Distributed Processing of a Fractal Array Beamformer

P Karagiannakis*, K Thompson*, J Corr*, IK Proudler†, S Weiss*

*Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, Scotland
†School of Electronic, Electrical and Systems Engineering, Loughborough University, Loughborough, UK
{philipp.karagiannakis,keith.thompson,stephan.weiss}@strath.ac.uk

Abstract. Fractals have been proven as potential candidates for satellite flying formations, where its different elements represent a thinned array. The distributed and low power nature of the nodes in this network motivates distributed processing when using such an array as a beamformer. This paper proposes such initial idea, and demonstrates that benefits such as strictly limited local processing capability independent of the array's dimension and local calibration can be bought at the expense of a slightly increased overall cost.

1. Introduction

Recent work established a Purina fractal geometry as a formation of fractionated spacecraft as an alternative to larger satellites [1,2]. Additionally, the Purina fractal's structural sparseness combines a significant aperture and therefore resolution while avoiding spatial aliasing as long as at least some sensors are sufficiently closely located [3–6], thus offering advantages that otherwise have to be achieved through thinning of arrays [7,8].

In order to exploit the fractionated nature of a satellite as proposed in [2], we aim to mirror its fractal structure in the processing architecture, since the lack of a central processing node motivates the design of a distributed beamformer. In the past such efforts have e.g. concentrated on the distributed estimation of the covariance matrix [9], distributed signal enhancement with bandwidth constraints [10] or the use of factor graphs [11] and specifically Pearl's algorithm [12], which could lead to the implementation of general algorithms in a distributed fashion. Some distributed algorithms have also been developed for spatially separated subarrays [13,14] with the main emphasis on the iterative approximation of jointly optimal results.

Our aim here is to use a hierarchical distributed processing structure which closely mirrors the fractal architecture of the array. In particular, we propose to use nested subarrays, whereby a subarray takes the shape of the generating fractal. The beamformer output can be hierarchically computed such that, independent of the dimension of the Purina array, the number of computations per node are strictly limited, even though the overall number of computations is slightly increased compared to directly processing the samples collected by all sensors.

Below, we first review characteristics and the generation of the Purina fractal array in Sec. 2.. The beamformer output, its quiescent response, and its distributed computation are outlined in Sec. 3., while some results and discussions are provided in
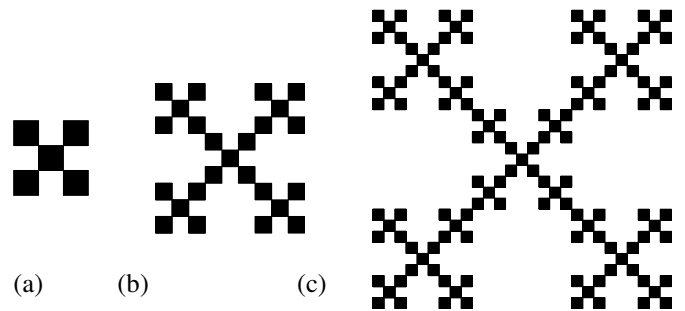

Figure 1. First three stages of growth of the Purina fractal array for (a) $P = 1$, (b) $P = 2$, and (c) $P = 3$.

Sec. 4.. Finally, conclusions are drawn in Sec. 5..

2. Purina Fractal Array

2.1 Purina Array Generation

The Purina fractal pattern yields a thinned 3-by-3 symmetric planar array, which at growth stage $P = 1$ has the simple subarray S_1

$$S_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad , \tag{1}$$

also referred to as the generating array. Higher growth stages $P \in \mathbb{N}$, $P > 1$ are defined recursively by

$$S_P = S_1 \otimes S_{P-1} \quad , \tag{2}$$

with $\otimes$ denoting the Kronecker product, whereby a unit entry means that an element is switched on, while zero indicates that the array element is switched off. Fig. 1 demonstrates the first three stages of growth for the Purina fractal array.

