INFLATE: Incremental Wireless Transmission for Sensor Information in Industrial Environments

Roman Naumann, Stefan Dietzel, and Björn Scheuermann
Humboldt-Universität zu Berlin, Berlin, Germany
Email: roman.naumann@hu-berlin.de, stefan.dietzel@hu-berlin.de, scheuermann@informatik.hu-berlin.de

***Abstract*—As smart factory trends gain momentum, there is a growing need for robust information transmission protocols that make available sensor information gathered by individual machines and enable their algorithmic exploitation. Wireless transmission enables often-called-for flexibility, yet it poses challenges for reliable and timely transmission of information. This paper proposes a wireless transmission scheme for sensor information of production cycles in industrial environments. We include a preview functionality based on a discrete cosine transform that allows for rapid detection of problematic characteristics. The transmitted information's precision is improved using incremental updates as wireless capacity permits. Further, we include compact meta data that allows receivers to bound the received information's error. Evaluation results show that, even with high packet loss, characteristic features of sensor information are available early, and that error bounds closely follow the actual error.**

***Index Terms*—DCT; industrial wireless; sensor information; information dissemination; semantic compression**

I. Introduction

Current research in smart factories calls for quick adaptation of machine parameters, which is enabled by transmitting machine sensor information to centralized servers. Traditionally, manufacturing technology has been working mostly autonomously, limiting its potential use: operators have to manually set up parameters of production machines, which may be a complex task and is often based on personal experience or trial and error. Business logic cannot utilize the latest information about machine productivity.

Current smart factory concepts aim to tap into the amount of available information to optimize and flexibilize the manufacturing process. Collecting sensor information from machines in a centralized processing system allows to create statistical models that can be used for business analysis or prevention of future failures. Systems can issue warnings to operators or initiate machine shut-downs if sensor information suggests problems. In addition to increased information exchange, smart factories are likely to dynamically rearrange machines or components to make the production process more flexible and enable shorter production cycles [1]. In summary, two main communication protocol requirements arise: the amount of information that needs to be disseminated rises considerably, while at the same time, the positions where information originates will change more often due to machine rearrangements.

So far, wired Ethernet has been used to facilitate information exchange in factories [2]. Ethernet cables, however, are not usually available at the machines' locations [3], and laying new cables increases deployment cost and reduces flexibility. Wireless technology is a promising option to eliminate these problems. However, wireless transmission is inherently unreliable. This is particularly true for workshops, in which metal is the prevalent material [3], [4].

Together with partners from academia and manufacturing industry we work on improvements for injection-molding processes with smart factory technology in the context of an EU-funded project [5]. In this paper, we present a wireless transmission scheme for sensor information that is tailored to the communication requirements in manufacturing industries. Specifically, we target environments where machines transmit sensor and status information to a centralized component, hereafter called sink, using a single-hop wireless link. Our transmission scheme addresses scenarios where environmental constraints prevent immediate transmission of all available sensor information, yet applications require quick initial feedback about machine operation. For instance, a centralized monitoring component may want to initiate an emergency shutdown if machines operate outside their tolerable parameter range. Such applications may not require exact information immediately, but they need a reasonable estimate of the current sensor values including a dependable error bound at an early point in time. Still, exact sensor information is favorable on a longer time scale for failure prediction or process optimization, because even small differences may provide hints at future machine failure or sub-optimal configuration parameters.

Our transmission scheme implements a characteristic information representation that encodes gathered sensor information and arranges its transmission to implement these requirements. The characteristic representation allows for rapid decisions based on an early preview of sensor information from a production cycle. Subsequently, our scheme transmits updates of the acquired information, incrementally improving the delivered data's quality as the channel permits. Eventually, the characteristic representation equals the original data.

Our main contributions are: (1) the design and evaluation of an incremental transmission scheme for sensor information of production cycles, which is tailored to manufacturing industry requirements; (2) an error-bounding algorithm that provides close bounds on the characteristic representation and works

© 2015 IEEE 10.1109/ANTS.2015.7413631

well with practical packet loss; and (3) a comprehensive evaluation of both components based on network simulations and on sensor information traces from real factory deployments.

