# A Compressed Sensing Algorithm for Magnetic Dipole Localization

de Gijsel, Stefan L. ; Vijn, Aad R.P. J.; Tan, Reinier G.

**DOI**
10.1109/JSEN.2022.3184814

**Publication date**
2022

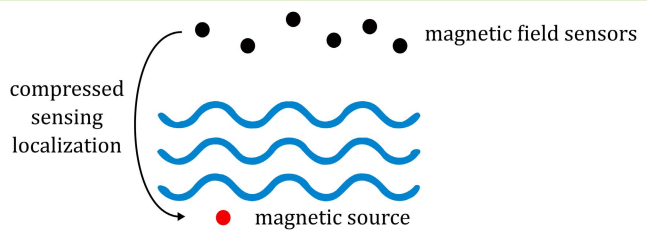
**Document Version**
Final published version

**Published in**
IEEE Sensors Journal

# A Compressed Sensing Algorithm for Magnetic Dipole Localization

Stefan L. de Gijsel, Aad R. P. J. Vijn, and Reinier G. Tan

***Abstract*—This paper proposes an algorithm to localize a magnetic dipole using a limited number of noisy measurements from magnetic field sensors. The algorithm is based on the theory of compressed sensing, and exploits the sparseness of the magnetic dipole in space. Beforehand, a basis consisting of magnetic dipole fields belonging to individual dipoles in an evenly spaced 3D grid within a specified search domain is constructed. In the algorithm, a number of sensors is chosen which measure all three magnetic field components. The sensors are chosen optimally using QR pivoting. Using the pre-constructed basis and the obtained field measurements, a sparse representation in the location domain is computed using $\ell_1$ optimization. Based on the resulting sparse representation, the location and magnetic moment of the magnetic dipole are estimated. An extension to an iterative method is implemented, where the basis and chosen sensors improve after every location estimate. Numerical simulations have been performed to verify the algorithm, and experiments have been done for validation. The proposed algorithm is shown to be effective in localizing magnetic dipoles.**




***Index Terms*— Compressed sensing, magnetic anomaly detection, magnetic sensors, sensor systems and applications.**

## I. Introduction

MAGNETIC Anomaly Detection (MAD) aims to detect, localize and identify magnetic anomalies that are typically hidden underwater or underground, using a wide variety of techniques. The target object creates a magnetic anomaly which can be measured using magnetic field sensors. For targets in the far field, magnetic anomalies can be modeled as a magnetic point dipole, of which the location and magnetic dipole moment can be estimated. In the field of MAD, various approaches are possible. Some are based on gradiometry, where multiple sensors are combined in one sensor configuration [1]. In [2], a method is obtained to solve location and moment from multiple gradiometric measurements. Other work describes the estimation of magnetic dipole parameters by measuring with a gradiometer above the target in a straight line [3], [4].

Manuscript received 30 March 2022; revised 31 May 2022; accepted 3 June 2022. Date of publication 27 June 2022; date of current version 1 August 2022. The associate editor coordinating the review of this article and approving it for publication was Prof. Pai-Yen Chen. *(Corresponding author: Stefan L. de Gijsel.)*

Stefan L. de Gijsel and Reinier G. Tan are with the Netherlands Organisation for applied scientific research, 2509 JG The Hague, The Netherlands (e-mail: stefan.degijsel@tno.nl; reinier.tan@tno.nl).

Aad R. P. J. Vijn is with the Delft Institute of Applied Mathematics, Delft University of Technology, 2600 AA Delft, The Netherlands, and also with the Netherlands Organisation for applied scientific research, 2509 JG The Hague, The Netherlands (e-mail: a.r.p.j.vijn@tudelft.nl).

Digital Object Identifier 10.1109/JSEN.2022.3184814

A different approach is to use a single magnetic field sensor. For either total field sensors or three-axis field sensors, estimation methods include using the magnetic gradient system [5], applying orthonormal basis function techniques [6], and minimum entropy or stochastic resonance algorithms [7].

In many problems, a non-linear optimization is required, since location of a magnetic dipole is incorporated in a non-linear way in its field description. Global optimization techniques such as particle swarm algorithms [5] or simulated annealing [8] can be used.

A different approach other than optimization of a non-linear problem is to approximate the problem by a linear problem. In [9], a linear algorithm is derived for the non-linear problem. In this paper, compressed sensing [10] is applied to the magnetic source localization problem to linearize the problem. The solution of the obtained linear system approximates the solution of the original problem.

The mathematical background of compressed sensing was developed in the 1980s, with new optimization techniques to recover sparse signals. In recent years, the field has been growing quickly, thanks to the development of efficient algorithms, growing computing power, and the establishment of solid mathematical foundations. Today, compressed sensing techniques are applied in many important applications, ranging from imaging techniques in cameras, MRI and seismology to radar and communication networks [11].

The structure of this paper is as follows. Section II gives an introduction to compressed sensing and discusses its main

aspects used in application to the magnetic dipole localization problem. Section III provides a problem description and a discussion of the implemented localization algorithm. In Section IV, a numerical example of the algorithm is given. Section V provides the setup for the experiments done to verify the algorithm, and Section VI discusses the results of these experiments. Finally, Section VII provides conclusions on the research.

## II. Compressed Sensing

Compressed sensing is the technique of recovering a signal using a limited number of measurements [10]. With this method, fewer measurements are needed than prescribed by classical information theory, such as the famous Shannon-Nyquist theorem [12]. The two conditions needed to apply compressed sensing are sparsity and incoherence.

For sparsity, the measured signal must be sparse in some basis. For a signal that can be represented in a sparse way it is possible to write

$$\mathbf{x} = \mathbf{\Psi}\mathbf{s}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^m$ is a signal, or state of a system, $\mathbf{\Psi} \in \mathbb{R}^{m\times n}$ is a transform basis and $\mathbf{s} \in \mathbb{R}^n$ is a sparse vector. Compressed sensing works by first estimating $\mathbf{s}$ from a limited part of the measurement vector $\mathbf{x}$. An estimate of the original signal $\mathbf{x}$ can then be produced from $\mathbf{x}$.

Performing $p$ measurements of the state $\mathbf{x}$ is described by the multiplication of $\mathbf{x}$ by a measurement matrix $\mathbf{C} \in \mathbb{R}^{p\times m}$ containing rows with zeroes and one value of 1, resulting in a measurement vector $\mathbf{y} \in \mathbb{R}^p$:

$$\mathbf{y} = \mathbf{C}\mathbf{x}. \tag{2}$$

Using (1), the system becomes

$$\mathbf{y} = \mathbf{C}\mathbf{\Psi}\mathbf{s} = \mathbf{\Theta}\mathbf{s}, \tag{3}$$

with $\mathbf{\Theta} \in \mathbb{R}^{p\times n}$. Now, $\mathbf{C}$ selects rows from $\mathbf{\Psi}$ that correspond to the chosen measurements, and these rows end up in in $\mathbf{\Theta}$.