2.2 Hierarchy and Labelling

For the analysis below, we will organise sensors according to their fractal scale, $p \in \mathbb{Z}$, $p \leq P$, which describes the different hierarchical layers of the architecture up to the full growth
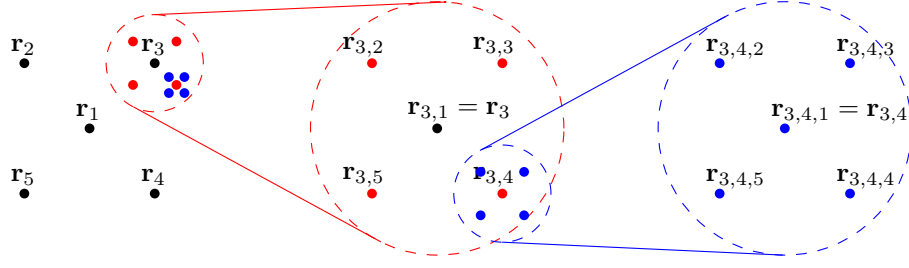
Figure 2. Nested labelling of array elements at fractal scale $p = 1$ with sensor locations $\mathbf{r}_k$, fractal scale $p = 2$ with locations $\mathbf{r}_{k,l}$, and fractal scale $p = 3$ with locations $\mathbf{r}_{k,l,m}$, with $k, l, m \in \{1 \ldots 5\}$.

stage P. The elements at the coarsest level, $p = 1$, are given a single index, elements at fractal scale $p = 2$ a double index, and so on, until the elements at the finest scale $p = P$ are labelled using P subscripts. For the three coarsest levels of a Purina fractal array, an example is provided in Fig. 2. Note that in general,

$$\mathbf{r}_{k,l,\ldots,r,q,1} = \mathbf{r}_{k,l,\ldots,r,q} \quad , \tag{3}$$

and in particular

$$\mathbf{r}_{1,1,\ldots,1,1} = \mathbf{r}_{1,1,\ldots,1} = \cdots = \mathbf{r}_1 \quad . \tag{4}$$

Using these sensor locations, below we will be able to define a distributed beamforming system exploiting the fractal scale structure of the Purina array, by labelling the narrowband beamforming coefficient and the data sample collected at time instance n in the sensor location denoted by a vector $\mathbf{r}_{k,l,\ldots,p,q}$ as $w_{k,l,\ldots,p,q}$ and $x_{k,l,\ldots,p,q}[n]$, respectively.

3. Distributed Beamformer

This section derives a beamformer formulation for using distributed processing of inputs based on the definition of the beamformer output in Sec. 3.1 and its coefficients for the quiescent case in Sec. 3.2. A restructuring of the equations in Sec. 3.3 yields a formulation with a slightly increased cost, which however allows to calibrate information that is only available within subarrays.

3.1 Beamformer Output

The overall beamformer response is given by

$$y[n] = \mathbf{w}^{\mathrm{H}} \mathbf{x}[n] \tag{5}$$

$$= \underbrace{\sum_{u_P=1}^{5} \cdots \sum_{u_2=1}^{5} \sum_{u_1=1}^{5}}_{P \text{ terms}} w_{u_P,\ldots u_2,u_1} x_{u_P,\ldots,u_2,u_1}[n] \quad , \tag{6}$$

whereby $\mathbf{w}$ and $\mathbf{x}[n]$ are the stacked coefficient and data vectors at time n, $\{\cdot\}^{\mathrm{H}}$ denotes Hermitian transpose. The computations that are required for one output sample $y[n]$ are constituted by 5^P multiply-accumulate operation, that would under normal circumstances be executed in a central processing node. Interestingly, the nesting of the summation terms in (6) provides a

natural hierarchy in calculating the output, whereby intermediate outputs of nested subarrays are defined as

$$y[n] = \sum_{u_P=1}^{5} y_{u_P}[n] \tag{7}$$

$$\vdots$$

$$= \sum_{u_P=1}^{5} \cdots \sum_{u_2=1}^{5} y_{u_P,\ldots,u_2}[n] \tag{8}$$

$$= \sum_{u_P=1}^{5} \cdots \sum_{u_2=1}^{5} \sum_{u_1=1}^{5} y_{u_P,\ldots,u_2,u_1}[n] \quad . \tag{9}$$

The quantities under the sum on the r.h.s. of (9) denote the output of subarrays at different fractal scales of the array, such that $y_P[n]$ are the outputs at the 5 nodes at the coarsest level as shown on the left side of Fig. 2, and outputs with an increasing number of subscripts refer to intermediate outputs at finer fractal scales.

