.1. Autoregressive source

In the first simulation, we consider an autoregressive source observed by two sensors.

$$\begin{aligned}
s(n) &= s(n-1) - .5s(n-2) + z(n) \\
x_j(n) &= s(n) + u_j(n), \quad j = 1, 2
\end{aligned}$$

in which  $z(n)$  and  $u_j(n)$  are white Gaussian noise. Using one sensor's data as helping information, we plot the coding gains  $G$  and  $G_s$  against the variance ratio  $\sigma_z^2/\sigma_u^2$  for different schemes. 'osf' indicates one-stage forward DPCM, 'tsd' means two-stage  $\epsilon$ -NLMS with different step sizes on temporal and spatial stages, and 'tss' denotes two-stage  $\epsilon$ -NLMS with the same step sizes. (The spatial coding gain

of one-stage forward method is evaluated by comparing its result to the output of a single forward temporal stage.) It is observed that the spatial coding increases with the observation SNR, while temporal gain quickly saturates as it is circumscribed by the source statistics. In addition, Our ex-

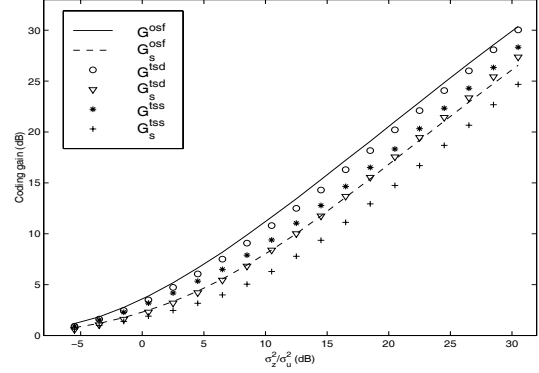


Fig. 4. Coding gain for an autoregressive source

perience reveals that appropriately choosing the relative step sizes of spatial and temporal stages  $\kappa = \mu_s/\mu_t$  can yield up to 2 dB gain improvement over simply setting  $\kappa = 1$ . This is explained as follows. Since the magnitude of  $e_0(n)$  is smaller than that of  $d_0(n)$ , using the same step sizes discourages the update of spatial weights  $\mathbf{w}_s$ . Choosing  $\kappa$  according to the magnitudes of  $e_0$  and  $d_0$  results in an adaptation that resembles the one-stage DPCM. It is cautioned, however, that setting  $\kappa$  too big undermines the temporal stage and tends to exaggerate the spatial coding gain.

### 3.2. Acoustic source

In the second simulation, we consider the acoustic data generated by a moving tank in a near field sensor array setup depicted in Fig. 5 [9]. 2000 samples are collected during

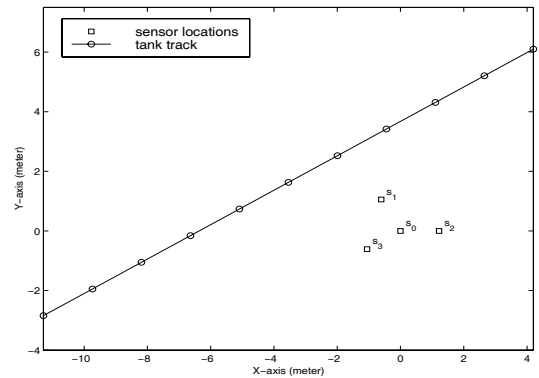


Fig. 5. Near field sensor array configuration

the period. Observations from sensor  $s_0$  are used as helping

**Table 1.** Coding gains (dB) of different schemes.

sensor	$G^{tsd}$	$G_s^{tsd}$	$G^{tss}$	$G_s^{tss}$	$G^{osf}$	$G_s^{osf}$
$s_1$	22.27	9.08	21.60	8.41	21.13	7.95
$s_2$	21.93	8.64	21.25	7.97	18.70	5.76
$s_3$	21.55	8.35	20.69	7.49	19.13	6.32