The second condition for compressed sensing is incoherence of the measurement matrix $\mathbf{C}$ with respect to the basis $\mathbf{\Psi}$. The coherence is given by

$$\mu(\mathbf{C}, \mathbf{\Psi}) = \sqrt{n}\max_{j,k} \left|\langle \mathbf{c}_k, \boldsymbol{\psi}_j \rangle\right| \tag{4}$$

where $\mathbf{c}_k$ is the $k^{\text{th}}$ row of $\mathbf{C}$ and $\boldsymbol{\psi}_j$ is the $j^{\text{th}}$ column of $\mathbf{\Psi}$ [13]. The coherence is a measure for how correlated the rows of $\mathbf{C}$ are with the columns of $\mathbf{\Psi}$, with a value of 1 corresponding to incoherence, and a value of $\sqrt{n}$ to full coherence. The more incoherent the system is, the less measurements are needed for the compressed sensing algorithm to converge to the sparsest solution with high probability [13]. In Section IV and Section V, the coherence values in the performed experiments are given.

The system (3) is underdetermined. The sparsest solution can be found by optimization in the $\ell_0$ pseudo-norm, which usually requires a brute-force search. In compressed sensing, the $\ell_0$ optimization is replaced by an $\ell_1$ optimization, which is convex [13]. For convex $\ell_1$ optimization, many good solvers are available.

For compressed sensing techniques to be powerful in real-life applications, they need to be able to handle noisy data. Consider a measurement $\mathbf{y}$ that is noisy:

$$\mathbf{y} = \mathbf{\Theta}\mathbf{s} + \mathbf{e}, \tag{5}$$

where $\mathbf{e} \in \mathbb{R}^n$ is an error term. Now the condition $\mathbf{y} = \mathbf{\Theta}\mathbf{s}$ cannot hold exactly, and therefore a relaxed condition is considered:

$$\text{Minimize } \|\mathbf{s}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{\Theta}\mathbf{s}\|_2 \leq \epsilon. \tag{6}$$

Here, $\epsilon$ is some error bound related to the noise level of $\mathbf{e}$. This optimization problem is convex, since both $\|\mathbf{s}\|_1$ and $\|\mathbf{\Theta}\mathbf{s} - \mathbf{y}\|_2 - \epsilon$ are convex functions. Therefore, every local minimum of the problem is also a global minimum.

### A. Choosing an Error Bound

In the minimization problem (6), an error bound $\epsilon$ needs to be chosen. Both sensor noise and the effect of a discrete dipole location grid contribute to this value.

Based on the Gaussian white noise with standard deviation $\sigma$ as described in Section III-B.3, an appropriate value for the error bound $\epsilon$ is chosen. The random vector $\mathbf{X} := \mathbf{y} - \mathbf{\Theta}\mathbf{s}$ has uncorrelated elements with zero mean and variance $\sigma^2$. For a choice of $\epsilon$, it is important to know the characteristic of $||\tilde{\mathbf{X}}||_2$.

Scaling $\mathbf{X}$ results in $\mathbf{X} = \sigma\tilde{\mathbf{X}}$, where $\tilde{X}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. Note that $||\tilde{\mathbf{X}}||_2 = \sqrt{\sum_{i=1}^p \tilde{X}_i^2}$ is distributed according to the chi distribution: $||\tilde{\mathbf{X}}||_2 \sim \chi_p$. This leads to the following mean and variance for $||\mathbf{X}||_2$:

$$\mu_{\mathbf{X}} := \mathrm{E}||\mathbf{X}||_2 = \sigma\mathrm{E}||\tilde{\mathbf{X}}||_2 = \sigma\sqrt{2}\frac{\Gamma((p+1)/2)}{\Gamma(p/2)}, \tag{7}$$

$$\sigma_{\mathbf{X}}^2 := \mathrm{Var}||\mathbf{X}||_2 = \sigma^2\mathrm{Var}||\tilde{\mathbf{X}}||_2 = \sigma^2(p - \mathrm{E}||\tilde{\mathbf{X}}||_2^2). \tag{8}$$

The gamma function fraction in (7) can be approximated using Stirling's gamma function approximation [14]:

$$\sqrt{2}\frac{\Gamma((p+1)/2)}{\Gamma(p/2)} = \sqrt{p}\left(1 - \frac{1}{4(p+1)} + \mathcal{O}\left(\frac{1}{p^2}\right)\right). \tag{9}$$

From this, the following approximate estimated value and variance are obtained:

$$\mu_{\mathbf{X}} \approx \sigma\sqrt{p}\left(1 - \frac{1}{4(p+1)}\right), \tag{10}$$

$$\sigma_{\mathbf{X}}^2 \approx \sigma^2\left(p - p\left(1 - \frac{1}{2(p+1)} + \frac{1}{16(p+1)^2}\right)\right) = \sigma^2 p\frac{8p+7}{16(p+1)^2}. \tag{11}$$

The chi distribution approaches the shape of a normal distribution for small values of $p$ already. In this paper, it is expected that the value of $p$ is sufficiently large, and therefore the distribution can be approximated by a Gaussian distribution. Hence, it is assumed that more than 97 percent of values is smaller than the mean plus two standard deviations:

$$P(||\mathbf{X}||_2 \leq \mu_{\mathbf{X}} + 2\sigma_{\mathbf{X}}) > 0.97. \tag{12}$$

The upper bound of this interval is

$$\mu_{\mathbf{X}} + 2\sigma_{\mathbf{X}} \approx \sigma\sqrt{p}\left(1 - \frac{1}{4(p+1)} + 2\frac{\sqrt{8p+7}}{4(p+1)}\right) \approx \sigma\left(\sqrt{p} + \sqrt{2}\right), \quad (13)$$

for sufficiently large $p$. Hence, $\epsilon = \sigma\left(\sqrt{p} + \sqrt{2}\right)$ is chosen as error upper bound if just sensor noise is present. As will be discussed further in Section IV-A, an additional error component due to grid discretization will be taken into account.

