# RELIABLE COMPUTATION OF NOMOGRAPHIC FUNCTIONS OVER GAUSSIAN MULTIPLE-ACCESS CHANNELS

*Mario Goldenbaum*[⋆‡]*, Holger Boche*[‡] *and Sławomir Stańczak*[⋆]

[⋆] Fachgebiet Informationstheorie und theoretische Informationstechnik,
Technische Universität Berlin, Germany

[‡] Lehrstuhl für Theoretische Informationstechnik,
Technische Universität München, Germany

## ABSTRACT

In this paper, a wireless sensor network is considered in which the objective is not to communicate individual sensor readings over a Gaussian multiple-access channel to a fusion center but rather to reliably compute some nomographic function thereof. Nomographic functions are exactly those multivariate functions that can be represented as a post-processed sum of pre-processed sensor readings. This special structure permits the utilization of the interference property of the Gaussian multiple-access channel for computing some nomographic functions at significantly higher rates than those achievable with traditional schemes. In this paper, a corresponding coding scheme is presented that protects the sum of pre-processed sensor readings against the channel noise by letting each node use the same nested lattice code.

***Index Terms*—** Distributed computation, nomographic functions, nested lattice codes, multiple-access channel, sensor networks

## 1. INTRODUCTION

Many wireless sensor network applications require the reliable computation of a pre-defined function of the measurements at a fusion center (e.g., arithmetic mean, maximum value) [1]. To solve the computation problem at hand, the transmissions of nodes are typically coordinated so that the fusion center receives interference-free transmit signals, which facilitates the reconstruction of each individual sensor reading. Once the fusion center is aware of all sensor readings, it subsequently computes the function of interest.

In [2] it is shown that this approach can be highly inefficient when the function to be computed at the fusion center is *linear*. In fact, the interference caused by concurrently transmitting nodes can be harnessed to compute function values at significantly higher rates than those achievable with any interference avoiding strategy (i.e., strategies recovering all the sensor readings at the fusion center).

Exploiting interference for efficiently computing *nonlinear* functions of the measurements is considered in [3]. The key idea is to apply a pre-processing function to each sensor reading prior to transmission and a post-processing function to the signal received by the fusion center (i.e., the sum of the individual transmit signals). Essentially, this allows an efficient computation of all *nomographic functions* over the channel, which are functions having a representation (2) [4].

M. Goldenbaum and S. Stańczak were supported in part by the German Research Foundation (DFG) under grant STA 864/3-2 and H. Boche by a start-up fund of Technische Universität München.

In contrast to the analog computation scheme proposed in [3], we present in this paper a simple digital one that extends the ideas of [5] to the computation of some *nonlinear* nomographic functions over a Gaussian multiple-access channel at a computation rate that is not achievable with conventional methods. The idea is as follows: each node first quantizes real-valued pre-processed sensor readings and then employs a nested lattice code from [6], [5] to protect the sum of messages against channel noise. Decoding the sum and applying the corresponding post-processing function provides a reliable estimate of the sought nomographic function value.[1]

## 2. SYSTEM MODEL AND PROBLEM STATEMENT

Consider a wireless sensor network consisting of $K \in \mathbb{N}$ nodes that are spatially distributed around a Fusion Center (FC). The nodes jointly observe a certain physical phenomenon resulting in real-valued sensor readings $s_i \in \mathbb{S}$, $i = 1, \ldots, K$, where $\mathbb{S} \subset \mathbb{R}$ denotes some compact measurement space.[2]

The objective of the network is not to communicate the individual $s_i$ to the FC but rather to reliably compute some pre-defined function $f \in \mathsf{F}(\mathbb{S}^K)$ thereof, called *desired function*. In doing so, all nodes share a synchronous Gaussian Multiple-Access Channel (MAC) where the channel output at the $t^{\text{th}}$ channel use has the form

$$Y[t] = \sum\nolimits_{i=1}^{K} X_i[t] + Z[t] \ , \ t \in \mathbb{Z} \ , \tag{1}$$

with $X_i[t] \in \mathbb{R}$ the corresponding input symbol of node $i$ and $Z[t] \sim \mathcal{N}(0, N)$, $N > 0$, independent and identically distributed (over channel uses) white Gaussian noise [7].

According to (1) is *summation* the natural mathematical operation of a Gaussian MAC, which can be beneficially exploited to compute linear functions in wireless networks much more efficiently than with interference avoiding strategies [2]. In [4] it is shown that essentially even every $f \in \mathsf{F}(\mathbb{S}^K)$ is computable over a Gaussian MAC since *every* real-valued function of $K$ variables has a *nomographic* representation.