**Table 2.** Coding gains (dB) by different cities.

city	$G^{tsd}$	$G_s^{tsd}$
Macao	19.27	10.08
Datong	9.81	.09

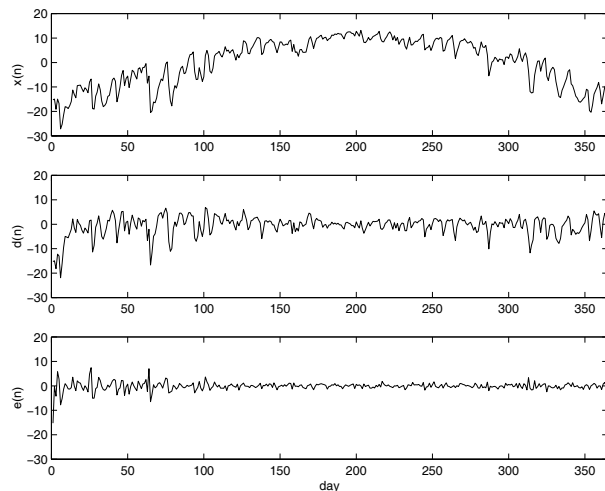
information to compress the data at sensor  $s_1$ ,  $s_2$ , and  $s_3$ . Since relative delay  $\tau_j$  varies when the tank passes by the array,  $\epsilon$ -NLMS adaptation performs better than the forward scheme that estimates the prediction coefficients for blocks of samples. This is displayed in Table 1, where we compare the coding gains (in dB) of  $\epsilon$ -NLMS adaptation and the forward scheme with block size 200. Notice ‘tsd’ has slightly higher gain than ‘tss’. We also observe that when the tank is closest to the sensor array, the forward method fares worst as  $\tau_j$  varies the most. On the other hand, the  $\epsilon$ -NLMS adaptation yields consistent results once it converges.

### 3.3. Weather data

So far, we have considered point sources. In the last simulation, we look at some correlated weather data obtained from NCDC [10]. We consider the daily mean temperature measured at three stations located at Hongkong, Macao, and Datong in 2003. We compress the data at Hongkong using those from Macao and Datong as side information. The coding gains (in dB) are given in Table 2. It shows that the spatial coding gain by Macao is way higher than that by Datong. This is expected because Hongkong and Macao are two cities near to one another, while Datong is at northern China, thousands miles away. In Fig. 6, we plot the input  $x(n)$  (with mean removed) and outputs of temporal and spatial stages at the encoder when samples from Macao are used as side information. The relative large error at the beginning is due to the initial weight convergence.

## 4. CONCLUSION

We discussed a two-stage DPCM scheme. Its ability to track the additional coding gain provided by distant side information makes it useful for joint compression/routing optimization in sensor networks. The  $\epsilon$ -NLMS adaptation reasonably adjusts to the changes on sample correlation. Simulations demonstrate that the algorithm provides results close to optima when the step sizes are appropriately set.

**Fig. 6.** Input and outputs of the encoder at Hongkong

## 5. REFERENCES

- [1] G. Pottie and W. Kaiser, “Wireless sensor networks,” *Communications of ACM*, vol. 43, no. 5, pp. 51-58, May 2000.
- [2] C. Intanagonwiwat et al. “Directed diffusion for wireless sensor networking,” *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp2-16, Feb 2003.
- [3] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, “On network correlated data gathering,” *Proc. IEEE Infocom*, Hongkong, March 7-11, 2004.
- [4] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice Hall, Englewood Cliffs, NJ, 1984.
- [5] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons Inc., Hoboken, NJ, 2003.
- [6] J. J. Dubnowski, R. W. Schafer, and L. R. Rabiner, “Real-time digital hardware pitch detector,” *IEEE Trans. ASSP*, vol. assp-24, no. 1, pp2-8, Feb 1976.
- [7] J. H. McClellan, “Parametric signal modeling,” chapter in *Advanced Topics in Signal Processing*, L. S. Lim and A. V. Oppenheim, eds., Prentice Hall, Englewood Cliffs, NJ, 1988.
- [8] A. H. Sayed and T. Kailath, “A look-ahead block schur algorithm for Toeplitz-like matrices,” *SIMAX*, vol. 16, no.2, pp. 388-413, Apr 1995.
- [9] K. Yao, Private communication
- [10] <http://www.ncdc.noaa.gov/oa/ncdc.html>, national climatic data center.