Particle Swarm Optimization With Rotation Axis Fitting for Magnetometer Calibration

BAGUS ADIWILUHUNG RIWANTO
TUOMAS TIKKA
ANTTI KESTILÄ
JAAN PRAKS, Member, IEEE
Aalto University, Espoo, Finland

This paper presents an improved multiobjective particle swarm optimization algorithm for magnetometer calibration in spacecraft. The proposed algorithm combines scalar checking with novel rotation axis fitting objective and avoids the requirement for perfectly aligned measurement axis. The improved approach is designed to solve 12 calibration parameters based on the knowledge of the magnetometer rotation axis direction. The performance of the novel algorithm is demonstrated with simulations and experimental data on Aalto-1 nanosatellite.

Manuscript received March 5, 2016; revised July 4, 2016 and October 24, 2016; released for publication October 26, 2016. Date of publication February 9, 2017; date of current version April 27, 2017.

DOI. No. 10.1109/TAES.2017.2667458

Refereeing of this contribution was handled by H. Schaub.

Authors' address: B. A. Riwanto, T. Tikka, A. Kestilä, and J. Praks are with the School of Electrical Engineering, Aalto University, Espoo 00076, Finland, E-mail: (bagus.riwanto@aalto.fi; tuomas.tikka@aalto.fi; antti.kestila@aalto.fi; jaan.praks@aalto.fi).

0018-9251 © 2017 OAPA

I. INTRODUCTION

Magnetometers are widely used in spacecraft as part of scientific payload or as the attitude sensor, with the latter application implemented by comparing the measured magnetic field vector with the modeled ambient magnetic field (e. g., magnetic field of Earth from international geomagnetic reference field (IGRF) model). However, magnetometer readings contain errors from various sources, e. g., nonorthogonality, misalignment, scale factor, hard iron, soft iron, bias, and measurement noise, which will affect the direction and magnitude of the magnetic field vector measured by the magnetometer [1]–[5]. Therefore, calibrating the magnetometers from those errors are important for the spacecraft to meet the performance requirements.

Various studies have proposed numerous magnetometer calibration methods, although attitude independent calibration using scalar checking method (also commonly referred as calibration in magnetic domain) [6], [7] are more common in spacecraft application because of its flexibility for on-orbit calibration [2], [4], [5], [8], [9]. Foster and Elkaim [2] adopted this approach and used two-step algorithm with least squares method to estimate intermediate parameters, which was then derived further to obtain the real calibration parameters, while Springmann and Cutler [5] improved the approach by introducing new time-varying components as a function of current measurements in the spacecraft electronics.

However, scalar checking method by itself imposes some limitations on the resolved calibration parameters: It only checks whether the magnitude of the magnetic field vector measured by the magnetometer matches the modeled ambient magnetic field [6], [7]. This means that the goal of this calibration approach is to calibrate the erroneous, ellipsoid-shaped measurement locus back into a sphere-shaped measurement locus as depicted in Fig. 1. Since no actual heading information is used as a reference, a rotational ambiguity exists in the solution—It can be understood easily since a calibrated sphere-shaped measurement locus can be rotated to arbitrary direction and still fulfill the scalar checking objective.

Various approaches have been documented in different studies in order to overcome this rotational ambiguity problem. Ali *et al.* [10] simplified the scaling matrix into a diagonal matrix (reducing the number of parameters to 6, including the offset vector), Elkaim [2] and Springmann and Cutler [5] assumed that one of the sensor axis is perfectly aligned with the reference axis, resulting in triangular scaling matrix (reducing the number of parameters to 9), while Gebre-Egziabher, Elkaim and Powell, [11] assumed that the misalignment coefficients are small, a condition used for initial estimate in two-step calibration method. All of them also neglected soft iron coefficients in nonaligned axes in order to eliminate the rotational ambiguity in scalar checking method. Renaudin *et al.* [3] proposed a calibration algorithm that minimizes those assumptions using adaptive least squares method. However, the results show that the algorithm is still sensitive to nondiagonal components of the

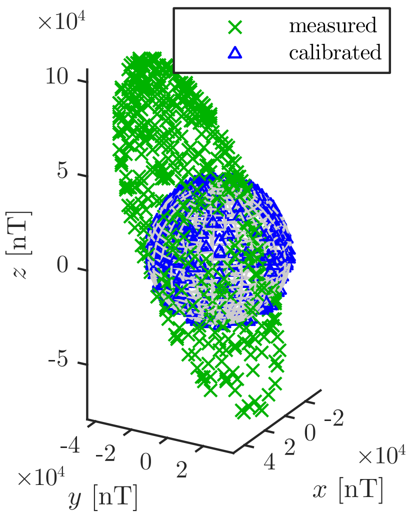


Fig. 1. 3-D plot of magnetic field vectors from a simulated erroneous (ellipsoid) and ideal (sphere) measurement locus. The axes represent the magnetic field strength in nT from the magnetometer frame. Note that measurement random noise is not included for viewing clarity.

total scale matrix associated with large errors from soft iron and misalignment factors. Liu *et al.* [12] proposed a technique to resolve the rotational ambiguity using the constant angle between the geomagnetic field vector and the gravity vector, although this is only useful for most land-based attitude sensor application where its location on Earth is relatively constant and gravity vector can be measured easily. Another approach is proposed in [13] by incorporating the pitch-roll state in a arbitrarily moving magnetometer to resolve the ambiguity problem, although this requirement means that the calibration algorithm is not fully attitude-independent.