**Definition 1.** Let $\mathbb{A}$ be any metric space and $K \geq 2$. Then, a function $f : \mathbb{A}^K \to \mathbb{R}$ for which there exist functions $\{\varphi_i \in \mathsf{F}(\mathbb{A})\}_{i=1}^{K}$

[1] *Notation:* The natural, integer and real numbers are denoted by $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{R}$. Let $\mathbb{A}^\ell$ be some topological space, then $\mathsf{C}^0(\mathbb{A}^\ell)$ denotes the space of real-valued continuous functions with domain $\mathbb{A}^\ell$. In contrast, $\mathsf{F}(\mathbb{A}^\ell)$ denotes the space of *every* function $f : \mathbb{A}^\ell \to \mathbb{R}$. The volume of a closed subset $\mathbb{D}$ of $\mathbb{R}^\ell$ is described by $\mathrm{Vol}(\mathbb{D})$.

[2] Assuming $\mathbb{S}$ to be compact is justified by the fact that every commercial sensor device has a limited range in which it is able to quantify observations.

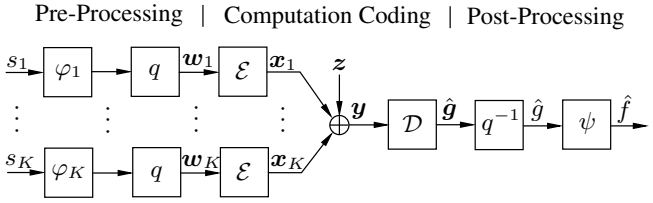


**Fig. 1**. System model for reliably computing a nomographic function $f(s_1,\dots,s_K) = \psi\big(\sum_{i=1}^K \varphi_i(s_i)\big)$ over a Gaussian MAC where appropriate data pre- and post-processing matches the channel to $f$.

and $\psi \in \mathsf{F}(\mathbb{R})$ such that $f$ can be represented in the form

$$f(x_1,\dots,x_K) = \psi\left(\sum\nolimits_{i=1}^K \varphi_i(x_i)\right) \tag{2}$$

is called *nomographic function*. The space of all nomographic functions with domain $\mathbb{A}^K$ is denoted by $\mathsf{N}(\mathbb{A}^K)$.

*Remark* 1. In what follows, the functions $\{\varphi_i\}_{i=1}^K$ are called *pre-processing functions* and $\psi$ is called *post-processing* function. These functions are chosen to match the summation structure of the Gaussian MAC to the desired function. See [4] for examples.

The problem to be solved in this paper is therefore to reliably and efficiently compute elements of $\mathsf{N}(\mathbb{S}^K)$ by using an appropriate nested lattice computation code from [6], [5].

## 3. RELIABLE COMPUTATION OF SOME NOMOGRAPHIC FUNCTIONS

In the following, we confine our attention to those elements of $\mathsf{N}(\mathbb{S}^K)$ for which we can find continuous pre- and post-processing functions. Note that although we assume fixed realizations of sensor readings (i.e., single function computations), the scheme proposed in this section can be easily extended to sequences of sensor readings.

### 3.1. Data Pre- and Post-Processing

Since the measurement space $\mathbb{S}$ is assumed to be compact, the ranges of pre-processing functions are also compact and we denote them by $\Pi_i$ in what follows (i.e., $\forall s_i \in \mathbb{S} : \varphi_i(s_i) \in \Pi_i$, $i = 1,\dots,K$). So, the union $\Pi \coloneqq \bigcup_{i=1}^K \Pi_i$ is compact as well and we denote by $\pi_{\max} \coloneqq \max_{\xi\in\Pi} |\xi|$ the maximal element of $\Pi$ in absolute value.

To reliably compute some real-valued nomographic functions over a finite capacity channel, each node has to first quantize its pre-processed sensor reading to a length-$k$ message. More precisely, each node uses the same mapping $q : \Pi \to \{0,\dots,b-1\}^k$, $q\big(\varphi_i(s_i)\big) = \boldsymbol{w}_i$, $i = 1,\dots,K$, where $b \geq 2$ is assumed to be prime (see Fig. 1). To get an idea of how $q$ is working recall that every $\varphi_i(s_i) \in \Pi$ has a unique $b$-adic expansion [8]

$$\varphi_i(s_i) = (-1)^{\eta_i} \sum\nolimits_{j=-\ell}^{\infty} \frac{w_{ij}}{b^j} = \lim_{m\to\infty} (-1)^{\eta_i} \sum\nolimits_{j=-\ell}^{m} \frac{w_{ij}}{b^j} , \tag{3}$$

where $w_{ij} \in \{0,\dots,b-1\}$ with $w_{ij} \neq b-1$ for infinitely many $j$. Observe that $\eta_i \in \{0,1\}$ depends on the sign of $\varphi_i(s_i)$.