### B. Characterizing and Solving the Minimization Problems

Consider again the minimization problem (6). It can be rewritten as a linear program with quadratic constraints (QCLP):

$$\min_{\mathbf{t},\mathbf{s}} \mathbf{1}^T\mathbf{t} \quad (14)$$

$$\text{s.t. } \mathbf{y}^T\mathbf{y} + \mathbf{s}^T\mathbf{\Theta}^T\mathbf{\Theta}\mathbf{s} - 2\mathbf{y}^T\mathbf{\Theta}\mathbf{s} \leq \epsilon^2 \quad (15)$$

$$c_i s_i \leq t_i \forall i \quad (16)$$

$$c_i s_i \geq -t_i \forall i. \quad (17)$$

The problem is convex, so it can be solved using efficient, existing convex optimization algorithms. The implementation in this research uses the CVX package [15], [16] for MATLAB with solver SDPT3, version 4.0 [17], [18], which implements interior point methods for SDP problems.

### C. Optimal Sensor Choice

It is assumed that a limited number of measurements is available. In the system (3) the condition number $\kappa(\mathbf{\Theta})$ determines how well the sparse solution $\mathbf{s}$ can be approximated from some noisy measurement $\mathbf{y}$. A system of equations with a smaller condition number is numerically more stable, which makes solving the system more robust. For a given number of available sensors, the goal is to minimize this condition number by choosing an appropriate measurement matrix $\mathbf{C}$.

To minimize the condition number, QR decomposition with column pivoting is used in combination with a Singular Value Decomposition (SVD) [19], [20]. QR decomposition with column pivoting is applied to $\mathbf{\Psi}^T$:

$$\mathbf{\Psi}^T\mathbf{P} = \mathbf{Q}\mathbf{R}, \quad (18)$$

where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is unitary, $\mathbf{R} \in \mathbb{R}^{n\times m}$ is upper triangular, and $\mathbf{P} \in \mathbb{R}^{m\times m}$ is a permutation matrix which permutes the columns of $\mathbf{\Psi}^T$ [21]. The ordering of rows of $\mathbf{P}$ determines the most significant measurements. The first rows of $\mathbf{P}$ form the measurement matrix $\mathbf{C}$.

The above decomposition is only applicable if $\mathbf{\Psi}$ is square. In the case where $\mathbf{\Psi}$ has more rows than columns, the condition number of $\mathbf{\Theta}^T\mathbf{\Theta}$ needs to be minimized [13]. However, in most cases $\mathbf{\Psi}$ has more rows than columns. In that case, this method is not directly applicable. A solution to that problem is using a reduced basis, as explained in Section II-D.

### D. Basis Reduction

In the case of undersampling, a model order reduction is required, which is performed by using the SVD. $\mathbf{\Psi}$ is replaced by a rank-$r$ approximation, where $r < \text{rank}(\mathbf{\Psi})$. A reduction of the $m \times n$ matrix $\mathbf{\Psi}$ to $r$ modes is obtained by performing an SVD:

$$\mathbf{\Psi} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (19)$$

where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ are unitary matrices, and $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is diagonal, with singular values $\sigma_i$ placed on the diagonal in descending order. Reduced versions of the matrices are created, where $\tilde{\mathbf{\Sigma}}$ contains the upper left $r \times r$ block of $\mathbf{\Sigma}$, and $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ the first $r$ columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. Then the rank-$r$ approximation $\mathbf{\Psi}_r$ is computed by

$$\mathbf{\Psi}_r = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T. \quad (20)$$

The Eckart-Young-Mirsky theorem states that $\mathbf{\Psi}_r$ is indeed the best rank-$r$ approximation to $\mathbf{\Psi}$ [22].

How well $\mathbf{\Psi}_r$ approximates $\mathbf{\Psi}$ is quantified by looking at the singular values corresponding to each of the columns of $\mathbf{\Psi}_r$. The energy each of these modes contains is represented by its singular value. The fraction of the energy $\eta$ of the first $r$ modes of $\mathbf{\Psi}$ is given by the sum of its singular values, divided by the total sum of all $n$ singular values:

$$\eta(r) = \frac{\sum_{i=1}^{r}\sigma_i}{\sum_{i=1}^{n}\sigma_i} \quad (21)$$

From the typical singular value distribution in Fig. 4, it becomes clear that the first few dozen modes already contain a very large part of the total energy. It is assumed that in this case sensors resulting from QR pivoting using $\mathbf{\Psi}_r$ are also optimal for representing the original basis $\mathbf{\Psi}$ [13].

## III. Magnetic Dipole Localization Algorithm

We propose a magnetic dipole localization algorithm that is able to estimate the location and moment of a single magnetic dipole in a search space $\Omega$. A magnetic dipole is defined by its center location $\mathbf{r}_c$ and its magnetic moment $\mathbf{m}$ [Am$^2$]. It has a magnetic field $\mathbf{B}_{\text{dip}}$ [T] described by

$$\mathbf{B}_{\text{dip}}(\mathbf{r}) = \frac{\mu_0}{4\pi}\left(\frac{3\hat{\mathbf{r}}(\mathbf{m}\cdot\hat{\mathbf{r}}) - \mathbf{m}}{r^3}\right), \quad (22)$$

where $\mathbf{r}$ [m] is the location where the field is measured, $r$ [m] is the distance between $\mathbf{r}$ and $\mathbf{r}_c$, $\hat{r}$ is the unit vector pointing from $\mathbf{r}_c$ to $\mathbf{r}$, and $\mu_0$ [H/m] is the vacuum permeability.

### A. Problem Description

In the magnetic dipole localization problem, a single magnetic dipole is considered. It has an unknown location in a search space $\Omega$, and an unkown magnetic moment. The goal of the implemented algorithm is to estimate the location and moment of this dipole using simulated measurements of a limited number of optimally chosen sensors. The problem is illustrated in Fig. 1.

The sparseness that is needed for compressed sensing is present in the location domain - it is assumed that there is only one dipole. It makes sense to think of the vector $\mathbf{s}$ in system (3) as representing unique locations and moments of single dipoles. An arbitrary dipole could then be represented by an 'average' of a few dipoles. This interpretation of the sparse

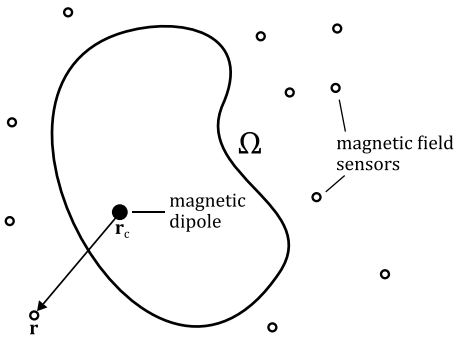



Fig. 1. Illustration of the localization problem, showing the search space $sf\,\Omega$ containing one magnetic dipole with center location $r_c$, and magnetic field sensors outside $sf\,\Omega$. The location of a field measurement is denoted by $r$.