This paper proposes a new approach in magnetometer calibration by combining the widely used scalar checking objective with a novel rotation axis fitting objective, which only requires additional knowledge of the magnetometer rotation axis instead of full knowledge of the system's attitude. The solution for these objectives, which is the set of 12 calibration parameters (3×3 matrix and 3×1 vector), is optimized using multiobjective particle swarm optimization (PSO) algorithm. This paper is presented in the following order: The definition of the calibration parameters from the magnetometer mathematical model is explained in Section II. The multiobjective PSO algorithm itself is explained in Section III, and the evaluation of the developed algorithm using simulated and experimental data is described in Section IV.

II. MATHEMATICAL MODEL OF MAGNETOMETER

In general, the relationship between the measured magnetic field vector $\check{\boldsymbol{b}}$ from the magnetometer and the reference ambient magnetic field $\boldsymbol{b}$ can be modeled as

$$\check{\boldsymbol{b}} = \boldsymbol{S}_m \boldsymbol{N}^{-1} \left(1 + \boldsymbol{C}_{\text{si}}\right) \left(\boldsymbol{b} + \boldsymbol{b}_{\text{hi}} + \mathbf{off}_m + \boldsymbol{\eta}_m\right) \qquad (1)$$

where $\boldsymbol{S}_m$ is the scale factor matrix that describes the proportionality of the input to the output for each axis; $\boldsymbol{N}$ is the nonorthogonality matrix that describes the true x-, y-, and z-axis direction of the sensor in the defined sensor frame; $\mathbf{off}_m$ is a vector that represents the sensor internal bias; $\boldsymbol{\eta}_m$ is a vector containing measurement noise; $\boldsymbol{b}_{\text{hi}}$ is the hard iron parameter vector modeled from permanent magnet dipoles in the environment (this error appears as a linear shift of the magnetic field lines); and $\boldsymbol{C}_{\text{si}}$ is the soft iron parameter matrix modeled from ferromagnetic materials that get induced by external magnetic field (i. e., the ambient magnetic field and hard iron disturbances) and, in effect, will distort the shape of the magnetic field lines. Other errors inherent to the instrument are also present, e. g., temperature fluctuation and cross-field effect, but these parameters are omitted from (1) for simplicity since they can be mathematically represented by the present parameters [3], [4], [14]. The model used in this paper also does not involve time-varying parameters.

It is important to note that most of the error parameters in (1) are not uniquely solvable if evaluated solely from scalar checking calibration process without any *a priori* knowledge on the magnetometer parameters or extensive testing procedure on individual component of the material surrounding the spacecraft. Thus, (1) can be simplified into

$$\check{\boldsymbol{b}} = \boldsymbol{S}_{m_c} \left(\boldsymbol{b} + \mathbf{off}_{m_c} + \boldsymbol{\eta}_m\right) \qquad (2a)$$

where 3×3 matrix $\boldsymbol{S}_{m_c}$ and 3×1 vector $\mathbf{off}_{m_c}$ are the compounded calibration parameters from the combination of $\boldsymbol{S}_m$, $\boldsymbol{N}$, $\boldsymbol{C}_{\text{si}}$, $\boldsymbol{b}_{\text{hi}}$, and $\mathbf{off}_m$ in (1). Note that in magnetometer calibration, the calibration parameters need to transform the measured magnetic field $\check{\boldsymbol{b}}$ into the corrected magnetic field $\hat{\boldsymbol{b}}$, flipping (2a) into

$$\hat{\boldsymbol{b}} = \boldsymbol{K}_m \check{\boldsymbol{b}} - \boldsymbol{k}_m \qquad (2b)$$

where the ambient magnetic field $\boldsymbol{b}$ is now substituted by the corrected magnetic field $\hat{\boldsymbol{b}}$, while the measurement noise $\boldsymbol{\eta}_m$ is assumed as zero-mean Gaussian random process. $\boldsymbol{K}_m = \boldsymbol{S}_{m_c}^{-1}$ and $\boldsymbol{k}_m = \mathbf{off}_{m_c}$ are the calibration parameters that need to be optimized by the calibration algorithm. With this definition, the scalar checking algorithm will serve to minimize the difference in the magnitude of the known reference field $\boldsymbol{b}$ and the corrected field $\hat{\boldsymbol{b}}$.

III. PSO ALGORITHM

Eberhart and Kennedy [15], [16] introduced PSO algorithm as nonlinear function optimization algorithm. Since then, PSO has been used in very wide range of field. Studies such as ones by Eberhart and Shi [17] and [18] documented various fields of study where PSO has been implemented. In this paper, Sections III-A and III-C describe the PSO for magnetometer scalar checking objective based on [10], [19], and [20], whereas Section III-D introduces the novel rotation axis fitting objective that is implemented in the multiobjective PSO.

A. Basic PSO Algorithm

Wu *et al.* [19] proposed using PSO algorithm for solving the parameters estimation problem in magnetometer calibration because of its better capability in finding global solution without a good initial estimate, and later optimized it with stretching PSO algorithm [20], whereas Ali *et al.* [10] optimized the standard PSO by combining it with a preliminary range of interest selection technique in order to reduce the number of iterations. So far, the PSO variations proposed were designed for optimizing the scalar checking objective only, while the rotational ambiguity is resolved by simplifying the calibration parameters in $\boldsymbol{K}_m$ from (2b) into a diagonal or triangular matrix, requiring assumptions that at least one sensor axis is perfectly aligned to the reference frame in order to reduce the number of unknown parameters from 12 to 9 or 6 parameters. Typical PSO algorithm contains the following iteration steps.