Consider now the approximation

$$\varphi_i(s_i) \approx \hat{\varphi}_i(s_i) = (-1)^{\eta_i} \sum\nolimits_{j=-\ell}^{m} w_{ij} b^{-j} \tag{4}$$

by terminating the infinite series (3). Then, setting $k \coloneqq m + \ell + 2$, the quantizer $q$ simply forms length-$k$ messages by extracting the weights and the sign from expansion (4) that represent the pre-processed sensor readings up to precision

$$\big|\varphi_i(s_i) - \hat{\varphi}_i(s_i)\big| < b^{-m} = b^{-k+\ell+2} . \tag{5}$$

Now, for computing some $f \in \mathsf{N}(\mathbb{S}^K)$, the FC needs a reliable estimate of the corresponding $g(s_1,\dots,s_K) \coloneqq \sum_{i=1}^K \hat{\varphi}_i(s_i)$, which is equivalent to reliably computing the mod $b$ sum of messages

$$\boldsymbol{g} \coloneqq \bigoplus_{i=1}^K \boldsymbol{w}_i \tag{6}$$

over the Gaussian MAC. Once the FC knows $\boldsymbol{g}$, the data post-processing (see Fig. 1) consists of the inverse quantizer that puts $\boldsymbol{g}$ into expansion (4), and of the post- processing function $\psi$ that provides an estimate of the desired function value $f(s_1,\dots,s_K)$.

*Remark* 2. To avoid any overflow in the modulo addition (6), we choose in (4) $\ell = \lfloor \log_b(K\pi_{\max}) \rfloor$.

A crucial step in achieving reliable computations is the protection of (6) against the noise, which we want to ensure by using a lattice coding scheme from [6], [5] as described in the following.

### 3.2. Lattice Computation Coding

First, we recall some notions on nested lattice codes from [9], [10].

**Definition 2.** An $n$-dimensional *lattice* $\Lambda$ is a discrete additive subgroup of the Euclidean space $\mathbb{R}^n$ that is closed under addition and subtraction (i.e., $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \Lambda \Rightarrow \boldsymbol{\lambda}_1 \pm \boldsymbol{\lambda}_2 \in \Lambda$). Any lattice can be specified by a generator/basis matrix $\boldsymbol{G} \in \mathbb{R}^{n\times n}$:

$$\Lambda = \{\boldsymbol{\lambda} = \boldsymbol{G}\boldsymbol{\mu} \,|\, \boldsymbol{\mu} \in \mathbb{Z}^n\} =: \boldsymbol{G}\mathbb{Z}^n .$$

**Definition 3.** A *quantizer* associated with lattice $\Lambda$ is a map $Q_\Lambda : \mathbb{R}^n \to \Lambda$ that assigns any point $\boldsymbol{\mu} \in \mathbb{R}^n$ to the nearest point in $\Lambda$:

$$Q_\Lambda(\boldsymbol{\mu}) = \arg\min_{\boldsymbol{\lambda}\in\Lambda} \|\boldsymbol{\mu} - \boldsymbol{\lambda}\|_2 .$$

**Definition 4.** The *fundamental Voronoi region* of an $n$-dimensional lattice $\Lambda$, denoted as $\mathcal{V}$, is the set of all points in $\mathbb{R}^n$ that quantize to the zero vector:

$$\mathcal{V} \coloneqq \{\boldsymbol{\mu} \in \mathbb{R}^n \,|\, Q_\Lambda(\boldsymbol{\mu}) = \boldsymbol{0}\} .$$

**Definition 5.** The second moment (per dimension) of a lattice $\Lambda \subset \mathbb{R}^n$ is defined as

$$\sigma^2(\Lambda) \coloneqq \frac{1}{n\,\mathrm{Vol}(\mathcal{V})} \int_{\mathcal{V}} \|\boldsymbol{x}\|_2^2 \,\mathrm{d}\boldsymbol{x} . \tag{7}$$

**Definition 6.** A lattice $\Lambda_\mathrm{s}$ is *nested* in a lattice $\Lambda_\mathrm{c}$ if $\Lambda_\mathrm{s} \subset \Lambda_\mathrm{c}$ (i.e., $\Lambda_\mathrm{s}$ is a sublattice of $\Lambda_\mathrm{c}$). The lattice $\Lambda_\mathrm{s}$ with fundamental Voronoi region $\mathcal{V}_\mathrm{s}$ is called shaping lattice whereas $\Lambda_\mathrm{c}$ with fundamental Voronoi region $\mathcal{V}_\mathrm{c}$ is called coding lattice.