3.2 Quiescent Beamformer Coefficients

Assuming a far field source at a narrowband frequency f which arrives at the array as a planar wave front with normal vector $\mathbf{k}$,

$$\mathbf{k}_{\varphi,\vartheta} = \begin{bmatrix} \cos \varphi \sin \vartheta \\ \sin \varphi \sin \vartheta \\ \cos \vartheta \end{bmatrix} \quad , \tag{10}$$

i.e. with azimuth φ and elevation angle ϑ, the relative time delay $\tau_{u_P,\ldots u_2,u_1}$ experienced at location $\mathbf{r}_{u_P,\ldots,u_2,u_1}$ relative to the centre element at $\mathbf{r}_1$ is given by

$$\tau_{u_P,\ldots,u_2,u_1} = \frac{1}{c} \mathbf{k}_{\varphi,\vartheta}^{\mathrm{T}} (\mathbf{r}_{u_P,\ldots,u_2,u_1} - \mathbf{r}_1) \tag{11}$$

with c denoting the propagation speed in the medium. The quantity $\mathbf{k}_{\varphi,\vartheta}/c$ is also known as the slowness vector of the source.

Given a sampling rate f_s, the narrowband source is charac-

terised by a steering vector $\mathbf{s}_{\varphi,\vartheta}$,

$$\mathbf{s}_{\varphi,\vartheta} = \begin{bmatrix} e^{-j\Omega\tau_{1,\ldots,1,1}} \\ \vdots \\ e^{-j\Omega\tau_{1,\ldots,1,5}} \\ e^{-j\Omega\tau_{1,\ldots,2,1}} \\ \vdots \\ e^{-j\Omega\tau_{5,\ldots,5,5}} \end{bmatrix} \quad , \qquad (12)$$

with $\Omega = 2\pi f / f_s$. For the quiescent case, (12) defines the optimum filter coefficients $\mathbf{w} = \mathbf{s}_{\varphi,\vartheta}^*$, i.e. the matched filter, in the mean square error sense.

3.3 Distributed Processing with Local Calibration

On the finest fractal scale, different from (11) we define the time shift relative to the centre of a subarray,

$$\tilde{\tau}_{u_P,\ldots,u_2,u_1} = \frac{1}{c}\mathbf{k}_{\varphi,\vartheta}^{\mathrm{T}}(\mathbf{r}_{u_P,\ldots,u_2,u_1} - \mathbf{r}_{u_P,\ldots,u_2}) \quad . \qquad (13)$$

Therefore, 5^{P-1} steering vectors $\tilde{\mathbf{s}}_{u_P,\ldots,u_2|\varphi,\vartheta} \in \mathbb{C}^5$,

$$\tilde{\mathbf{s}}_{u_P,\ldots,u_2|\varphi,\vartheta} = \begin{bmatrix} 1 \\ e^{-j\Omega\tilde{\tau}_{u_P,\ldots,u_2,2}} \\ \vdots \\ e^{-j\Omega\tilde{\tau}_{u_P,\ldots,u_2,5}} \end{bmatrix} \qquad (14)$$

emerge at the finest scale. The time delays can therefore be adjusted based on local knowledge of the actual locations $\mathbf{r}_{u_P,\ldots,u_2,u_1}$ within each subarray.