1) *Swarm initialization:* Initialize n_p numbers of particles (the *swarm size*) with random initial "position" $\boldsymbol{p}_i(k=0)$ and random initial "velocity" $\boldsymbol{v}_i(k=0)$ in an abstract n^c-dimensional space (the *swarm dimension*), or, written in mathematical notation:

$$\boldsymbol{p}_i(k)=[p_{i1}(k), p_{i2}(k), \ldots, p_{in^c}(k)] \equiv \left[p_{ij}|_{j=1,\ldots,n^c}(k)\right]$$
$$\boldsymbol{v}_i(k)=[v_{i1}(k), v_{i2}(k), \ldots, v_{in^c}(k)] \equiv \left[v_{ij}|_{j=1,\ldots,n^c}(k)\right]$$

where the subscript i is the particle index ($i=[1,n_p]\in\mathbb{Z}$), subscript j is the particle component index ($j=[1,n^c]\in\mathbb{Z}$), and k is the iteration index. The number of unknown parameters n_c corresponds to the swarm dimension n^c, i. e., $n^c = n_c$. For simplicity, n_c will refer to both the swarm dimension and number of parameters from here onwards. Each particle also has a *local* best position $\boldsymbol{pbest}_i$ that is defined as the position that will produce the best fitness in the search history of each particle i. For the initialization phase, the local best position is the initial position itself $\left(\boldsymbol{pbest}_i(k=0)\equiv \boldsymbol{p}_i(k=0)\right)$. In swarm-wide scope, a *global* best position $\boldsymbol{gbest}$ is defined as the best position in the whole swarm, i.e., the local best position of the particle with the best fitness.

2) *Swarm positions update:* The velocity and position of each particle are updated with the basic formula [21], [22]

$$\begin{aligned} v_{ij}(k+1) &= wv_{ij}(k) \\ &\quad + \text{rand}(0,1)c_1(pbest_{ij}(k)-p_{ij}(k)) \\ &\quad + \text{rand}(0,1)c_2(gbest_j(k)-p_{ij}(k)) \quad (3a) \\ p_{ij}(k+1) &= p_{ij}(k)+v_{ij}(k+1) \quad (3b) \end{aligned}$$

where rand(0,1) is a random number in the range of $[0,\ldots,1]\in\mathbb{R}$, w is the *inertia weight* parameter, c_1 is the *cognitive rate* parameter, and c_2 is the *social rate* parameter. The inertia weight parameter defines the particle tendency to stay on its original course, resisting influence from the global and local best position. The cognitive rate parameter defines the influence of the particle memory of its best location, whereas the social rate parameter defines the influence of the swarm global best position.

3) *Best solution evaluation:* $\boldsymbol{pbest}_i$ for each particle i is replaced by the new particle position if the fitness value is better than the last one. The problem in this paper deals with minimization problems; thus, the objective of the swarm is to minimize the fitness value, or mathematically written as

$$\boldsymbol{pbest}_i(k) = \arg\min_{\boldsymbol{p}_i} f(\boldsymbol{p}_i(1,\ldots,k)) \qquad (4a)$$

while, as previously explained, the global best position is obtained from the local best position with the best fitness among all particles, or, in mathematical notation

$$\boldsymbol{gbest}(k) = \arg\min_{\boldsymbol{pbest}_i} f(\boldsymbol{pbest}_i(k)) \qquad (4b)$$

where $f()$ is the fitness function that returns the fitness value

$$F_i = f(\boldsymbol{p}_i), \qquad \text{for } i=1,\ldots,n_p \qquad (4c)$$

with the notation for the fitness value of the local and global best positions (F_i and F_g, respectively) are written as

$$F_i \equiv f(\boldsymbol{pbest}_i) \qquad F_g \equiv f(\boldsymbol{gbest}). \qquad (4d)$$

After the local best position for each particle is evaluated with (4a), the global best position is then reevaluated with (4b): if a new local best position with a fitness value lower than the current global best position exists, then the global best position will be updated with the new, better position—else, the global best position stays the same.

4) *Iteration evaluation:* The algorithm will determine whether the swarm has reached its goal by checking several predefined conditions, e. g., the current global best fitness is lower than a certain threshold, the number of iterations has reached a certain number, or the swarm has converged into a certain value and the search space is not explored to a certain range. If all of the predefined conditions are not met, then the iteration will continue. On the other hand, if any of the condition (or a combination of them) is met, then the algorithm will stop and return the solution contained in its $\boldsymbol{gbest}$. In this paper, the algorithm ends when a set maximum number of iteration $k_{\max}$ has been reached.

B. PSO Architecture and Parameters

Beyond the basic PSO iteration process, several settings in the swarm topology and parameters are chosen for this paper, which are summarized in this section. More detailed description of the algorithm and additional test results can be found in [23].

1) *PSO Topology:* The PSO implemented in this paper is a *global topology* PSO, where every particle in the swarm can communicate with each other directly, thus enabling an instant propagation of $\boldsymbol{gbest}$ information across the swarm. In contrast, *local topology* PSOs are also available in different versions [24].

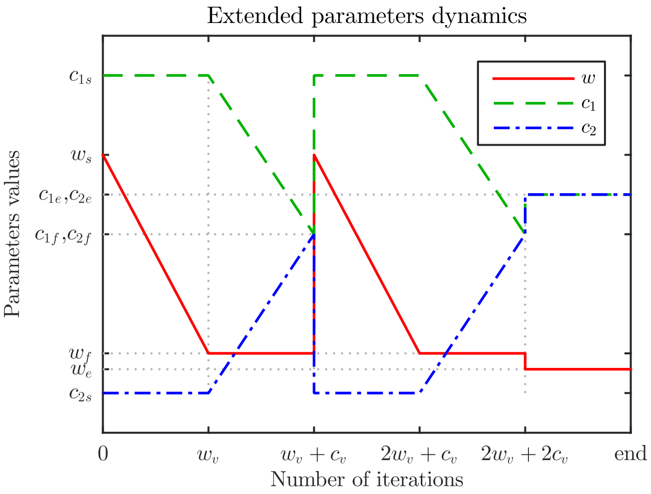


Fig. 2. Plot of PSO parameters values against the number of iterations, depicting the extended dynamic parameters variations implemented in this paper. This technique is used for promoting exploration of the search space and assisting the swarm in escaping local minima.