Fig. 2 depicts an example of a nested lattice pair, where $\boldsymbol{G}\mathbb{Z}^2$ with basis matrix $\boldsymbol{G} = \begin{pmatrix} \sqrt{3}/2 & 0 \\ 1/2 & 1 \end{pmatrix}$ generates the coding lattice.

**Definition 7.** The *modulo operation* with respect to a lattice $\Lambda$ provides for any $\boldsymbol{\mu} \in \mathbb{R}^n$ the quantization error

$$[\boldsymbol{\mu}] \bmod \Lambda \coloneqq \boldsymbol{\mu} - Q_\Lambda(\boldsymbol{\mu}) ,$$

which is always in $\mathcal{V}$. For all $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^n$ and $\Lambda \subset \Lambda'$ it satisfies a distributive and commutative law:

$$[\boldsymbol{\mu} + \boldsymbol{\nu}] \bmod \Lambda = \big[[\boldsymbol{\mu}] \bmod \Lambda + \boldsymbol{\nu}\big] \bmod \Lambda \tag{8}$$

$$[Q_{\Lambda'}(\boldsymbol{\mu})] \bmod \Lambda = [Q_{\Lambda'}([\boldsymbol{\mu}] \bmod \Lambda)] \bmod \Lambda . \tag{9}$$

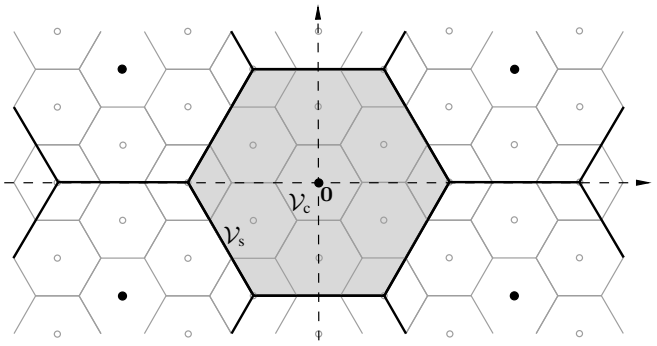


**Fig. 2**. Part of a nested lattice $\Lambda_s \subset \Lambda_c$ in $\mathbb{R}^2$ with $\mathcal{V}_s$ the fundamental Voronoi region of the shaping lattice $\Lambda_s$ (black dots) and $\mathcal{V}_c$ the fundamental Voronoi region of the coding lattice $\Lambda_c$ (white dots).

**Definition 8.** Given any pair of nested lattices $\Lambda_s \subset \Lambda_c$. Then, a *nested lattice code* $\mathcal{C}$ is defined as the set of all points from $\Lambda_c$ that are within the fundamental Voronoi region of the shaping lattice:

$$\mathcal{C} \coloneqq \Lambda_c \cap \mathcal{V}_s \ . \tag{10}$$

With the above definition, the rate (in bits per channel use) of an $n$-dimensional nested lattice code $\mathcal{C}$ is

$$R = \frac{1}{n}\log_2(|\mathcal{C}|) = \frac{1}{n}\log_2\left(\frac{\mathrm{Vol}(\mathcal{V}_s)}{\mathrm{Vol}(\mathcal{V}_c)}\right) \ . \tag{11}$$

*Remark* 3. The essential structural property of a nested lattice code is linearity, which means that any sum of lattice codewords modulo the shaping lattice is a codeword itself:

$$\forall \boldsymbol{x}_1, \ldots, \boldsymbol{x}_K \in \mathcal{C} \Rightarrow \left[\sum\nolimits_{i=1}^{K} \boldsymbol{x}_i\right] \bmod \Lambda_s \in \mathcal{C} \ . \tag{12}$$

#### *3.2.1. Encoding*

To protect the sum (6) against noise, each sensor node uses the same $n$-dimensional nested lattice code $\mathcal{C}$, chosen from the code sequences constructed in [10], with a shaping lattice that is scaled such that the second moment is equal to some power constraint $P > 0$ (i.e., $\sigma^2(\Lambda_s) = P$). Consequently, each node is equipped with the same lattice encoder (see Fig. 1)

$$\mathcal{E} : \{0, \ldots, b-1\}^k \to \mathcal{C} \subset \mathbb{R}^n \tag{13}$$

that maps length-$k$ messages to length-$n$ lattice codewords (i.e., $\boldsymbol{x}_i = \mathcal{E}(\boldsymbol{w}_i)$). Due to the appropriate scaling of the shaping lattice, each codeword meets the transmit power constraint $\frac{1}{n}\|\boldsymbol{x}_i\|_2^2 \leq P$.