At the next coarser level, 5^{P-2} groups of steering vectors $\tilde{\mathbf{s}}_{u_P,\ldots,u_3|\varphi,\vartheta} \in \mathbb{C}^{25}$ are assembled by weighting contributions of the sub-steering vectors in (14). This weighting reflects the calibration w.r.t. the time difference at this fractal scale,

$$\tilde{\mathbf{s}}_{u_P,\ldots,u_3|\varphi,\vartheta} = \begin{bmatrix} \tilde{\mathbf{s}}_{u_P,\ldots,u_3,1|\varphi,\vartheta} \\ e^{-j\Omega\tilde{\tau}_{u_P,\ldots,u_3,2}}\tilde{\mathbf{s}}_{u_P,\ldots,u_3,2|\varphi,\vartheta} \\ \vdots \\ e^{-j\Omega\tilde{\tau}_{u_P,\ldots,u_3,5}}\tilde{\mathbf{s}}_{u_P,\ldots,u_3,5|\varphi,\vartheta} \end{bmatrix} \quad , \qquad (15)$$

whereby the time delays $\tilde{\tau}_{u_P,\ldots,u_3,u_2}$ represent calibrations w.r.t. the central nodes of the next finer fractal scale,

$$\tilde{\tau}_{u_P,\ldots,u_3,u_2} = \frac{1}{c}\mathbf{k}_{\varphi,\vartheta}^{\mathrm{T}}(\mathbf{r}_{u_P,\ldots,u_3,u_2} - \mathbf{r}_{u_P,\ldots,u_3}) \quad . \qquad (16)$$

The process of (14) and (15) can be iterated until the coarsest fractal scale $p = 1$ is reached.

At the coarsest fractal scale $p = 1$, finally the complete steering vector

$$\mathbf{s}_{\varphi,\vartheta} = \begin{bmatrix} \tilde{\mathbf{s}}_{1|\varphi,\vartheta} \\ e^{-j\Omega\tilde{\tau}_2}\tilde{\mathbf{s}}_{2|\varphi,\vartheta} \\ \vdots \\ e^{-j\Omega\tilde{\tau}_5}\tilde{\mathbf{s}}_{5|\varphi,\vartheta} \end{bmatrix} \quad , \qquad (17)$$
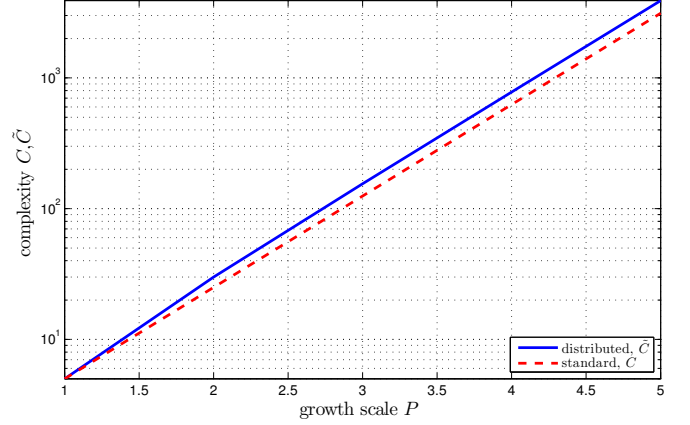

Figure 3. Complexity for standard (C) and distributed processing ($\tilde{C}$) as a function of the growth stage P.

with

$$\tilde{\tau}_{u_p} = \frac{1}{c}\mathbf{k}_{\varphi,\vartheta}^{\mathrm{T}}(\mathbf{r}_{u_P} - \mathbf{r}_1) \qquad (18)$$

is obtained, which matches the original steering vector $\mathbf{s}_{\varphi,\vartheta} \in \mathbb{C}^{5^P}$ in (12).

The computational structure in calculating the output (9) can be performed to match the nested iterative structure of steering vectors presented by (14), (15) and (17). At the finest scale, outputs $\tilde{y}_{u_P,\ldots,u_3,u_2}[n]$ are determined as

$$\tilde{y}_{u_P,\ldots,u_3,u_2}[n] = \sum_{u_1=1}^{5} \tilde{w}_{u_P,\ldots,u_2,u_1} \cdot x_{u_P,\ldots,u_2,u_1}[n] \quad , \qquad (19)$$

with the coefficients $\tilde{w}_{u_P,\ldots,u_2,u_1}$ matched to the modified steering vectors $\tilde{\mathbf{s}}_{u_P,\ldots,u_2|\varphi,\vartheta}$ in (14). From this finest level upwards, at each fractal scale phase corrections as in (15) and (17) are applied when adding up outputs in a divide-and-conquer fashion to finally reach $y[n]$ at the coarsest fractal scale.