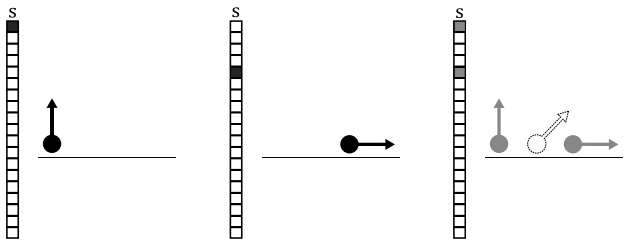



Fig. 2. The interpretation of the sparse vector $s$. The left and middle vector represent two different dipoles, and from the right vector, an 'average' dipole (dotted line) can be reconstructed.

vector is illustrated in Fig. 2. The location is incorporated into the magnetic dipole field description in a non-linear way. Describing the magnetic dipole by a weighted average of other magnetic dipoles in a grid allows for solving a linear system of equations. Section III-B.1 describes how the basis $\mathbf{\Psi}$ is constructed.

The assumption here is that the field of a dipole can be well represented as an average of the fields of other dipoles close by. A grid discretization error will be present, and an analysis on this error was performed. In a simulated test, four random magnetic dipoles were placed at the corners of a square with each time a different distance between them. The combined field of these dipoles was computed, as well as the field of the single dipole with location and moment calculated as in (25). The fields were calculated at different heights: the further away from the source(s), the better the separate dipole fields blend together to one field.

### B. Description of the Algorithm

The main contribution of this paper is described in this section. A compressed sensing algorithm for estimating field, location and moment of a dipole has been implemented. The algorithm must first be initialized by determining the basis $\mathbf{\Psi}$, and then follows several steps to produce an estimate. This section describes both the initialization and the different steps.

The main algorithm can be summarized by the following steps, given that just a selection of $k$ optimal sensors of the array can be used. The steps are also visualized in a flow diagram in Fig. 3.

1) Initialize basis $\mathbf{\Psi}$.
2) Choose $k$ optimal sensors using QR pivoting to perform measurements. This results in a measurement matrix $\mathbf{C}$. Multiplying by the initialized basis $\mathbf{\Psi}$ gives the required matrix $\mathbf{\Theta}$.
3) Perform a measurement $\mathbf{y}$ of size $3k$ from these chosen sensors.
4) Solve the $\ell_1$ optimization problem (6).
5) Classification: compute an estimated location and moment from the resulting sparse vector $\mathbf{s}$. A reconstructed dipole field also follows.

The upcoming subsections describe the basis initialization and all steps of the algorithm in more detail.

*1) Basis Initialization:* For a compressed sensing algorithm to work, a basis $\mathbf{\Psi}$ must be found that allows for transformation between the original and sparse domain of the signal. To do this, a set of initialization locations has been chosen, covering the search space $\mathbf{\Omega}$, as described in Section III-A. Specifically, a predefined 3D grid of locations covering $\mathbf{\Omega}$ is defined.

Every entry of the sparse vector $\mathbf{s}$ corresponds to a single dipole at one of the initialization locations, with moments of 1 $\text{Am}^2$ in positive and negative $x$, $y$, and $z$ directions. Every one of these dipoles creates a unique field at the sensors. The signal $\mathbf{x}$ in the compressed sensing system (3) is hence defined as the magnetic field $\mathbf{B}$ at the locations of all sensors in the sensor array, ordered as $[B_{x,1}, B_{y,1}, B_{z,1}, B_{x,2}, \ldots, B_{z,n_{\text{total}}}]$, with $1, 2, \ldots, n_{\text{total}}$ representing the $n_{\text{total}}$ sensors available. A measurement $\mathbf{y}$ contains a selection of this field, with just the $x$-, $y$-, and $z$-component of selected sensors. Hence, $\mathbf{C}$ is a selection matrix, containing selected rows from the identity matrix. The entries of $\mathbf{s}$ correspond to unique magnetic dipoles. The measurements used for initialization of $\mathbf{\Psi}$ are noise-free calculations using (22).

*2) Sensor Choice:* It is assumed that just $k$ of the total $n_{\text{total}}$ sensors are available to perform measurements. Every sensor measures three components of the field, giving a total of $p = 3k$ measurements. The sensors are chosen according to the method described in Section II-C and Section II-D. It must be noted that the resulting pivots from the QR algorithm indicate specific measurement indices, not sensors. Therefore, sensors have been chosen in order of appearance in the pivot sequence of one of their components.

*3) Performing Measurements:* The $k$ chosen sensors measure the three components of the magnetic field. All measurements are stored in a vector $\mathbf{y} \in \mathbb{R}^{3k}$.

*4) $\ell_1$ Optimization:* An $\ell_1$ optimization is performed to find a sparse solution vector. In the optimization problem (6) the 1-norm of $\mathbf{s}$ is minimized, and therefore the algorithm would prefer activating multiple smaller moments close to the sensor array instead of activating the expected dipole moments near the actual location. To resolve this undesirable behavior, a penalty vector $\mathbf{c} \in \mathbb{R}^n$ is introduced. Since $B \propto \frac{1}{r^3}$ where $r$ is the distance between the dipole and the sensor, a sensible choice is to make the penalty of a dipole location also dependent on $\frac{1}{r^3}$. Section IV-A describes how this is done in the specific case of a horizontal sensor array. The resulting convex minimization problem including penalty vector is

$$\text{Minimize } ||\mathbf{c} \odot \mathbf{s}||_1 \quad \text{s.t.} \quad ||\mathbf{y} - \mathbf{\Theta}\mathbf{s}||_2 \leq \epsilon, \tag{23}$$

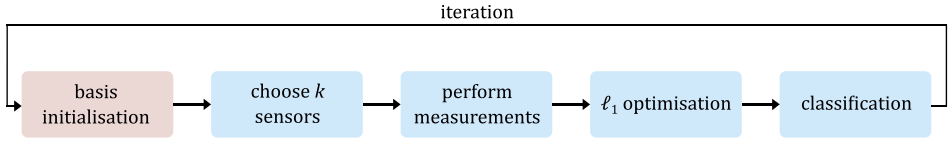



Fig. 3. Flow diagram of the algorithm steps.

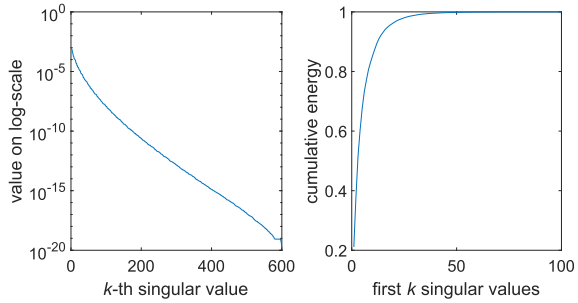



Fig. 4. Left: singular values in descending order. Right: the cumulative energy of the first 100 singular values.