The coding rate at each node is with (11) equal to

$$R = \frac{k}{n}\log_2(b) = \frac{1}{n}\log_2\left(\frac{\mathrm{Vol}(\mathcal{V}_s)}{\mathrm{Vol}(\mathcal{V}_c)}\right) \ . \tag{14}$$

In what follows, we assume that the encoding function (13) fulfills

$$\mathcal{E}^{-1}\left(\left[\sum\nolimits_{i=1}^{K} \mathcal{E}(\boldsymbol{w}_i)\right] \bmod \Lambda_s\right) = \bigoplus_{i=1}^{K} \boldsymbol{w}_i \ . \tag{15}$$

The existence of such linearity preserving encoders is shown in [6].

#### *3.2.2. Decoding*

Once the sensor nodes simultaneously used the Gaussian MAC $n$ times, the FC is aware of the receive vector $\boldsymbol{y} = \sum_{i=1}^{K} \boldsymbol{x}_i + \boldsymbol{z}$, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, N\boldsymbol{I}_n)$ (see Fig. 1). Then, to obtain an estimate of (6), the FC applies a decoding function $\mathcal{D} : \mathbb{R}^n \to \{0, \ldots, b-1\}^k$ that consists of an *Euclidean nearest neighbor decoder* [9] followed by the inverse of the encoding function due to (15):

$$\mathcal{D}(\boldsymbol{y}) = \mathcal{E}^{-1}\left([Q_{\Lambda_c}(\boldsymbol{y})] \bmod \Lambda_s\right) = \hat{\boldsymbol{g}} \ . \tag{16}$$

Obviously, the nearest neighbor decoder quantizes the receive vector onto the coding lattice and subsequently reduces the outcome to the shaping lattice, which guarantees that the resulting lattice point is a codeword (see [9] and [10] for more details).

Let $\delta > 0$ be arbitrary. Then, the modulo $b$ sum of messages is said to be decoded with error probability $\delta$, if $\mathbb{P}(\boldsymbol{g} \neq \hat{\boldsymbol{g}}) < \delta$. Since we have not introduced any probability distribution on the pre-processed sensor readings, the error probability is small for every codeword and thus for every choice of $\{\varphi_i(s_i) \in \Pi\}_{i=1}^{K}$. Hence, $\mathbb{P}(\boldsymbol{g} \neq \hat{\boldsymbol{g}})$ is a maximum error probability [5].

### 3.3. Achievable Computation Rate

Given the computation scheme described above, the following theorem provides as an extension of [5, Thm. 2] the corresponding achievable computation rate at which some elements from $\mathsf{N}(\mathbb{S}^K)$ can be computed within precision $\varepsilon > 0$ (in probability).

**Theorem 1.** *Let $f \in \mathsf{N}(\mathbb{S}^K)$ with continuous pre- and post-processing functions. Furthermore, let $\varepsilon > 0$ be arbitrary and the maximum transmit power at each node $P > N$. Then, $f$ can be computed over the Gaussian MAC with error probability $\mathbb{P}(|f - \hat{f}| > \varepsilon) \to 0$ exponentially fast for $n \to \infty$ and sufficiently large $k$ and $b$ if the rate (14) fulfills*

$$R < \frac{1}{2}\log_2(P/N) \ . \tag{17}$$

*Proof.* Consider any $f(s_1, \ldots, s_K) = \psi(\sum_i \varphi_i(s_i))$ with continuous pre- and post-processing functions. According to the approximations (4) let $\tilde{f}(s_1, \ldots, s_K) \coloneqq \psi(\sum_i \hat{\varphi}_i(s_i))$. Now, let $\varepsilon > 0$ be arbitrary and choose $k$ and $b$ in dependency of the pre- and post-processing functions such that $|f - \tilde{f}| \leq \varepsilon$. Then, simultaneously transmitting the corresponding encoded messages $\boldsymbol{x}_i = \mathcal{E}(\boldsymbol{w}_i)$, with $\boldsymbol{w}_i = q\big(\varphi_i(s_i)\big)$, results with the decoder (16) in

$$\mathcal{D}(\boldsymbol{y}) = \mathcal{E}^{-1}\left(\left[Q_{\Lambda_c}\left(\sum\nolimits_{i=1}^{K} \boldsymbol{x}_i + \boldsymbol{z}\right)\right] \bmod \Lambda_s\right) \tag{18}$$