2) *PSO Main Parameters:* The main PSO parameters (i. e., w, c_1, and c_2) are implemented with a dynamic parameters adapted from [22]. The dynamics of the parameters values as the iteration progresses are depicted in Fig. 2. First, a gradually decreasing w early in the iteration is followed by gradually converging c_1 and c_2. The parameters values start from the starting values $w_s = 3, \ldots, 5$, $c_{1s} = 4$, and $c_{2s} = 0$, then vary for a certain period $w_v = c_v = (0.05, \ldots, 4)k_{\max}$, where $k_{\max}$ is the maximum number of iterations, until they reached the final value $w_f = 0.5$ and $c_{1f} = c_{2f} = 2$. Then, this pattern of w, c_1, and c_2 dynamics is repeated over the iterations to extend the exploration of search space, until they stabilized at the end value of $w_e = 0.382$ and $c_{1e} = c_{2e} = 2.5$. Those final and end values were analytically determined with the constriction factor rule [17], [21], [25], [26]. The purpose of this extended parameters dynamics is for promoting exploration of the search space earlier in the iteration, and later helping the swarm in escaping local minima as the swarm starts to stabilize [22].

3) *Number of Particles:* The effect of the number of particles n_p on the algorithm performance varies depending on the problem—One study recommends a guideline of 20–100 particles to produce optimal performance, although determining the optimal number of particles for a specific application remains a trial-and-error problem [21]. In this paper, the PSO runs on the swarm size of 30–60 particles.

4) *Initialization Condition:* The initialization condition is defined by the parameters $\boldsymbol{pmax}$ and $\boldsymbol{vmax}$. $\boldsymbol{pmax}$ is an n_c-dimension vector that defines the initial positions of the swarm, where each component of $\boldsymbol{pmax}$ represents the components of $\boldsymbol{K}_m$ and $\boldsymbol{k}_m$. For this paper, the initialization range for each component $\boldsymbol{pmax}_j$ is estimated around the realistic expected value of the solution, e. g., a range of $[-1, 1]$ for the components of $\boldsymbol{K}_m$ and $[-5000, 5000]$ nT for the components of $\boldsymbol{k}_m$. On the other hand, initialization of the swarm velocities is performed in a similar manner defined by $\boldsymbol{vmax}$, which will also be used for the boundary condition. This initialization process can be written mathematically as

$$p_{ij}(k=0) = pmax_{s,j} + \text{rand}(0,1)\left(pmax_{e,j} - pmax_{s,j}\right) \quad (5a)$$

$$v_{ij}(k=0) = -vmax_j + \text{rand}(0,2)vmax_j \quad (5b)$$

where $\boldsymbol{pmax}_j$ and $vmax_j$ are the individual components of $\boldsymbol{pmax}$ and $\boldsymbol{vmax}$ for each swarm component j, respectively, with the structure $\boldsymbol{pmax}_j = [pmax_{s,j}, pmax_{e,j}]$ defining the start and end of swarm position initialization range and $vmax_j$ defining the maximum allowed velocity of the swarm. Note that $\boldsymbol{pmax}$ can define the swarm positions within any range in the search space, while $\boldsymbol{vmax}$ can only define the swarm velocities centered at zero.

Algorithm 1: Procedure for Limiting Swarm Velocities.

```
procedure LimVel ($\boldsymbol{v}$, $\boldsymbol{vmax}$)
  for $i = 1, \ldots, n_p$ do ▷ each particle in the swarm
    for $j = 1, \ldots, n_c$ do ▷ each particle component
      if $|v_{ij}| > vmax_j$ then ▷ over speed limit
        $v_{ij} = \text{sign}(v_{ij})vmax_j$
      else ▷ inside speed limit, no change
      end if
    end for
  end for
  return $\boldsymbol{v}$
end procedure
```

5) *Boundary Condition:* The basic boundary condition which is implemented in this paper is the swarm velocity limit represented by $\boldsymbol{vmax}$, whose implementation is described in Algorithm 1. A good starting point for defining the velocity limit is by setting each component of $\boldsymbol{vmax}$, $vmax_j$, to a value equal with the range of its respective $\boldsymbol{pmax}_j$ (i. e., $vmax_j = \left(pmax_{e,j} - pmax_{s,j}\right)/2$), thus enabling the swarm to explore the region outside its initialization range to a certain extent. Then, the algorithm is evaluated further and checked for consistency using a range of $\boldsymbol{vmax}$, which is

$$vmax_j = \frac{pmax_{e,j} - pmax_{s,j}}{v_{\text{lim}}}, \qquad v_{\text{lim}} = 2, \ldots, 30. \quad (6)$$

Note that v_{lim} is inversely proportional with the velocity limit imposed on the swarm. This is useful if the actual optimal solution happens to lie outside the initialization range, while limiting the possibilities of swarm explosion [21], [27].