4. Discussion, Simulations and Results

4.1 Computational Complexity

The complexity of the direct formulation in (6) via a scalar product requires $C = 5^P$ multiply-accumulates, which might need to be afforded in a central processing node, where data, weights, and any calibration for displaced sensors might be required. For the proposed computational structure in Sec. 3.3, the hierachical processing structure requires a total of

$$\tilde{C} = \sum_{p=1}^{P} 5^p > C \quad . \qquad (20)$$

However, for sufficiently large P, the relative difference between $\tilde{C}$ and C diminishes as shown in Fig. 3, since both approaches possess a complexity of order $\mathcal{O}(5^P)$. However, for the distributed approach, the requirement of not more than 5 multiply-accumulate operations per sensor node — independent of P — emerges as a major benefit. Also, the distributed
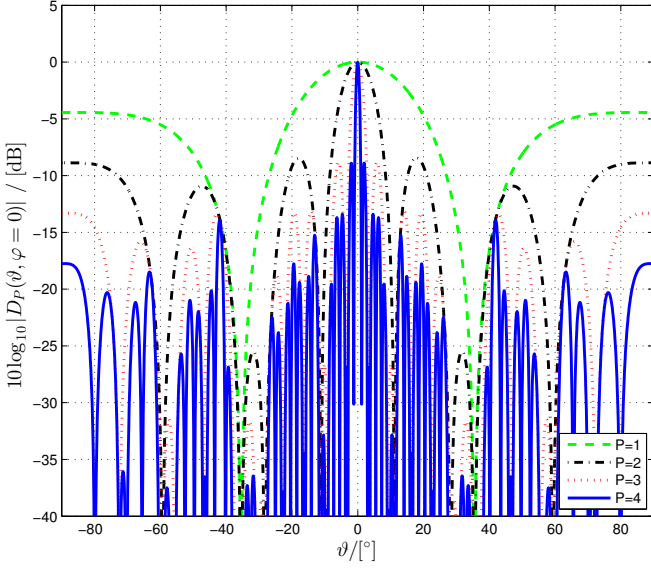
Figure 4. Quiescent beampatterns of Purina array for different growth stages $P = 1, 2, 3$ and 4, assuming that in each case the array elements' minimum spacing satisfies spatial sampling.

Figure 6. Quiescent beampatterns of Purina array adjusted to sample correctly with growth stage $P = 4$, while processing of finer fractal scales for $p = 1, 2$, and 3 operate on a subsampled array.
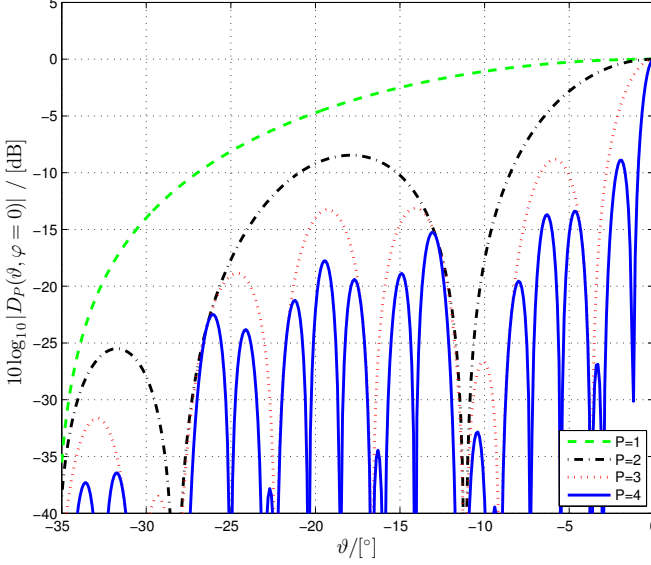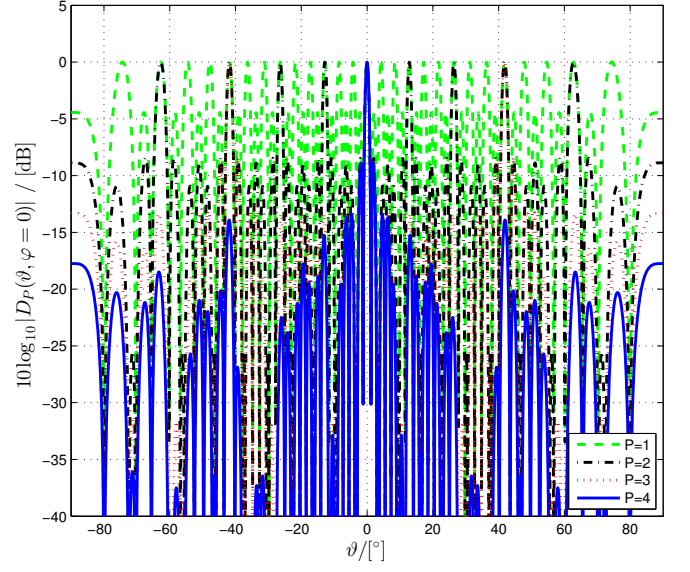