Section II reviews related work on sensor networks and signal compression techniques. Section III identifies unique requirements of manufacturing industries and introduces plastic industries as an example use case. The transmission scheme itself is described in Section IV and evaluated in Section V. Section VI concludes the paper.

II. Related Work

The information dissemination requirements in our use case relate to common use cases for wireless sensor networks (WSNs), which are ad-hoc networks composed of inexpensive, low-power sensor nodes. Common application scenarios include health monitoring, military, disaster recovery, and security [6]. Information dissemination is often organized in trees leading towards centralized sinks. As sensor networks may consist of large numbers of nodes, there is a large body of work on scalable information transmission using information compression and summarization techniques [7].

Research on compression for WSNs mostly optimizes for energy consumption and computational effort [7]. Marcelloni *et al.* [8], [9] propose lossless compression algorithms for WSNs. They make use of the high correlation between consecutive samples to reduce the range of values before using these deltas as input to an entropy encoder. Kolo *et al.* [10] improve upon this work by employing two entropy encoders and dynamically selecting the encoder which yields the best compression ratio. In comparison, the industrial setting of our protocol is not as restrictive as WSNs with regard to computational resources. A key difference to our work is that samples are compressed with knowledge of preceding samples only, whereas our transmission scheme uses the higher amount of available computational resources for more advanced processing on the full data from a production cycle. This is the foundation for our protocol's early approximate previews with error bounds. Another area of research in WSNs is lossy compression of sensor information, surveyed by Fasolo *et al.* [11], in which the sink reconstructs a signal that deviates from the original sensor samples. Such lossy compression mechanisms, however, assume that a number of sensor nodes observe spatially correlated data, such as the temperature distribution within a certain geographical area. In factories, information correlation is temporal, e. g., observed pressure during a manufacturing cycle, rather than spatial. Therefore, existing summarization mechanisms that aggregate information from multiple nodes are not applicable.

Willig *et al.* [12] and Christin *et al.* [13] provide an overview of industrial wireless sensor networks. The protocols and techniques described in their work are agnostic to the transmitted data. We, however, embrace the idea of an unreliable communication channel and use knowledge about the encoded sensor information for smarter retransmissions. Also, many contributions on industrial WSNs focus energy considerations. In the settings that we consider, source nodes are located at manufacturing machines and can be supplied with power there, so that energy is not an issue.

To implement our protocol's characteristic representation feature, we use Discrete Cosine Transform (DCT). The compression technique most similar to our work is the JPEG still picture compression standard [14]. After applying several transformations to linearize the two-dimensional input information, JPEG uses DCT to prioritize more important information that characterizes the input image's main features. JPEG decompression also supports incrementally increasing image quality, not unlike our Incremental Fault-Tolerant Transmission Scheme (INFLATE). Apart from the different domain, our protocol has two key advantages: first, it bounds the error of the signal's preview, i. e., the characteristic representation. While JPEG compression allows a preview of the image, it cannot quantify by how much the preview differs from the original. Second, INFLATE is more than a compression algorithm that could be used on top of a reliable transport. Rather, it embraces unreliable transport: INFLATE allows the sink to build a characteristic representation and bound its error even if network packets are lost during transmission.

III. System Model

Our transmission scheme is applicable to a wide range of industrial applications where rich sensor information is collected during production cycles and needs to be transmitted to a centralized system where it is evaluated and acted upon. As an example use case, we consider factories for plastic injection molding. Injection molding is a manufacturing process for producing parts by injecting plastic into special forms, called molds. The technique is also used for other materials, such as glass or metal.

Machine parameters, such as injection speed, mold temperature, or holding pressure, depend on the mold used and the exact type of plastic injected. Bad machine parameters result in an increased risk of producing defective parts. Setup of a mold in an injection-molding machine is usually done based on the operator's previous experiences and documentation. Especially first-time setup of molds is a trial-and-error process [15]. Recent results show that using artificial intelligence algorithms to derive optimized machine parameters from an archive of production data and sensor information can improve setup time and reduce defective parts [15]–[17]. On a shorter time scale, sensor information could facilitate detection of erroneous behavior, triggering operator alarms.