6) *Refinement Procedure:* Refinement procedure was first proposed in [22] for assisting the swarm in escaping suboptimal solutions in inverse multiple magnetic dipole modeling problem by re-running the algorithm with the initialization region $\boldsymbol{pmax}$ set to an *a priori* knowledge from the previous PSO run solution $\boldsymbol{gbest}(k)$. An overview of this refinement procedure process, adapted for the magnetometer calibration parameters estimation problem, is described in Algorithm 2. Note that a new

Algorithm 2: Refinement Procedure Adapted From [22].

```
▷ The refinement procedure requires knowledge of the previous PSO run solution ($\boldsymbol{gbest}$) for an a priori values of its initialization conditions.
procedure Refine ($\boldsymbol{gbest}$, $\boldsymbol{p}_0$)
    for $j = 1, \ldots, n_c$ do                        ▷ for every swarm component $j$
        $\boldsymbol{pmax}_j = [gbest_j - p_{0,j}\ ,\ gbest_j + p_{0,j}]$
            ▷ define the initial position range
        $vmax_j = (pmax_{e,j} - pmax_{s,j})\,/v_{\text{lim}}$
            ▷ define the velocity limit
        for $i = 1, \ldots, n_p$ do                    ▷ for every swarm particle $i$
            initialize $p_{ij}$ and $v_{ij}$ using (5)
        end for
    end for
    $\boldsymbol{p}_i\big|_{i=\text{randi}(1,n_p)} = \boldsymbol{gbest}$
        ▷ Set one random particle in the swarm equal to previous global best (optional to ensure at least the same fitness value with the previous iteration)
    ...
        ▷ Continue with normal PSO iterations
end procedure
```

constant $p_{0,j}$, whose value is determined manually, is defined as the approximated search space around the *a priori* solution $gbest_j$ for each component j.

In principle, the refinement procedure serves to “shake” the swarm from the previously reached solution by reinitializing the swarm in a smaller initialization region. Technically, this is done by introducing a small value for $\boldsymbol{p}_0$ to generate a swarm around the *a priori* solution from previous PSO run with even smaller $\boldsymbol{vmax}$ from the previous run, since $\boldsymbol{vmax}$ is defined by the total range of $\boldsymbol{pmax}$ itself, as in (6). This is, technically, a redundant procedure along with the dynamic parameters, although the dynamic parameters retain the momentum of the swarm from previous iteration, while the refinement procedure allows more flexibility in determining the initial dynamics of the swarm. In this paper, refinement procedure is useful for tradeoff with smaller number of iteration limit $k_{\max}$ to evaluate the optimal settings for the algorithm.

For overview, the common values of key PSO parameters applied in this paper are summarized in Table I (some values are not fixed, so it can be evaluated separately in order to check the algorithm for consistency), while a flowchart describing the overall PSO algorithm is given in Fig. 3.

TABLE I
Specific PSO Parameters Applied in This Paper for Simulation and Experimental Tests

PARAMETER	VALUE	DESCRIPTION
$k_{\max}$	$200, \ldots, 1000$	Maximum number of iterations.
n_p	35	Number of swarm particles.
w_s	5	Starting value for w.
w_f	0.5	Final value for w.
w_e	0.382	End value for w.
c_{1s}	4	Starting value for c_1.
c_{2s}	0	Starting value for c_2.
$c_{1f},\ c_{2f}$	2	Final value for c_1 and c_2.
$c_{1e},\ c_{2e}$	2.5	End value for c_1 and c_2.
w_v	$(0.05, \ldots, 0.4)k_{\max}$	Iteration length for one cycle of varying w.
c_v	$(0.05, \ldots, 0.4)k_{\max}$	Iteration length for one cycle of varying c_1 and c_2.
$pmax_{j=1,\ldots,9}$	1	Initial boundary condition.
$pmax_{j=10,\ldots,12}$	5×10^3	Initial boundary condition.
v_{lim}	50	Speed limit modifier from (6).

C. Scalar Checking Objective

As explained in Section II, the calibration algorithm goal is to optimize the calibration parameters $\boldsymbol{K}_m$ and $\boldsymbol{k}_m$ from (2b). The estimated calibration parameters themselves are contained in the PSO solution (for each particle) with the mathematical structure

$$\boldsymbol{p}_i = [K_{m11,i}, K_{m12,i}, K_{m13,i}, K_{m21,i}, K_{m22,i}, K_{m23,i}, K_{m31,i}, K_{m32,i}, K_{m33,i}, k_{m1,i}, k_{m2,i}, k_{m3,i}]^\top \tag{7}$$

where

$$\boldsymbol{K}_{m,i} \equiv \begin{bmatrix} K_{m11,i} & K_{m12,i} & K_{m13,i} \\ K_{m21,i} & K_{m22,i} & K_{m23,i} \\ K_{m31,i} & K_{m32,i} & K_{m33,i} \end{bmatrix}, \qquad \boldsymbol{k}_{m,i} \equiv \begin{bmatrix} k_{m1,i} \\ k_{m2,i} \\ k_{m3,i} \end{bmatrix}.$$

In scalar checking objective, the algorithm has to ensure that all the calibrated magnetic field vector magnitude $\left|\check{\boldsymbol{b}}\right|$ is equal to a known reference magnetic field magnitude $|\boldsymbol{b}|$, independent of the magnetometer orientation itself. Thus, the fitness function for magnetometer calibration can be defined as the difference between $|\boldsymbol{b}|$ and $\left|\hat{\boldsymbol{b}}\right|$, or written mathematically as

$$F_{1,i} = \sum_{s=1}^{n_s} \left[|\boldsymbol{b}_s| - |\hat{\boldsymbol{b}}_{is}| \right]^2 \tag{8a}$$

or, by substituting (2b) into (8a)

$$F_{1,i} = \sum_{s=1}^{n_s} \left[|\boldsymbol{b}_s| - |\boldsymbol{K}_{m,i}\check{\boldsymbol{b}}_{is} - \boldsymbol{k}_{m,i}| \right]^2 \tag{8b}$$

where $i = [1, n_p] \in \mathbb{Z}$ and $s = [1, n_s] \in \mathbb{Z}$ are the index for the particle in the swarm and the index for measurement data, respectively.