Figure 5. Detail view of Fig. 4, showing the main beam for $P = 1$ and the iterative inscribed characteristics for finer fractal scaled Purina fractal arrays.
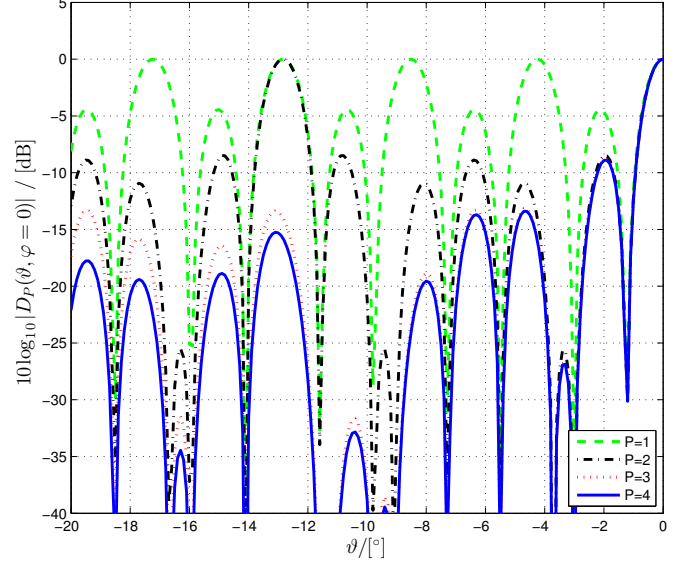

Figure 7. Detail view of Fig. 6, showing the main beam for $P = 4$ and the iterative inscribed characteristics for coarser fractal scales of the Purina arrays.

structure is easier to calibrate, as dislocations of sensors only have to be known at the local subarray level, which matches the control strategy for flying a Purina array in formation, as outlined in [3].

4.2 Beampatterns

A number of sample beampatterns for the Purina array beamformer are shown in Figs. 4 and 5. These beampatterns emerge from a beamformer $D_P(\vartheta, \varphi)$ matched to receive a signal from broadside, $\vartheta = 0°$, and are calculated by probing the array with a set of steering vectors $\mathbf{s}_{\varphi,\vartheta}$ as defined in (12) for variable ele-

vation ϑ,

$$D_P(\vartheta, \varphi) = \mathbf{w}^{\mathrm{H}} \mathbf{s}_{\varphi,\vartheta} \qquad . \tag{21}$$

The azimuth is in this case set to zero, $\varphi = 0°$. Since for every value of P, the minimum distance between array elements is set to fulfil correct spatial sampling, no aliasing occurs, and an increase in P corresponds to an increase in resolution as characterised by the narrowing beamwidth at $\vartheta = 0°$, and lower sidelobe levels. Note that the fractal structure of the array results in "inscribed" or majorised beampatterns where $|D_{P+1}(\vartheta, \varphi)| < |D_P(\vartheta, \varphi)| \forall \vartheta, \varphi, P$.