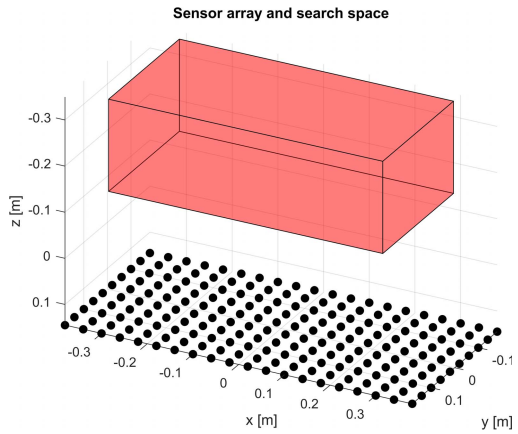



Fig. 5. An illustration of the setup, where the black dots indicate the sensors of the array, and the red box indicates the space in which the dipole is located. Note that the convention of the $z$-axis pointing downwards is used.

where $\odot$ denotes the Hadamard product, i.e. elementwise multiplication. The error bound $\epsilon$ is chosen as in Section II-A. If a grid discretization error with standard deviation $\sigma_{\text{grid}}$ is present, the error bound is taken as

$$\epsilon = \sqrt{\sigma^2(p + 2\sqrt{2p} + 2) + \sigma_{\text{grid}}^2 p}. \tag{24}$$

*5) Classification: Estimating Dipole Location and Moment:* To estimate the dipole's location and magnetic moment, weighted averages are taken. The sparse vector elements are used as weights to average over the initialization locations and moments. For the magnetic moment, the $n$ elements of $\mathbf{s}$ can be taken as weights. For the location, the same method is used, but one must take a grid discretization error into account. The estimated location $\hat{\mathbf{x}}$ and moment $\hat{\mathbf{m}}$ are given by:

$$\hat{\mathbf{x}} = \sum_{i=1}^{n} s_i \mathbf{x}_i; \quad \hat{\mathbf{m}} = \sum_{i=1}^{n} s_i \mathbf{m}_i, \tag{25}$$

where $\mathbf{x}_i$ and $\mathbf{m}_i$ are the location and moment corresponding to dipole $i$ in the initialization, respectively. Fig. 2 has illustrated this method of averaging.

### C. Extension of the Algorithm Using Iterations

To improve the estimates, the algorithm steps can be applied iteratively. This has been implemented by shrinking the search space. Based on the previous location estimate, a new basis is constructed as in Section III-B.1. Instead of using the full search space to choose initialization locations from, the search space is shrunk around the previous location estimate. Using the new basis, the same steps as in Section III-B are followed, using $k$ new sensors, and a new estimate is produced. Section IV-A describes the specific implementation in the numerical example provided in this paper.

## IV. Numerical Example

In this section, a numerical example of the algorithm is given. First, the setup and specific implementation for this example are described. Several results of a twin experiment for the implemented localization algorithm are given, where the influence of important variables and parameters in the algorithm is analyzed.

### A. Setup and Implementation

The considered setup is as follows: the search space spans from $x = -0.3$ m to 0.3 m, from $y = -0.15$ m to 0.15 m and from $z = -0.35$ m to $-0.15$ m. The sensor array is placed at $z = 0.145$ m and sensors are evenly spaced between $x = -0.38$ m and 0.38 m and between $y = -0.18$ m and 0.18 m, with a distance of 4 cm between sensors in both directions. An illustration of this setup is given in Fig. 5.

In each run of the twin experiment, a magnetic dipole with magnetic moment $\mathbf{m}$ is placed at a location $\mathbf{x}$. The formula of a magnetic dipole field (22) is used for simulating measurements. To every component $(x, y, z)$ of every sensor measurement some Gaussian noise is added, independent and identically distributed with a standard deviation $\sigma$. The algorithm produces an estimate $\hat{\mathbf{m}}$ of the magnetic moment, and $\hat{\mathbf{x}}$ of the location. The location error is computed by $\|\hat{\mathbf{x}} - \mathbf{x}\|_2$, moment magnitude error by $\frac{\|\hat{\mathbf{m}} - \mathbf{m}\|_2}{\|\mathbf{m}\|_2}$, and moment angle error is defined as the angle between $\hat{\mathbf{m}}$ and $\mathbf{m}$. An example of a field reconstruction using $k = 8$ sensors is shown in Fig. 6. In this configuration, the mutual coherence $\mu(\mathbf{C}, \mathbf{\Psi})$

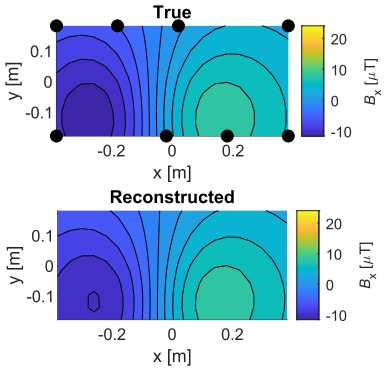



Fig. 6. Comparison between the *x*-components of the true magnetic field and field reconstruction in the twin experiment, both in nT. 8 sensors are chosen optimally - their locations are denoted by black dots.

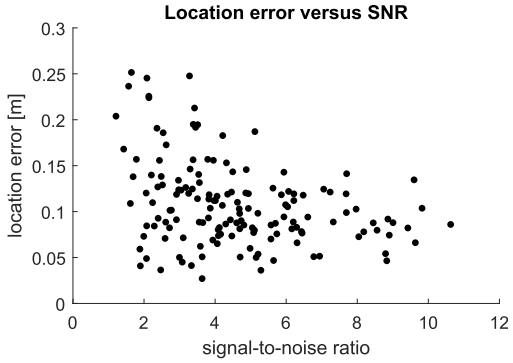



Fig. 7. Scatter plot of location errors versus average signal-to-noise ratio, for 150 numerical simulation runs with 8 chosen sensors.

is equal to 7.7, where $\sqrt{n} = 67.9$. The computation time on a machine with an $8^{\text{th}}$ generation Intel i7 processor is about 1.6 seconds for a single iteration, and about 2.5 seconds for three iterations.

As described in Section III-A, an analysis of the grid discretization error was made. The search space is located approximately 30 cm from the sensor array in this numerical experiment. At this height, a choice of 0.04 m grid distance ensures negligible influence of the grid discretization on the constructed magnetic dipole fields.