Wireless delivery of sensor information from injection-molding machines is challenging: workshops contain many metal objects, which may block line-of-sight signal propagation. Multi-path propagation can lead to high bit-error rates and an unreliable connection between sending and receiving nodes.

Temperature and pressure sensors at various places in and around the mold are particularly interesting for detecting production of waste parts and for deriving better machine parameters. This data is collected during a part's production cycle. Each production cycle takes between a few seconds and half a minute, depending on mold, material, and injection parameters. Between two production cycles, the machine mold cools down.

We will describe our protocol for a single sensor and a single production cycle. The sensor's signal is a sequence of samples $x = (x_1, x_2, \dots, x_N)$ from N time steps. Multiple production cycles simply result in our protocol being executed consecutively; multiple sensors can be supported by running multiple instances in parallel.

IV. INFLATE: Incremental Transmission Scheme

We design INFLATE, a transmission scheme that combines low-latency transmission of characteristic representations with eventual transmission of highly accurate information, while providing dependable error bounds at the receiver at any intermediate point in time. INFLATE consists of three main components: Section IV-A describes how collected sensor information is transformed using DCT, which puts most of the information of the signal into the lower frequencies's cosine coefficients. Early transmission of these coefficients provides the sink with a characteristic signal representation early on. In Section IV-B, we describe an algorithm that provides error bounds, which the sink can use to estimate current information representation precision. Finally, we design a retransmission mechanism that leverages DCT properties and is suitable for industrial environments in Section IV-C.

A. Information transformation and transmission

The Discrete Cosine Transform (DCT) and its inverse (IDCT) define mappings between a temporal sequence of N samples $x = (x_1, x_2, \dots, x_N)$ and a semantically equivalent set of N cosine coefficients $X = (X_1, X_2, \dots, X_N)$. Here, x is the data collected at the sensor node during the production cycle, which is to be transmitted to the sink. We define the DCT and IDCT analogously to [18]:

$$\operatorname{dct}(x)\colon \quad X_k = \sum_{n=1}^{N} x_n \cos\left(\frac{\pi(k-1)}{N}\Big(n - \frac{1}{2}\Big)\right), \tag{1}$$

$$\operatorname{idct}(X)\colon \; x_k = \frac{1}{2}X_1 + \sum_{n=1}^{N-1} X_{n+1} \cos\left(\frac{\pi n}{N}\Big(k - \frac{1}{2}\Big)\right). \tag{2}$$

The sensor node applies the DCT to the sensor information, which yields $X = \operatorname{dct}(x)$. It then packs the coefficients X into network packets. In INFLATE, each network packet contains a block of consecutively indexed coefficients.

By receiving such packets, the sink learns more and more about X as the transmission proceeds. Due to packet loss, the set of successfully received coefficient indices will likely not be consecutive at all times. As soon as all coefficients are available, perfect reconstruction of x is possible. If coefficients are missing, we can still construct a preview of x, though. This offers key advantages towards satisfying our requirements: in contrast to a loss of packets with time-domain sensor values, missing coefficients do not lead to gaps in the data; instead, it leads to reduced precision. Using the mechanism explained in Section IV-B, we can bound this preview's maximum deviation from the true signal.

Let $I = \{1, \dots, N\}$ be the set of all coefficient indices and $R \subseteq I$ the set of indices of those coefficients that are currently known by the sink. For constructing the preview $\widehat{x}_R$ based on this knowledge, the sink sets all unknown coefficients to zero. That is,

$$\widehat{x}_R = \operatorname{idct}\Big(\widehat{X}_R\Big), \quad \text{where } \widehat{X}_R = \Big(\widehat{X}_{R,1}, \dots, \widehat{X}_{R,N}\Big) \tag{3}$$
$$\text{with } \widehat{X}_{R,k} = \begin{cases} X_k & \text{if } k \in R, \\ 0 & \text{otherwise.} \end{cases}$$

Note that no additional transmissions are required for the preview functionality. Uncompressed coefficients have the same size as the original samples.