For illustration purposes, Figs. 6 and 7 show a Purina array with element distances adjusted to satisfy correct spatial sam-

pling for the case $P = 4$, with the beam pattern matching the ones shown in Figs 4 and 5. If only coarser fractal scales $p < P$ are processed, subarrays are spatially subsampled and spatial aliasing can be noticed. Interestingly, again the fractal structure of the array results in majorised beampatterns.

5. Conclusions

We have considered distributed processing for a Purina fractal array, which emerges from a generating subarray to reach a growth stage P over a number of fractal scales $p = 1 \ldots P$. The considered processing consisted of the calculation of a beamformer output, which can exploit the fractal structure to define the distributed processing architecture. As a simple example, we have assumed a quiescent beamformer, which is optimal in a scenario where a single source is embedded in isotropic noise.

The advantages of the discussed processing architecture lie in the fixed maximum complexity per node in the distributed procedure. In addition to limiting the processing power, transmit power is conserved through short hops. Further, the distributed approach matches the position control strategy of the Purina array for formation flying, and allows to consider calibration information in the form of locally known dislocation of sensor elements when computing the beamformer output.

References

[1] J. Leitner, "Formation flying: The future of remote sensing from space," NASA Tech. Report, Goddard Space Flight Center, MD, Tech. Rep., 2004.

[2] G. Punzo, D. J. . Bennett, and M. Macdonald, "A fractally fractionated spacecraft," in *62nd International Astronautical Congress*, Cape Town, South Africa, October 2011.

[3] G. Punzo, P. Karagiannakis, D. Bennet, M. Macdonald, and S. Weiss, "Enabling and exploiting self-similar central symmetry formations," *IEEE Transaction on Aerospace and Electronic Systems*, 2013, (accepted).

[4] D. Werner, R. L. Haupt, and P. Werner, "Fractal antenna engineering: the theory and design of fractal antenna arrays," *IEEE Antennas and Propagation Magazine*, vol. 41, no. 5, pp. 37–58, October 1999.

[5] P. Karagiannakis, S. Weiss, G. Punzo, M. Macdonald, J. Bowman, and R. Stewart, "Impact of a Purina fractal array geometry on beamforming performance and complexity," in *21st European Signal Processing Conference*, Marrakech, Morocco, September 2013, (to appear).

[6] P. Karagiannakis and S. Weiss, "Analysis of a purina fractal beamformer," in *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2013, (accepted).

[7] R. M. Leahy and B. D. Jeffs, "On the Design of Maximally Sparse Beamforming Arrays," *IEEE Transactions on Antennas and Propagation*, vol. 39, no. 8, pp. 1178–1188, August 1991.

[8] G. Cardone, G. Cincotti, and M. Pappalardo, "Design of wideband arrays for low side-lobe level beam patterns by simulated annealing," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 49, no. 8, pp. 1050–1059, August 2002.

[9] U. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4919–4935, November 2008.

[10] A. Bertrand and M. Moonen, "Distributed LCMV beamforming in a wireless sensor network with single-channel per-node signal transmission," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3447–3459, 2013.

[11] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, January 2004.

[12] I. Proudler, S. Roberts, S. Reece, and I. Rezek, "An iterative signal detection algorithm based on Bayesian belief propagation ideas," in *15th International Conference on Digital Signal Processing*, Cardiff, UK, July 2007, pp. 355–358.

[13] K. Zarifi, S. Affes, and A. Ghrayeb, "Distributed processing techniques for beamforming in wireless sensor networks," in *3rd International Conference on Signals, Circuits and Systems*, Medenine, Tunisia, November 2009, pp. 1–5.

[14] H. Ge, I. Kirsteins, and X. Wang, "Adaptive beamforming using distributed antenna arrays: Joint versus distributed processing," in *Forty Fourth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2010, pp. 1107–1111.