D. Novel Rotation Axis Fitting Objective

The rotation axis fitting objective can be introduced to the algorithm as follows. Consider a set of magnetic field measurements from the magnetometer as the spacecraft tumbles as such that the measurement locus can be separated into at least two loci l, where each locus forms a circle (not necessarily a complete circle) on a plane with

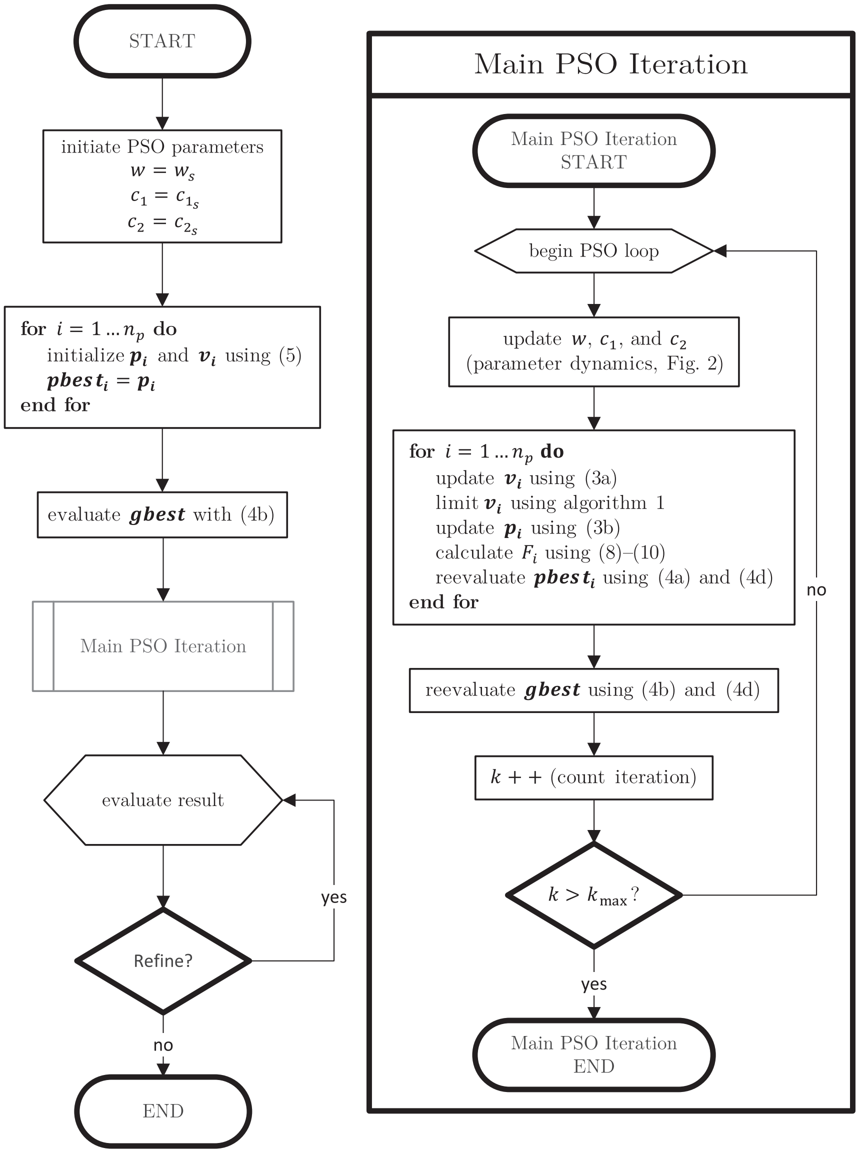

START
initiate PSO parameters
$w = w_s$
$c_1 = c_{1_s}$
$c_2 = c_{2_s}$
for $i = 1 \dots n_p$ do
initialize $\boldsymbol{p_i}$ and $\boldsymbol{v_i}$ using (5)
$\boldsymbol{pbest_i} = \boldsymbol{p_i}$
end for
evaluate $\boldsymbol{gbest}$ with (4b)
Main PSO Iteration
evaluate result
yes
Refine?
no
END
Main PSO Iteration
Main PSO Iteration START
begin PSO loop
update w, c_1, and c_2
(parameter dynamics, Fig. 2)
for $i = 1 \dots n_p$ do
update $\boldsymbol{v_i}$ using (3a)
limit $\boldsymbol{v_i}$ using algorithm 1
update $\boldsymbol{p_i}$ using (3b)
calculate F_i using (8)–(10)
reevaluate $\boldsymbol{pbest_i}$ using (4a) and (4d)
end for
no
reevaluate $\boldsymbol{gbest}$ using (4b) and (4d)
$k++$ (count iteration)
$k > k_{\max}$?
yes
Main PSO Iteration END

Fig. 3. Flowchart for the PSO algorithm applied in this paper. Note that the termination condition of the main PSO loop is simply by counting iteration until it reaches maximum, and refinement procedure is decided manually by inspecting the results.