$$= \mathcal{E}^{-1}\left(\left[Q_{\Lambda_c}\left(\left[\sum\nolimits_{i=1}^{K} \boldsymbol{x}_i\right] \bmod \Lambda_s + \boldsymbol{z}\right)\right] \bmod \Lambda_s\right) \tag{19}$$

$$= \mathcal{E}^{-1}\left([Q_{\Lambda_c}(\boldsymbol{x} + \boldsymbol{z})] \bmod \Lambda_s\right) \ , \tag{20}$$

where (19) follows from the commutative and distributive properties (9) and (8), respectively. Because of (12), we have $\boldsymbol{x} \in \mathcal{C}$ and thus $\boldsymbol{y}$ is a codeword corrupted by Gaussian noise.

Then, with the sequence of nested lattices from [10], we have $\forall \boldsymbol{x} \in \mathcal{C} : \lim_{n\to\infty} \mathbb{P}\big([Q_{\Lambda_c}(\boldsymbol{x} + \boldsymbol{z})] \bmod \Lambda_s \neq \boldsymbol{x}\big) = 0$ exponentially fast as long as the rate (14) is smaller than $\frac{1}{2}\log_2\big(\frac{P}{N}\big)$. Consequently, $\mathbb{P}(\boldsymbol{g} \neq \hat{\boldsymbol{g}}) \to 0$ exponentially fast in $n$ and thus also

$$\mathbb{P}\big(\hat{g}(s_1, \ldots, s_K) \neq g(s_1, \ldots, s_K)\big) \to 0 \ , \tag{21}$$

where $g(s_1, \ldots, s_K) = \sum_{i=1}^{K} \hat{\varphi}_i(s_i)$ and $\hat{g}(s_1, \ldots, s_K) = q^{-1}(\hat{\boldsymbol{g}})$ the corresponding estimate at the FC. Now, choose $\hat{\boldsymbol{g}}$ such that

$\psi\big(\hat{g}(s_1,\ldots,s_K)\big) \neq \psi\big(g(s_1,\ldots,s_K)\big)$. Then, it follows that the particular choice of $\hat{\boldsymbol{g}}$ also implies $\hat{g}(s_1,\ldots,s_K) \neq g(s_1,\ldots,s_K)$, since $\psi$ is a function. Summarizing all such outage events into the sets $A \coloneqq \{\hat{\boldsymbol{g}} \,|\, \hat{g}(s_1,\ldots,s_K) \neq g(s_1,\ldots,s_K)\}$ and $B \coloneqq \big\{\hat{\boldsymbol{g}} \,|\, \psi\big(\hat{g}(s_1,\ldots,s_K)\big) \neq \psi\big(g(s_1,\ldots,s_K)\big)\big\}$, we have $B \subseteq A$ and therefore $\mathbb{P}(B) \leq \mathbb{P}(A)$ due to the monotonicity of probability and the measurability of $\psi$. Hence, we can conclude from (21) also $\mathbb{P}\big(\psi\big(\hat{g}(s_1,\ldots,s_K)\big) \neq \psi\big(g(s_1,\ldots,s_K)\big)\big) = \mathbb{P}(\hat{f} \neq \tilde{f}) \to 0$ exponentially fast in $n$. This in turn implies $\mathbb{P}(|f-\hat{f}| > \varepsilon) \to 0$ for $n \to \infty$, $k$ and $b$ sufficiently large and $R < \frac{1}{2}\log_2(P/N)$, because almost sure convergence implies convergence in probability. Since $\varepsilon > 0$ was chosen arbitrary, this proves the theorem. ■

**Example 1.** Let $\mathbb{S}$ be a subset of the nonnegative reals and let the desired function be the geometric mean $f(s_1,\ldots,s_K) = \big(\prod_{i=1}^K s_i\big)^{1/K}$. With the continuous pre-processing functions $\varphi_i(s_i) = \log_{\mathrm{e}}(s_i)$, $i = 1,\ldots,K$, and the continuous (and therefore measurable) post-processing function $\psi(g) = \exp_{\mathrm{e}}(g/K)$ is $f \in \mathsf{N}(\mathbb{S}^K)$. Thus, the geometric mean is according to Theorem 1 computable with arbitrary precision up to rate (17).

### 3.4. Number of Required Channel Uses

Due to technical reasons, we consider in the following a certain subset of $\mathsf{N}(\mathbb{S}^K)$ that results from requiring pre-processing functions to be continuous and post-processing functions to be $\alpha$-Hölder continuous defined as follows [11].