For the penalty vector $\mathbf{c}$ described in Section III-B.4 values are chosen equal to $\frac{1}{d^3}$, where $d$ is the distance from that specific magnetic dipole to the closest sensor.

For iterations, a smaller cube-shaped volume is used as updated search space. The center of this cube is the estimated location from the first iteration. The search cube in this implementation starts with an edge length equal to eight times the grid distances between the initial dipoles, hence containing at most $8^3$ dipole locations. Every iteration, the search radius shrinks by the grid distance, eliminating in each dimension two rows of grid points that were considered in the previous iteration. In this algorithm, iteration stops after a predefined number of iteration steps, or if the search box would become smaller than twice the grid distance. A different approach could be to stop iterating if the distance between two consecutive location estimates becomes smaller than a certain value.

### B. Noise

It is expected that a larger signal-to-noise ratio (SNR) makes it harder to reconstruct the noise-free field. For 150 runs, the individual location errors have been determined. The realizations are made with $k = 8$ chosen sensors. These are plotted against the average SNR in Fig. 7. It is clear that the average error, as well as the maximum encountered error, decreases for larger SNR, as expected. For a SNR slightly larger than 1, the average location error is still reasonable. In the presence of noise the algorithm can therefore still give an approximation of the location.

### C. True Dipole Location

To see if the true dipole location influences the results, the scatter plot in Fig. 7 has been reproduced to show the influence of three parameters: vertical distance of the true dipole to the sensor array, horizontal distance of the true dipole to the origin, and distance of the true dipole to the closest initialization grid point.

In Fig. 8a, the vertical distance to the sensor array is analyzed. Naturally, magnetic dipoles closer to the sensor array cause a larger SNR. Comparing results for equal SNR, there is no difference in location error between magnetic dipoles far away and close to the sensor array. This indicates that adding the penalty vector to the minimization problem successfully eliminates the preference for dipoles close to the sensor array.

In Fig. 8b, a comparison is shown for horizontal distance of the true dipole to the origin, which is the center of the search space. The algorithm does not perform better or worse for dipoles closer to the edges.

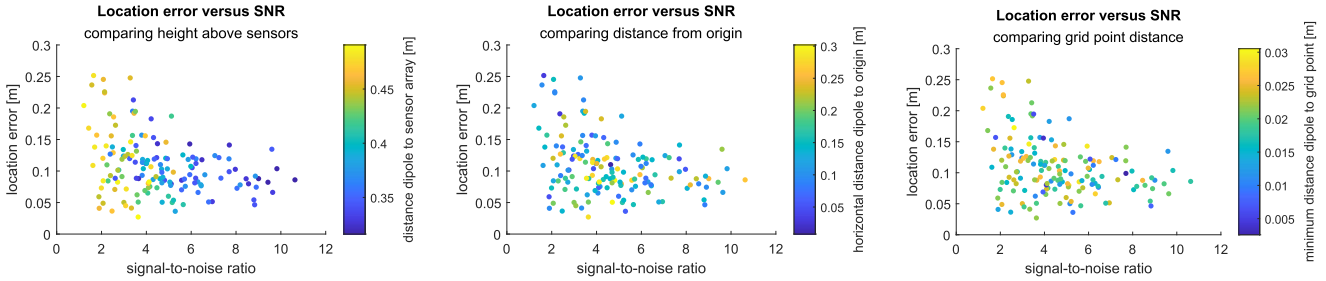
The last considered factor is how close the true dipole is to one of the initialization locations (grid points). The results are shown in Fig. 8c. Again, no clear distinction in location error is visible between true dipoles close to a grid point and dipoles further away.

### D. Iterations

To illustrate how iterations help in finding the correct location, the error distributions for 1, 2, and 3 iterations over 150 runs have been plotted in Fig. 9. The realizations are made with $k = 8$ chosen sensors, including noise with 0.5 $\mu$T standard deviation. The first iteration is completed using the full basis, over the full search space. Starting from the second iteration, the size of the search space is reduced as described in Section III-C.

The average location error is gradually reduced, as expected. By eliminating the possibility to activate dipoles too far away from the estimated location, the estimate becomes more precise. Furthermore, new sensors are chosen using QR pivoting in every iteration, based on the smaller bases. The activated sensors concentrate towards the estimated location every iteration, to better describe the field originating from a dipole in that area.

The shrinking of the search space is done gradually: although the estimate is already quite good on average after the first iterations, the number of times the estimate is relatively far away from the real location is not negligible. If the box is

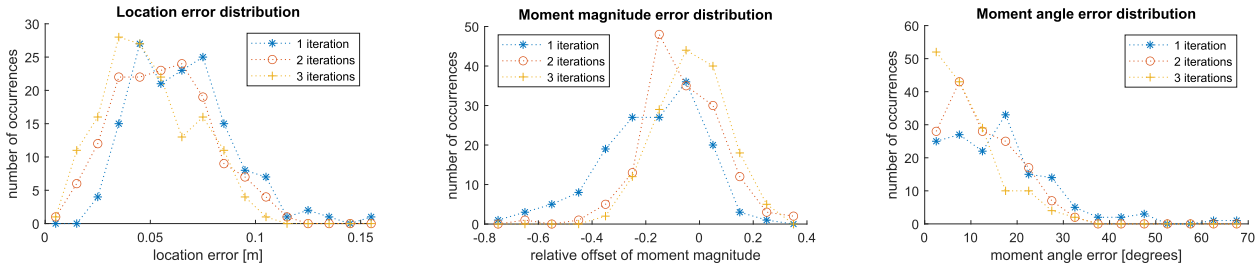



(a) Vertical distance between dipole and sensor array

(b) Horizontal distance between dipole and the origin

(c) Distance between dipole and closest initialization grid point

Fig. 8. Scatter plots of location errors versus signal-to-noise ratio, for 150 numerical simulation runs with 8 chosen sensors. Colors indicate dipole properties.




(a) Location errors

(b) Moment magnitude errors

(c) Moment angle errors

Fig. 9. Distributions of errors for 150 experimental runs and 8 chosen sensors, comparing 1, 2, and 3 iterations.

shrunk too aggressively, the real dipole location might end up outside the search box, eliminating the chances of finding this true location. Therefore, to create a robust algorithm, only one layer of initial dipoles is removed on each side of the search box every iteration.

On average, the moment magnitude is estimated well, staying within 10 percent above or below the true value. It is interesting to note the structural error - although not too large - in moment angle. This error is by definition always positive, so it might match with the average structural error in location. Naturally, these influence each other: if the dipole location estimate contains an error, the algorithm must have compensated for this by rotating the estimated dipole bit, to still be able to match the resulting field with the obtained measurement.