different normal directions. This means that for each locus, the spacecraft ideally needs to rotate on a single axis. If the vectors representing each rotation axis of the spacecraft that forms the circle locus are known, then the objective of the PSO is to select the proper calibration parameters that will rotate these circular loci as such that the plane normal $\mathring{\boldsymbol{n}}_{il}$ (the plane containing a circle of measurement locus l after calibrated by particle i) is aligned with its respective known rotation axis $\mathring{\boldsymbol{\omega}}_l$. Thus, a second fitness function can be defined from the new objective:

$$F_{2,i} = \sum_{l=1}^{n_l} \left[1 - \left(\mathring{\boldsymbol{n}}_{il} \cdot (-\mathring{\boldsymbol{\omega}}_l) \right) \right]^2 \tag{9}$$

where the subscripts i and l are the index for the particle in the swarm and the index for individual measurement locus that lies on a single plane, respectively, the unit vector $\mathring{\boldsymbol{n}}_{il}$ is the normal direction of the plane which contains the calibrated measurement locus l as estimated by particle i, where its positive direction follows the right-hand rule of the data sequence in the calibrated measurement locus, and the unit vector $\mathring{\boldsymbol{\omega}}_l$ is the rotation axis that corresponds to the measurement locus l, which can be obtained from gyroscope data, direct observation (e. g., in preflight test), or estimated from vector sensors such as star trackers [28], [29]. Note that the direction of $\mathring{\boldsymbol{\omega}}_l$ is inverted with a minus sign because the direction of magnetometer rotation is the opposite of the measurement locus sequence. In summary, the second fitness function $F_{2,i}$ represents the sum of angular difference between $\mathring{\boldsymbol{n}}_{il}$ and $\mathring{\boldsymbol{\omega}}_l$ for $i = 1, \dots, n_p$ and $l = 1, \dots, n_l$, which will be minimized by the PSO.

Also note that the plane normal direction and the rotation axis are both represented by unit vectors $\mathring{\boldsymbol{a}} \equiv \boldsymbol{a}/|\boldsymbol{a}|$, since only the direction information is of interest for this

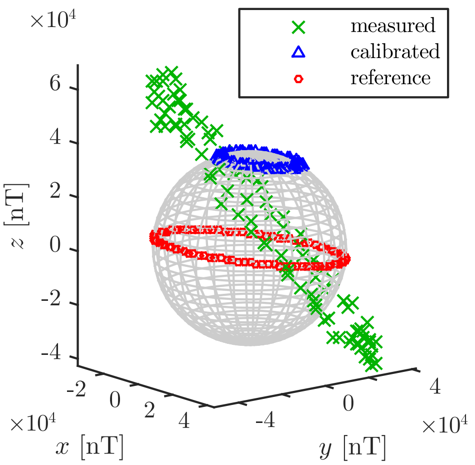


Fig. 4. 3-D plot of magnetic field vectors (the axes represent the magnetometer frame in nT), depicting a case with scale and offset ambiguity caused by insufficient information from one locus. The calibrated data fits into a plane with correct normal direction and uniform distance from the center (the locus fits on a sphere surface), although in a wrong offset and scale compared to the reference, which is the simulated error-free data. The data shown contains 20% random noise relative to the reference.

purpose. This is also the reason for incomplete circle loci to be sufficient: Since only the plane normal direction information is required for fulfilling this objective, then an incomplete circle, as long as it fits into a plane, could provide enough information on the plane direction. However, it should be kept in mind that although this is the case for the rotation axis fitting objective, such limited information in the data (a minimum of two small curves) might be insufficient for the scalar checking objective, which requires some coverage of the measurement locus in all directions. On the other hand, the reason for the requirement of at least two circle loci is because one circle locus only provides a single constraint for the data plane in two axes, leaving scale and offset ambiguity in one axis that is aligned with the plane normal as well as a rotational ambiguity along that axis. An example of this loss of scale and offset information is depicted in Fig. 4, where the data is erroneously scaled and offset as such that the calibrated data still fulfill its ill-constrained objectives.

As previously mentioned, the value of $\mathring{\boldsymbol{\omega}}_l$ can be determined directly from direct observation during the calibration test, where the spacecraft is rotated manually along its spacecraft body axes (or from other sensors, e. g., gyroscopes or star trackers for on-orbit applications). On the other hand, the value of $\mathring{\boldsymbol{n}}_{il}$ is calculated using orthogonal distance regression plane fitting, where the sum of squared orthogonal distances from the points on the calibrated measurement locus to the plane is minimized. Mathematical tools to solve this least squares problem are well documented, and this paper implements a singular value decomposition (SVD) method [30], [31]. This method is summarized in Algorithm 3.

First, a general estimate of the rotation axis from the calibrated data is calculated from the cross product between the first measurement in the locus $\hat{\boldsymbol{b}}_{i1,l}$ and the next measurement $\hat{\boldsymbol{b}}_{i(2,\ldots,n_{sl}),l}$ after the magnetometer frame is rotated with arbitrary angular difference (between 0° and 180°). Then, the sign of the dot product with the previously estimated $\mathring{\boldsymbol{n}}_{il}$ will show whether the result follows the right-hand rule. In general, the second evaluation point $\hat{\boldsymbol{b}}_{i(2,\ldots,n_{sl}),l}$ need to be evaluated manually as the total measurement locus is segmented into several loci. If

$$\left(\mathring{\boldsymbol{n}}_{il} \cdot \left(\hat{\boldsymbol{b}}_{i1,l} \times \hat{\boldsymbol{b}}_{i(2,\ldots,n_{sl}),l}\right)\right) \approx 0$$

then the first and/or second evaluation point has to be changed with another point since it indicates either that those two points have angular displacement of $\approx 0^\circ$ or $\approx 180^\circ$ relative to each other or that the data at those points is affected by very large noise that it alters the estimated rotation axis vector. It is also important to note that this implementation of rotation axis fitting method cannot differentiate between the measurement locus caused by the actual rotation of the spacecraft in inertial space and the one caused by the varying magnetic field direction, where the latter case does not fit into (9) and might appear in the data from a long flight period in polar orbit.