**Definition 9.** A function $h : \mathbb{D} \subseteq \mathbb{R} \to \mathbb{R}$ is called *$\alpha$-Hölder continuous*, or belongs to the function space $\mathsf{C}^{0,\alpha}(\mathbb{D})$, if and only if there exist positive real constants $L$ and $\alpha$, where $0 < \alpha \leq 1$, such that

$$\forall x_1, x_2 \in \mathbb{D} : \big|h(x_1) - h(x_2)\big| \leq L\big|x_1 - x_2\big|^{\alpha} . \qquad (22)$$

Accordingly, we define the following function space of interest:

$$\mathsf{N}^{0,\alpha}(\mathbb{S}^K) \coloneqq \Big\{ f : \mathbb{S}^K \to \mathbb{R} \,\Big|\, \exists(\varphi_1,\ldots,\varphi_K,\psi) \in \mathsf{C}^0(\mathbb{S}) \times \cdots$$
$$\cdots \times \mathsf{C}^0(\mathbb{S}) \times \mathsf{C}^{0,\alpha}(\mathbb{R}) : f(s_1,\ldots,s_K) = \psi\Big(\sum\nolimits_{i=1}^K \varphi_i(s_i)\Big)\Big\}.$$

*Remark* 4. Unfortunately, $\mathsf{N}^{0,\alpha}(\mathbb{S}^K) \subset \mathsf{N}(\mathbb{S}^K)$ is a nowhere dense subset of $\mathsf{C}^0(\mathbb{S}^K)$, which can be deduced from [4, Thm. 2] and the fact that $\alpha$-Hölder continuity is stronger than (pointwise) continuity.

**Theorem 2.** *Let $f \in \mathsf{N}^{0,\alpha}(\mathbb{S}^K)$ be arbitrary. Then, for computing $f$ with a given sufficiently small precision $\varepsilon > 0$ over the Gaussian MAC, the number of channel uses has to be*

$$n > 2\frac{\log_2\big(L^\star(\pi_{\max}K^2b^2)^\alpha\varepsilon^{-1}\big)}{\alpha\log_2(P/N)} , \qquad (23)$$

*where $L^\star$ is the smallest possible constant in (22).*

*Proof.* Let $z \coloneqq \sum_{i=1}^K \varphi_i(s_i)$ and $\hat{z} \coloneqq \sum_{i=1}^K \hat{\varphi}_i(s_i)$. Then, we conclude from Remark 2 and (5) with the triangle inequality

$$\begin{aligned} |z-\hat{z}| &\leq \sum\nolimits_{i=1}^K \big|\varphi_i(s_i) - \hat{\varphi}_i(s_i)\big| \\ &< Kb^{-k+\lfloor\log_b(K\pi_{\max})\rfloor+2} \\ &< \pi_{\max}K^2b^{-k+2} . \end{aligned}$$

Since $f \in \mathsf{N}^{0,\alpha}(\mathbb{S}^K)$ and therefore $\psi \in \mathsf{C}^{0,\alpha}(\mathbb{R})$, we then have

$$|f-\hat{f}| = \big|\psi(z) - \psi(\hat{z})\big| \leq L|z-\hat{z}|^\alpha < L\pi_{\max}^\alpha K^{2\alpha}b^{-\alpha(k-2)} \leq \varepsilon$$

for sufficiently large $k, b$. Choosing $L^\star$ to be the smallest possible $L$ for $\psi$ and taking the logarithm on the right hand side results in

$$\tfrac{1}{\alpha}\log_2\big(L^\star(\pi_{\max}K^2b^2)^\alpha/\varepsilon\big) \leq k\log_2(b) = n\left(\tfrac{1}{2}\log_2\big(\tfrac{P}{N}\big) - \xi\right),$$

where the equality follows with $\xi > 0$ from (14) and Theorem 1. Assuming $\varepsilon$ to be sufficiently small ensures that $k$ and $b$ are large enough. ■

**Example 2.** Let the desired function be the Euclidean norm $f(s_1,\ldots,s_K) = \sqrt{\sum_{i=1}^K s_i^2}$. Obviously is $f \in \mathsf{N}(\mathbb{S}^K)$ with $\varphi_i(s_i) = s_i^2$, $i = 1,\ldots,K$, and $\psi(g) = \sqrt{g}$. Since the pre-processing functions are continuous and $\psi$ is $\frac{1}{2}$-Hölder continuous, we even have $f \in \mathsf{N}^{0,\frac{1}{2}}(\mathbb{S}^K)$.

*Remark* 5. Note that Theorem 2 is only a necessary condition for $n$.