B. Error bounds

The provided preview's error decreases as more packets arrive at the sink. Full precision is achieved when all coefficients have arrived. Before that point, it is desirable that the sink can characterize the deviation between the current preview $\widehat{x}_R$ and the not-yet-known, correct signal x. Therefore, we implement a mechanism that provides an upper bound for this error.

We define the maximum relative preview error, hereafter simply called *error*, of a preview $\widehat{x}_R = (\widehat{x}_{R,1}, \dots, \widehat{x}_{R,N})$ as follows:

$$\operatorname{err}(x, \widehat{x}_R) = \max_{k=1}^{N} \left| \frac{x_k - \widehat{x}_{R,k}}{x_k} \right|. \tag{4}$$

With this definition, a preview sample $\widehat{x}_{R,k}$ with $\operatorname{err}(x, \widehat{x}_R) = p$ implies that

$$x_k \in \left[\frac{\widehat{x}_{R,k}}{1+p}, \frac{\widehat{x}_{R,k}}{1-p} \right]. \tag{5}$$

To estimate the current error at the sink, INFLATE uses a concept we term e-values. Intuitively, the e-value $e(x, M)$ characterizes the error if coefficients with indices in $M \subset I$ are still missing, i. e., when $R = I \setminus M$. More formally,

$$e(x, M) = \operatorname{err}\Big(x, \operatorname{idct}\big(\widehat{X}_{I \setminus M}\big)\Big). \tag{6}$$

Because e-value calculation requires knowledge of x, it can only be performed by the sensor node. Our goal is to allow the sink node to obtain knowledge about the e-value for the currently missing coefficients. We will also use e-values to guide INFLATE's reliability mechanism in order to preferentially (re)transmit those packets that result in the highest possible reduction of remaining errors at the sink. Transmitting e-values for all possible combinations of missing coefficients would imply exponential overhead and is thus not viable. We therefore transmit only specific e-values, selecting them in a way that still allows for good error estimates.

Assume the sensor node sends a packet containing a block of coefficients $X_j, \dots, X_k$. First, we include an e-value e_{rest} that characterizes the remaining error, assuming the sink has received all coefficients $X_1, \dots, X_k$ and none of the coefficients $X_{k+1}, \dots, X_N$:

$$e_{\text{rest}} = e\big(x, \{\, k+1, \dots, N \,\}\big). \tag{7}$$

In addition, we observe that transmission errors are likely to affect single packets. Consequently, we include the e-values associated with missing *only* the predecessor block or the successor block and no other coefficients; we call these e_{pred} and e_{succ}, respectively.

Using these three e-values, the sink can calculate error bounds for any transmission errors that affect at most two consecutive coefficient blocks. To this end, we use the following property, which we prove in the appendix.

Theorem 1 (Add-Property): Let K and L be sets of missing coefficient indices. Then

$$e(x, K) + e(x, L) \geq e(x, K \cup L). \tag{8}$$

That is, adding e-values for sets of missing coefficients yields an upper bound on the actual e-value for the sets' union. Applying this theorem, our protocol estimates the current error at the sink by adding known e-values associated with missing coefficients.

Consider the following example: a set of sensor values is fully described by coefficients $X_1, \dots, X_{10}$. The sensor node has transmitted the first seven coefficients, but the third coefficient's block was lost during transmission. The sink, hence, has knowledge of:

$$X_1, X_2, X_4, X_5, X_6, X_7.$$

As $e\big(x, \{\, 3 \,\}\big)$ was transmitted as e_{pred} in the packet carrying X_4 (and also as e_{succ} in X_2's packet), the sink calculates:

$$E = e\big(x, \{\, 3 \,\}\big) + e\big(x, \{\, 8, 9, 10 \,\}\big), \tag{9}$$

which is equal to or larger than $e\big(x, \{\, 3, 8, 9, 10 \,\}\big)$ by the above theorem, providing a valid upper bound on the error.

C. Smart retransmissions

So far, we have implemented a mechanism that transcodes time-varying sensor information using DCT to prioritize characteristic information during transmission, as well as a mechanism to bound its errors. While a key feature of our transmission scheme is its resilience to packet losses, it is desirable to retransmit missing coefficients to achieve eventual transmission of complete, precise information. Moreover, important coefficient blocks may be lost during transmission, making retransmissions a necessity.