## V. Experimental Setup

The experimental setup is similar to the one provided in the problem description in Section IV-A, and the illustration in Fig. 5. The measurement setup also consists of a horizontally placed sensor array, now placed at $z = 0.045$ m. It spans between $x = -0.38$ m and 0.38 m and between $y = -0.18$ m and 0.18 m with 0.04 m distance between sensors in both directions, which makes 200 sensors and 600 measurements in total. The 3-axis magnetic sensors are of the type PNI RM3100. Since each field component is measured by a different physical sensor at a slightly different location, there can be an offset of a few mm in the sensor locations per component. As before, all sensor locations have been used in the initialization step to construct the basis. However, in the application of the algorithm sensor choice was simulated in this test using only the measurements of 10 chosen sensors. The data of the other 190 sensors was acquired, but not used in applying the algorithm.

A small magnet was moved over a grid in an S-shaped path. The grid was 9 grid positions wide and 7 grid positions long, from $x = -0.38$ m to $x = 0.38$ m and from $y = -0.18$ m to $y = -0.18$ m. The magnet that was used produces a field close to that of a perfect magnetic dipole, with an estimated magnetic moment of approximately 0.05 Am$^2$.

The sensor noise level has been estimated using two initial measurements of the background field. 200 sensors have each measured the three magnetic field components twice. From these measurements a standard deviation of $\sigma = 21$ nT is derived.

For basis construction, a grid distance of 0.04 m was used. Again, the grid discretization error needs to be estimated. The dipole was held at 0.24 m from the grid, and tests showed an average discretization error of approximately 25 nT. An adjusted value $\epsilon \approx 114$ nT is chosen according to (24), in which $\sigma$ and $\sigma_{\text{grid}}$ are taken equal to 21 nT and 25 nT, respectively. An alternative approach could be to decrease the grid distance in basis construction. However, this increases both the size of stored matrices and the computing power needed to optimize exponentially.

In this configuration, the mutual coherence $\mu(\mathbf{C}, \mathbf{\Psi})$ is equal to 10.5, where $\sqrt{n} = 67.9$.

## VI. Experimental Results

The experimental runs have been conducted using the measurements of only 10 sensors at the same time. The results for 1 iteration of the algorithm are compared to the results

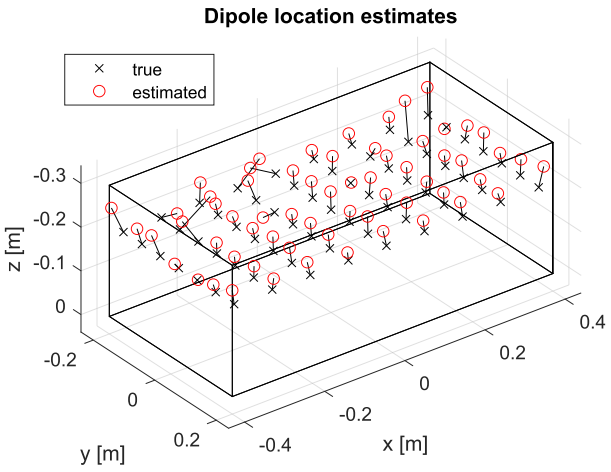



Fig. 10. 3D view of true locations compared to compressed sensing location estimates from experimental runs after 3 iterations.

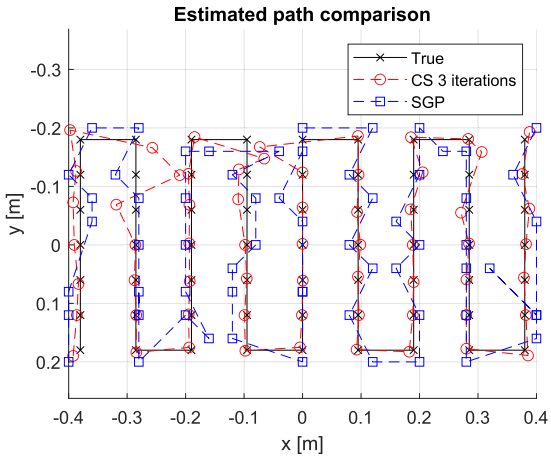



Fig. 11. Top view of true locations compared to location estimates from experimental runs, for the compressed sensing algorithm with 3 iterations and the single grid point method.

for 3 iterations. The magnetic moment was not measured or estimated independently from this algorithm, and therefore only location estimates and errors are discussed.

The resulting location estimates for 3 iterations are shown in Fig. 10. This plot shows the difference between the true magnetic dipole location and the location estimate. The horizontal location error is small, except for some outliers in the region where $y$ is negative. The outliers can be explained by the observation that in this region, the sensors produced a larger than average error. The location estimates can be influenced by these errors. Another related source of the problem could be the assumption of an equal noise level for every sensor.

It can be observed that estimates of the $z$-coordinate are too small. Since an $\ell_1$ optimization within error bounds is performed, every possibility is used to minimize the $\ell_1$ norm term. A large feasible region is needed to ensure a good estimate when the data is noisier than average. The introduction of a penalty term scaling with $\frac{1}{r^3}$ enables the algorithm to produce reasonable height estimates, but leaves room for improvement.

The performance of the algorithm is compared to the performance of an alternative algorithm: the single grid point (SGP) method. In this method, the measured field at 10 randomly chosen sensor locations is compared to the magnetic fields created by a dipole at each of the initialization locations (the grid points). For each of the grid points, it is calculated how well the measurement can be explained by three dipoles placed in $x$-, $y$-, and $z$-direction. The estimated location $\hat{r}$ is the location with the smallest residual:

$$\hat{r} = \arg\min_{r} \|\mathbf{y} - \mathbf{A}(r)\mathbf{b}\|_2 , \tag{26}$$

where $r$ are the grid point locations, $\mathbf{A}$ contains the magnetic fields at the chosen sensor positions of the three magnetic dipoles in $x$-, $y$-, and $z$-direction, and $\mathbf{b}$ is the least-squares solution of $\mathbf{Ab} = \mathbf{y}$.

In Fig. 11, a top view of the location estimates of both algorithms is shown, compared to the true path. The compressed sensing algorithm performs slightly better than the SGP method. The location estimation results are summarized in Fig. 12. This figure also shows that the performance of the

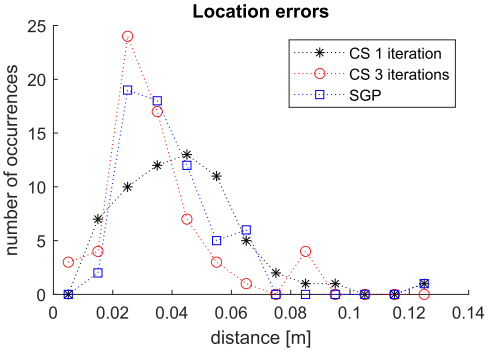



Fig. 12. Location estimate errors for compressed sensing with 1 and 3 iterations, and for the single grid point method.

algorithm improves when the number of iterations is increased from 1 to 3. These results are in line with those from the numerical example in Fig. 9.