### 3.5. Performance Comparison

As mentioned in the introduction, the standard approach for solving computation problems in sensor networks is to recover all sensor readings at the FC to *subsequently* compute the desired function value. For a simple comparison with the computation scheme described above suppose that the FC aims at reliably recovering all the pre-processed sensor readings instead. Therefore, the nodes employ a standard multiple-access code, whereas the FC successively decodes all of the messages $\boldsymbol{w}_i$, $i = 1,\ldots,K$, with vanishing average error probability at a maximum rate $R = \frac{1}{2K}\log_2(1+\frac{KP}{N})$ [7]. As a result, the FC can compute any $f \in \mathsf{N}(\mathbb{S}^K)$ within accuracy $\varepsilon > 0$ for sufficiently large message lengths at a rate that is approximately $K$ *times smaller* as the achievable rate in Theorem 1.

## 4. DISCUSSION

Communicating real-valued sensor readings with arbitrary precision over a Gaussian channel generally fails due to the capacity that is bounded away from infinity (i.e., for limited power and bandwidth). The same applies if the objective is to compute precise function values of those measurements with the advantage that the additive structure of the Gaussian MAC can be exploited. For instance, if a fixed pre-defined accuracy suffices, some linear and nonlinear nomographic functions can be computed considerably faster than with any interference avoiding strategy. This results from the structure of nomographic functions and the properties of nested lattice codes.

## 5. RELATION TO PRIOR WORK

Besides [2] and our own work, the computation of special functions over a MAC is considered in for instance [12]–[15]. To achieve performance gains, all of these works assume a structural match between the function to be computed and the operation the underlying MAC naturally performs. In this context, the authors of [16] analyze how a corresponding mismatch impacts the performance gains and they claim that for most pairs of desired functions and MACs an interference avoiding strategy is optimal. Note that the pre- and post-processing functions in this paper are used to settle mismatches between the Gaussian MAC and nonlinear desired functions.

As already mentioned, Nazer and Gastpar propose in [6] a lattice coding scheme that allows the reliable computation of linear combinations of node messages. The same setting is in [17] extended to transmitting nodes that can cooperate, which can lead to increased computation rates.

## 6. REFERENCES


[1] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 755–764, April 2005.

[2] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.

[3] M. Goldenbaum and S. Stańczak, "Robust analog function computation via wireless multiple-access channels," *submitted to IEEE Trans. Commun.*, 2012, [Online]. Available: http://arxiv.org/abs/1210.2967.

[4] M. Goldenbaum, H. Boche, and S. Stańczak, "Analyzing the space of functions analog-computable via wireless multiple-access channels," in *Proc. 8th Int. Symp. Wireless Communication Systems (ISWCS)*, Aachen, Germany, Nov. 2011, pp. 779–783.

[5] B. Nazer, A. G. Dimakis, and M. Gastpar, "Local interference can accelerate gossip algorithms," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 876–887, Aug. 2011.

[6] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, 2nd edition, 2006.

[8] U. Kulisch, *Computer Arithmetic and Validity: Theory, Implementation, and Applications*, vol. 33 of *de Gruyter Studies in Mathematics*, De Gruyter, Berlin·New York, 2008.

[9] U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log(1 + \mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.

[10] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Trans. Inf. Theory*, vol. 51, no. 10, pp. 3401–3416, Oct. 2005.

[11] H. Triebel, *Higher Analysis*, Johann Ambrosius Barth, Leipzig·Berlin·Heidelberg, 1992.

[12] T. M. Duman and M. Salehi, "Decentralized detection over multiple-access channels," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 2, pp. 469–476, Apr. 1998.

[13] G. Mergen and L. Tong, "Type based estimation over multiaccess channels," *IEEE Trans. Signal Process.*, vol. 54, no. 2, pp. 613–626, Feb. 2006.

[14] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: Physical-layer network coding," in *Proc. 12th Annu. Int. Conf. Mobile Computing and Networking (ACM MobiCom)*, Los Angeles, CA, USA, Sept. 2006, pp. 358–365.

[15] L. Keller, N. Karamchandani, and C. Fragouli, "Function computation over linear channels," in *Proc. IEEE Int. Symp. Network Coding (NetCod)*, Toronto, Canada, June 2010, pp. 1–6.

[16] N. Karamchandani, U. Niesen, and S. Diggavi, "Computation over mismatched channels," *accepted for publication in IEEE J. Sel. Areas Commun.*, 2013, [Online]. Available: http://arxiv.org/abs/1204.5059.

[17] M. Nokleby and B. Aazhang, "Unchaining from the channel: Cooperative computation over multiple-access channels," in *Proc. IEEE Information Theory Workshop (ITW)*, Paraty, Brazil, Oct. 2011, pp. 593–597.