We leverage our precision estimate mechanism to implement smarter retransmissions than those implemented by sequential protocols, such as TCP. In our protocol, the sensor node bases retransmission decisions on the estimated precision improvement at the sink using the e-values introduced in Section IV-B.

To implement retransmissions, the sensor node maintains a queue of all packets that have not yet been sent. The queue is ordered by descending precision improvement to ensure that the sink receives blocks with important coefficient first. To trigger retransmissions, the sink periodically sends negative acknowledgments. Having received a negative acknowledgment for a block, the sensor node re-adds it to its transmission queue, again ordering it using the associated precision improvement.

V. Evaluation

To evaluate INFLATE's transmission time, representation error, and our error bound's quality, we use the discrete-event network simulator NS-3 [19]. A single hop, from machine to sink, is simulated via YANS Wifi model [20] with IEEE 802.11g MAC and 2.4 GHz PHY.

We assume a blocked line of sight between machine and sink. The effects of multipath propagation and large-scale path loss are accounted for by Rayleigh and log-distance propagation loss models, one superimposed on the other, as Hashemi [21] suggests.

We use pre-recorded sensor information from real injection-molding machines. Each production cycle contains data from four temperature and pressure sensors at the machine and within the mold, each sampling at 500 Hz. A production cycle spans 25 s, resulting in 12 500 samples per sensor. Temperature and pressure information use Kelvin and bar (absolute), respectively, as units of measurement.

We repeat our simulations using samples from 165 production cycles. Moreover, we use 30 independent substreams of NS-3's MRG32k3a pseudo-random number generator [22]. We plot the arithmetic means of all simulation runs. In addition, we plot error bars, which indicate 95% confidence intervals.

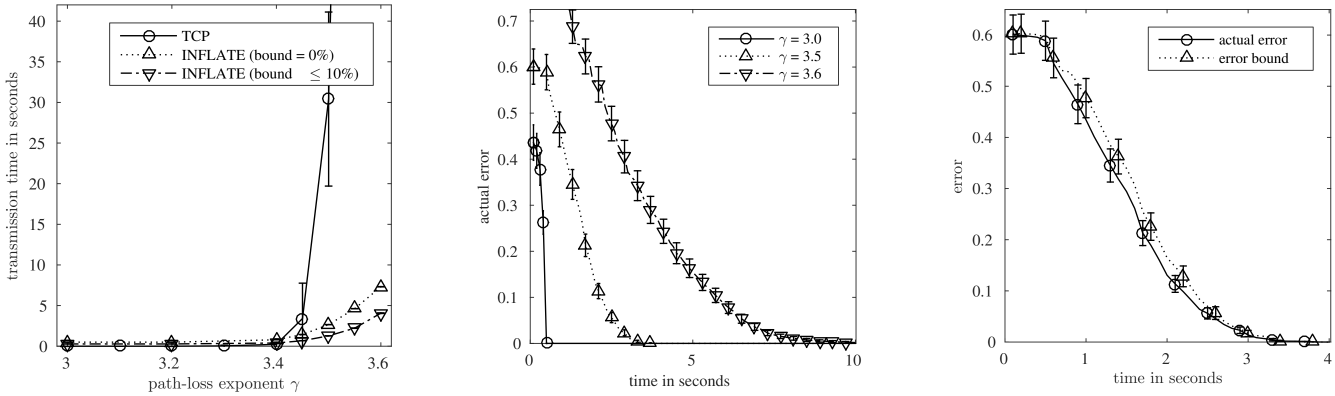
Interviews with industry partners identified a desirable distance of up to 30 meters between machine and sink, which we, therefore, use for our evaluation. To determine realistic packet loss configurations, we performed packet reception tests in a real-world plastic-industry workshop environment. Our experiments indicate packet loss levels of 50% and more, which is in line with previous results [3]. We vary the log-distance exponent γ in our simulation to match different reception conditions.