## VII. Conclusion

Compressed sensing is an interesting method for processing relatively small amounts of data. An algorithm for localization of a magnetic dipole in a 3D space was implemented. The algorithm is able to make accurate predictions of location and dipole moment of a randomly placed dipole. This is shown by simulation results, and verified using real-world measurements using a small magnet.

As expected, increasing the SNR results in better location and magnetic moment estimates. Increasing the number of sensors chosen improves the results. The implemented algorithm features an option for iteration in combination with optimal sensor choice using QR pivoting. Iterating has shown to be indeed able to improve localization performance. Analysis shows that there are no biases in the algorithm towards certain source dipole locations.

In future research, the influence of sensor position should be analyzed. For instance, variation in height of the sensor layer or different distributions of the sensors can be investigated. Furthermore, variations of random sensor choice should

be investigated. These factors can influence the incoherence between the measurement matrix and the sparsifying basis.

## REFERENCES


[1] J. P. Rego and W. H. Cegielski, "Gradiometry survey and magnetic anomaly testing of Castros de Neixón, Galicia, Spain," *J. Archaeolog. Sci.*, vol. 46, pp. 417–427, Jun. 2014.

[2] W. M. Wynn, "Magnetic dipole localization with a gradiometer: Obtaining unique solutions," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, vol. 4. Singapore, Aug. 1997, pp. 1483–1485.

[3] M. Birsan, "Recursive Bayesian method for magnetic dipole tracking with a tensor gradiometer," *IEEE Trans. Magn.*, vol. 47, no. 2, pp. 409–415, Feb. 2011.

[4] M. Birsan, "Inversion of magnetic dipole parameters using a scalar field gradiometer," *IEEE Sensors J.*, vol. 21, no. 6, pp. 7434–7438, Mar. 2021.

[5] L. Fan *et al.*, "Tracking of moving magnetic target based on magnetic gradient system with total field magnetometers," *Sensor Rev.*, vol. 38, no. 4, pp. 501–508, Sep. 2018.

[6] B. Ginzburg, L. Frumkis, and B. Z. Kaplan, "Processing of magnetic scalar gradiometer signals using orthonormalized functions," *Sens. Actuators A, Phys.*, vol. 102, nos. 1–2, pp. 67–75, Dec. 2002.

[7] C. Wan, M. Pan, Q. Zhang, F. Wu, L. Pan, and X. Sun, "Magnetic anomaly detection based on stochastic resonance," *Sens. Actuators A, Phys.*, vol. 278, pp. 11–17, Aug. 2018.

[8] A. Sheinker, B. Lerner, N. Salomonski, B. Ginzburg, L. Frumkis, and B.-Z. Kaplan, "Localization and magnetic moment estimation of a ferromagnetic target by simulated annealing," *Meas. Sci. Technol.*, vol. 18, no. 11, pp. 3451–3457, Sep. 2007.

[9] C. Hu, M. Q.-H. Meng, and M. Mandal, "A linear algorithm for tracing magnet position and orientation by using three-axis magnetic sensors," *IEEE Trans. Magn.*, vol. 43, no. 12, pp. 4096–4101, Dec. 2007.

[10] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[11] S. Qaisar, R. M. Bilal, W. Iqbal, M. Naureen, and S. Lee, "Compressive sensing: From theory to applications, A survey," *J. Commun. Netw.*, vol. 15, no. 5, pp. 443–456, Oct. 2013.

[12] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, no. 1, pp. 10–21, Feb. 1949.

[13] S. L. Brunton and J. N. Kutz, "Sparisty and compressed sensing," in *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2019, ch. 3, pp. 84–114.

[14] M. R. Spiegel, S. Lipschutz, and J. Liu, *Schaum's Outline of Mathematical Handbook of Formulas and Tables*, 3rd ed. New York, NY, USA: McGraw-Hill, 2008, p. 148.

[15] M. Grant and S. Boyd. (2014). *CVX: MATLAB Software for Disciplined Convex Programming*. [Online]. Available: http://cvxr.com/cvx

[16] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control* (Lecture Notes in Control and Information Sciences), vol. 371, V. Blondel, S. Boyd, and H. Kimura, Eds. London, U.K.: Springer, 2008, pp. 95–110.

[17] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3—A MATLAB software package for semidefinite programming," *Optim. Methods Softw.*, vol. 11, nos. 1–4, pp. 545–581, Jan. 1999.

[18] E. Clark, J. N. Kutz, and S. L. Brunton, "On the implementation and usage of SDPT3—A MATLAB software package for semidefinite-quadratic-linear programming," *Int. Ser. Oper. Res. Manag. Sci.*, vol. 166, pp. 715–754, Jan. 2012.

[19] E. Clark, J. N. Kutz, and S. L. Brunton, "Sensor selection with cost constraints for dynamically relevant bases," *IEEE Sensors J.*, vol. 20, no. 19, pp. 11674–11687, Oct. 2020.

[20] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns," *IEEE Control Syst.*, vol. 38, no. 3, pp. 63–86, Jun. 2018.

[21] Å. Björck, "Rank-deficient problems," in *Numerical Methods in Matrix Computations*, vol. 59. New York, NY, USA: Springer, 2015, ch. 2, pp. 291–316.

[22] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936.


**Stefan L. de Gijsel** received the B.Sc. degree in applied physics and the B.Sc. and M.Sc. degrees in applied mathematics from the Delft University of Technology. During his master's, he specialized in computational science and engineering. He is currently working with the Netherlands Organisation for applied scientific research.

**Aad R. P. J. Vijn** received the B.Sc. degree in applied physics and the B.Sc. and M.Sc. degrees in applied mathematics from the Delft University of Technology, where he received the Ph.D. degree in mathematical physics in 2021. During his master's and Ph.D., he specialized in computational science and engineering, numerical analysis, and electrodynamics. He is currently working with the Netherlands Organisation for applied scientific research. He is a Teacher with the Delft University of Technology.

**Reinier G. Tan** received the M.Sc. degree in astrophysics from Leiden University. In 2009, he joined the Radar Technology Group at the Netherlands Organisation for applied scientific research, working on novel radar processing techniques. In 2017, he joined the Department of Electronic Defence, where he works on magnetic sensing. Recently, his work is focused on new magnetic sensing applications using unmanned systems.