During simulation, a maximum of 104 coefficients are encoded in each network packet. This keeps the Internet Protocol datagram size below 576 bytes, which is in line with RFC 791 recommendations [23, Sec. 3.1] and findings on the relation between packet size and loss [24].

We evaluate our protocol's three main characteristics:

1) the transmission time for different packet loss settings compared to traditional TCP,
2) the actual representation error induced by INFLATE, and
3) the accuracy of our error bound.

Figure 1a shows the sensor information's delivery time for TCP and INFLATE at different path loss exponents (i. e., link qualities). For INFLATE, we plot both the transmission time for full transmission (0% error bound) and the time after which the error bound calculated at the sink has fallen below 10%. The y-axis shows the corresponding delivery times. TCP's delivery time degrades drastically with decreasing connection



(a) Transmission time of TCP versus INFLATE.

(b) Actual error for connection quality.

(c) Error bound ($\gamma = 3.5$).

Fig. 1: Simulation results.

quality, which we attribute to TCP congestion control. For $\gamma < 3.4$ (almost no packet loss), TCP outperforms INFLATE with only $20\,\mathrm{ms}$ versus $500\,\mathrm{ms}$, since TCP can already begin transmission during the production cycle. Such conditions seldom occur in practice, though. INFLATE, on the other hand, requires the sensor information from the whole production cycle before it can apply the DCT, and it only transmits in bursts *after* the production cycle, leaving the medium available most of the time. As soon as the links become more challenging ($\gamma \geq 3.5$, packet loss 50–60%), TCP performance degrades significantly: transmissions took an average of $30\,\mathrm{s}$ seconds. Even for $\gamma \geq 3.6$ (packet loss $\geq 80\%$)—where TCP could not transmit the production cycle within 15 minutes for half of the simulation runs and took, on average, $11\,\mathrm{min}$ for the other half—INFLATE's mean transmission time is only $7.3\,\mathrm{s}$. Moreover, INFLATE's transmission time consistently halves when we consider a tolerable error of 10% instead of 0%. That is, INFLATE provides a good estimate much earlier in practical scenarios.

Figure 1b shows how the actual representation error at the sink behaves over time for three different loss coefficients. As can be seen, the error reduces quickly when the most important coefficients have arrived. In particular, $\gamma = 3.0$ and $\gamma = 3.5$ result in the actual error quickly converging to zero. Even for $\gamma > 3.6$, where we observe packet loss rates of 80% and more, the error converges to zero within $10\,\mathrm{s}$.

An important feature of INFLATE is the characteristic representation, including an error bound. Therefore, we compare our transmission scheme's estimated error bound—which can be calculated at the sink—with the actual representation error. Figure 1c shows the actual error and the error bound observed during our simulations for $\gamma = 3.5$ (packet loss 50–60%). We observe that the mean difference between error bound and actual error during transmission is only 1.7%.

Differences between error bound and actual error are similar for other γ values. For $\gamma = 3.0$, we observed a mean deviation of zero; for $\gamma = 3.6$ (packet-loss over 80%) the error bound deviates, on average, by only 6.2% from the actual error, once it is available.

Summarizing, INFLATE fulfills the requirements we identified in Section III. Transmission time is significantly shorter than that of TCP, especially when considering applications that tolerate certain error bounds. Due to the DCT, the sensor information representation's error bound improves quickly with increasing transmission time, allowing for quick responses based on characteristic features. Finally, the transmitted error bound closely resembles the actual error.

VI. Conclusion and Future Work

We introduced a transmission scheme that is tailored for industrial applications where sensor information is collected in production cycles. INFLATE provides a characteristic representation of production data based on its Discrete Cosine Transform (DCT). Simulation results show that the transmission scheme quickly converges to a highly precise characteristic representation of transmitted sensor information. The metadata to bound the error at the sink is sufficient even with high rates of packet loss. Also, the error bound is close to the actual error, which is crucial to allow for qualified decision making at the sink.

Our comparison of INFLATE to TCP shows that, with greater packet loss, TCP performance degrades significantly, whereas INFLATE is much more resilient to packet loss. This is an important property, because wireless connectivity in industrial workshops is often problematic.

In future work, we plan to perform lossless compression on the coefficients. Such compression could reduce the transmission time for both the characteristic representation and the full signal. Another possible extension is multi-hop routing, which is commonly found in sensor networks.

Acknowledgments

We wish to thank Francesc Bonada for kindly supplying and helping us with data from the injection-molding process.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 636892 in the context of PREVIEW [5].

References

[1] D. Lucke, C. Constantinescu, and E. Westkämper, "Smart factory - a step towards the next generation of manufacturing," in *Manufacturing Systems and Technologies for the New Frontier*, M. Mitsuishi, K. Ueda, and F. Kimura, Eds., Springer London, 2008.

[2] G. Prytz, "A performance analysis of EtherCAT and PROFINET IRT," in *ETFA '08*, IEEE, Sep. 2008.

[3] B. Milic, S. Brack, and R. Naumann, "Hierarchical configuration system for wireless mesh networks in manufacturing industry," in *WMNC '14*, IEEE, 2014.

[4] E. Tanghe, W. Joseph, L. Verloock, *et al.*, "The industrial indoor channel: large-scale and temporal fading at 900, 2400, and 5200 MHz," *IEEE Trans. Wirel. Commun.*, vol. 7, no. 7, Jul. 2008.

[5] *Preview Project EU - Predictive System for Injection Mould Process Optimisation*, http://www.preview-project.eu/.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, *et al.*, "Wireless sensor networks: a survey," *Comput. Netw.*, vol. 38, no. 4, 2002.

[7] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *ITCC '05*, vol. 2, IEEE, 2005.

[8] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *IEEE Commun. Lett.*, vol. 12, no. 6, Jun. 2008.

[9] F. Marcelloni and M. Vecchio, "An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks," *Comput. J.*, 2009.

[10] J. G. Kolo, S. A. Shanmugam, D. W. G. Lim, *et al.*, "An adaptive lossless data compression scheme for wireless sensor networks," *J. Sens.*, vol. 2012, 2012.

[11] E. Fasolo, M. Rossi, J. Widmer, *et al.*, "In-network aggregation techniques for wireless sensor networks: a survey," *IEEE Wirel. Commun.*, vol. 14, no. 2, Apr. 2007.

[12] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," *Proc. IEEE*, vol. 93, no. 6, Jun. 2005.

[13] D. Christin, P. S. Mogre, and M. Hollick, "Survey on wireless sensor network technologies for industrial automation: the security and quality of service perspectives," *Future Internet*, vol. 2, no. 2, Apr. 8, 2010.

[14] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, 1991.

[15] C. Shen, L. Wang, and Q. Li, "Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method," *J. of Materials Process. Technology*, vol. 183, no. 2–3, Mar. 23, 2007.

[16] B. Ozcelik and T. Erzurumlu, "Comparison of the warpage optimization in the plastic injection molding using ANOVA, neural network model and genetic algorithm," *J. of Materials Process. Technology*, vol. 171, no. 3, Feb. 1, 2006.

[17] A. Tellaeche and R. Arana, "Rapid data acquisition system for complex algorithm testing in plastic molding industry," in *WASET '13*, 2013.

[18] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *Comput. IEEE Trans. On*, vol. 100, no. 1, 1974.

[19] T. R. Henderson, M. Lacage, G. F. Riley, *et al.*, "Network simulations with the ns-3 simulator," *SIGCOMM Demonstr.*, vol. 14, 2008.

[20] M. Lacage and T. R. Henderson, "Yet another network simulator," in *WNS2 '06*, ACM, 2006.

[21] H. Hashemi, "The indoor radio propagation channel," *Proc. IEEE*, vol. 81, no. 7, 1993.

[22] P. L'Ecuyer, R. Simard, E. J. Chen, *et al.*, "An object-oriented random-number package with many long streams and substreams," *Oper. Res.*, vol. 50, no. 6, Dec. 2002.

[23] J. Postel, "Internet protocol," RFC, no. 791, Sep. 1981.

[24] J. Korhonen and Y. Wang, "Effect of packet size on loss rate and delay in wireless links," in *WCNC '05*, vol. 3, IEEE, 2005.

Appendix

We first show a Lemma based on transformation matrices, which we subsequently use to prove Theorem 1. In the following, an orthogonal transformation matrix A of size $N \times N$ is used for DCT and IDCT, which can be derived by scaling. $X = Ax$ corresponds to the DCT and $x = A^T X$ to the IDCT, where x and X are matrices of size $N \times 1$.

Lemma 1: For a signal x with N samples and coefficients $X = Ax$, let R, M, $\widehat{X}_R$ and $\widehat{x}_R$ be defined as in Section IV-A, the following equation holds:

$$\left|\frac{x_m - \widehat{x}_{R,m}}{x_m}\right| = \left|\frac{1}{x_m}\sum_{i \in M} A^T_{mi} X_i\right| \quad \forall m \in \{\, 1, \ldots, N \,\}\,. \tag{10}$$

Proof:

$$\widehat{x}_R = A^T \widehat{X}_R = A^T (I_N \widehat{X}_R) \tag{11}$$

Split the identity matrix I_N into N sparse matrices Z^k of size $N \times N$, their elements defined by

$$Z^k_{ij} = \begin{cases} 1 & \text{if } i = j = k \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Since $I_N = \sum_{k=1}^{N} Z^k$, we can substitute:

$$\begin{aligned} A^T(I_N \widehat{X}_R) &= A^T(Z^1 \widehat{X}_R + Z^2 \widehat{X}_R + \cdots + Z^N \widehat{X}_R) \\ &= A^T(Z^1 \widehat{X}_R) + A^T(Z^2 \widehat{X}_R) + \cdots + A^T(Z^N \widehat{X}_R). \end{aligned} \tag{13}$$

Considering that

$$x = A^T(Z^1 X) + A^T(Z^2 X) + \cdots + A^T(Z^N X) \tag{14}$$

and that $Z^i \widehat{X}_R$ only contains zeroes if $i \in M$, but $Z^i \widehat{X}_R = Z^i X$ if $i \notin M$, we can instead write:

$$\widehat{x}_R = x - A^T \left(Z^{i_1} X\right) - \left(Z^{i_2} X\right) - \cdots - \left(Z^{i_{|M|}} X\right) \tag{15}$$

$$= x - \sum_{i \in M} A^T \left(Z^i X\right) = x - \sum_{i \in M} \begin{vmatrix} A^T_{1i} X_i \\ \vdots \\ A^T_{Ni} X_i \end{vmatrix} \tag{16}$$

Applying index m ($\forall m \in \{\, 1, \cdots, N \,\}$) yields

$$\widehat{x}_{R,m} = x_m - \sum_{i \in M} A^T_{mi} X_i, \tag{17}$$

which can be used at the left-hand side of the lemma. ■

Theorem 1 (Add-Property): Let K and L be sets of missing coefficient indices. Then

$$e(x, K) + e(x, L) \geq e(x, K \cup L). \tag{8}$$

Proof: Using Lemma 1, $e(x, K) + e(x, L)$ equals:

$$\max_{m=1}^{N} \left|\frac{1}{x_m}\sum_{k \in K} A^T_{mk} \cdot X_k\right| + \max_{m=1}^{N} \left|\frac{1}{x_m}\sum_{l \in L} A^T_{ml} \cdot X_l\right| \tag{18}$$

$$\geq \max_{m=1}^{N} \left(\left|\frac{1}{x_m}\sum_{k \in K} A^T_{mk} \cdot X_k\right| + \left|\frac{1}{x_m}\sum_{l \in L} A^T_{ml} \cdot X_l\right|\right) \tag{19}$$

$$\geq \max_{m=1}^{N} \left|\frac{1}{x_m}\sum_{k \in K} A^T_{mk} \cdot X_k + \frac{1}{x_m}\sum_{l \in L} A^T_{ml} \cdot X_l\right| \tag{20}$$

$$\geq \max_{m=1}^{N} \left|\frac{1}{x_m}\sum_{i \in K \cup L} A^T_{mi} \cdot X_i\right| = e(x, K \cup L), \tag{21}$$

where the triangle inequality $|x| + |y| \geq |x + y|$ is used on every element between (19) and (20).

■