Algorithm 3: Algorithm for orthogonal regression plane fitting using SVD [30].

```
1: calculate $\bar{\hat{\boldsymbol{b}}}_{i,l} = \sum_{s_l=1}^{n_{sl}} \hat{\boldsymbol{b}}_{is_l,l}/n_{sl}$
2: construct $\boldsymbol{A} = [\hat{\boldsymbol{b}}_{i1,l} - \bar{\hat{\boldsymbol{b}}}_{i,l}, \ldots, \hat{\boldsymbol{b}}_{in_{sl},l} - \bar{\hat{\boldsymbol{b}}}_{i,l}]$
3: decompose $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$
                        ▷ singular value decomposition
4: find $\mathrm{col}_{\min} = \arg\min_x \left\{\mathrm{diag}_x\left(\boldsymbol{S}\right)\right\}$
                  ▷ column with minimum singular value
5: $\mathring{\boldsymbol{n}}_{il} = \boldsymbol{U}\{:, \mathrm{col}_{\min}\}$
     ▷ plane normal vector corresponds to the column
       of minimum singular value
6: $\mathring{\boldsymbol{n}}_{il} = \mathrm{sign}\left(\mathring{\boldsymbol{n}}_{il} \cdot \left(\hat{\boldsymbol{b}}_{i1,l} \times \hat{\boldsymbol{b}}_{i(2,\ldots,n_{sl}),l}\right)\right)\mathring{\boldsymbol{n}}_{il}$
     ▷ correction of the plane normal positive direction
       by comparing it with the measurement sequence
```

Finally, the fitness values from the two objectives have to be combined. This paper applies the fixed weight aggregation technique to combine the fitness values for magnetometer calibration parameters estimation problem because of its simplicity [24]. The total fitness function combines $F_{1,i}$ from (8) and $F_{2,i}$ from (9) into

$$F_i = c_{f_1}\frac{\sqrt{F_{1,i}}}{n_s} + c_{f_2}\frac{\sqrt{F_{2,i}}}{n_l} \tag{10}$$

where c_{f_1} and c_{f_2} are the fixed weight for each objective. Note that both fitness values are normalized with respect to their own squared sum index elements. Since this method is nonanalytical, the initial values for the weight is approximated from the ratio between the fitness values of the two functions under several simulation model. The weights are then adjusted to achieve global optimum. A balanced solution can be achieved with $c_{f_1} = 1$ and $c_{f_2} = 4 \times 10^4$. Obviously, the important part for achieving a nondominated solution is the ratio between the two weights, although the

TABLE II
Comparison of Estimated Model Parameters From different Simulation Models in Relation to the True Reference Value

PARAMETER	REFERENCE	MODEL 1, EKF: $n_l = 3$, $\lvert\boldsymbol{b}\rvert = 35\,000$ [nT], $\delta = 2000$ [nT]	MODEL 1, PSO: $n_l = 3$, $\lvert\boldsymbol{b}\rvert = 35\,000$ [nT], $\delta = 2000$ [nT]	MODEL 2, PSO: $n_l = 3$, $\lvert\boldsymbol{b}\rvert = 35\,000$ [nT], $\delta = 5500$ [nT]	MODEL 3, PSO: $n_l = 2$, $\lvert\boldsymbol{b}\rvert = 35\,000$ [nT], $\delta = 2000$ [nT]	MODEL 4, PSO: $n_l = 2$, $\lvert\boldsymbol{b}\rvert = 17\,000$–$48\,000$ [nT], $\delta = 2000$ [nT]	MODEL 5, PSO: $n_l = 3$ (incomplete circle), $\lvert\boldsymbol{b}\rvert = 35\,000$ [nT], $\delta = 2000$ [nT]
off_{m_c1} $[nT]$	12 600	12 886	12 586	12 053	12 781	11 590	11 565
off_{m_c2} $[nT]$	−200	−5930	−811	−1236	−182	−772	−263
off_{m_c3} $[nT]$	1800	−3167	1420	1577	2309	1886	157
S_{m_c11}	0.749	1.021	0.758	0.784	0.725	0.774	0.82
S_{m_c22}	1.122	0.998	1.129	1.205	1.141	1.098	1.114
S_{m_c33}	2.433	2.486	2.424	2.583	2.49	2.388	2.578
S_{m_c12}	0.354	0.124	0.378	0.392	0.365	0.324	0.315
S_{m_c13}	0.677	0.242	0.683	0.719	0.665	0.767	0.608
S_{m_c23}	–0.209	–0.369	–0.18	–0.277	–0.265	–0.112	–0.229
S_{m_c21}	–0.242	0.124	–0.23	–0.204	–0.239	–0.255	–0.212
S_{m_c31}	–0.395	0.242	–0.403	–0.442	–0.458	–0.428	–0.137
S_{m_c32}	–1.043	–0.369	–0.984	–0.993	–1.03	–1.117	–0.999
F_g		16 511	199.05	470.09	185	179.39	200.36

absolute values themselves are important in the context of fitness value scaling.

It is also important to note that, with the defined objectives, the PSO solution space still contains two “global” minima. One minimum point is the reflection of the other minimum, where the scalar checking objective from (8) is fulfilled, but the measurement locus plane fitting objective from (9) is fulfilled under a reflected frame definition with a left-handed triad. Correcting the estimated calibration parameter $\boldsymbol{K}_m$, which is a three-dimensional (3-D) transformation matrix, can be performed by reflecting it back if it contains a reflection transformation using

$$\boldsymbol{K}_m = \text{sign}\left(\lvert\boldsymbol{K}_m\rvert\right)\boldsymbol{K}_m \tag{11}$$

since a transformation matrix that contains a reflection transformation will have a negative determinant.

IV. TEST RESULTS

A. Simulation Test

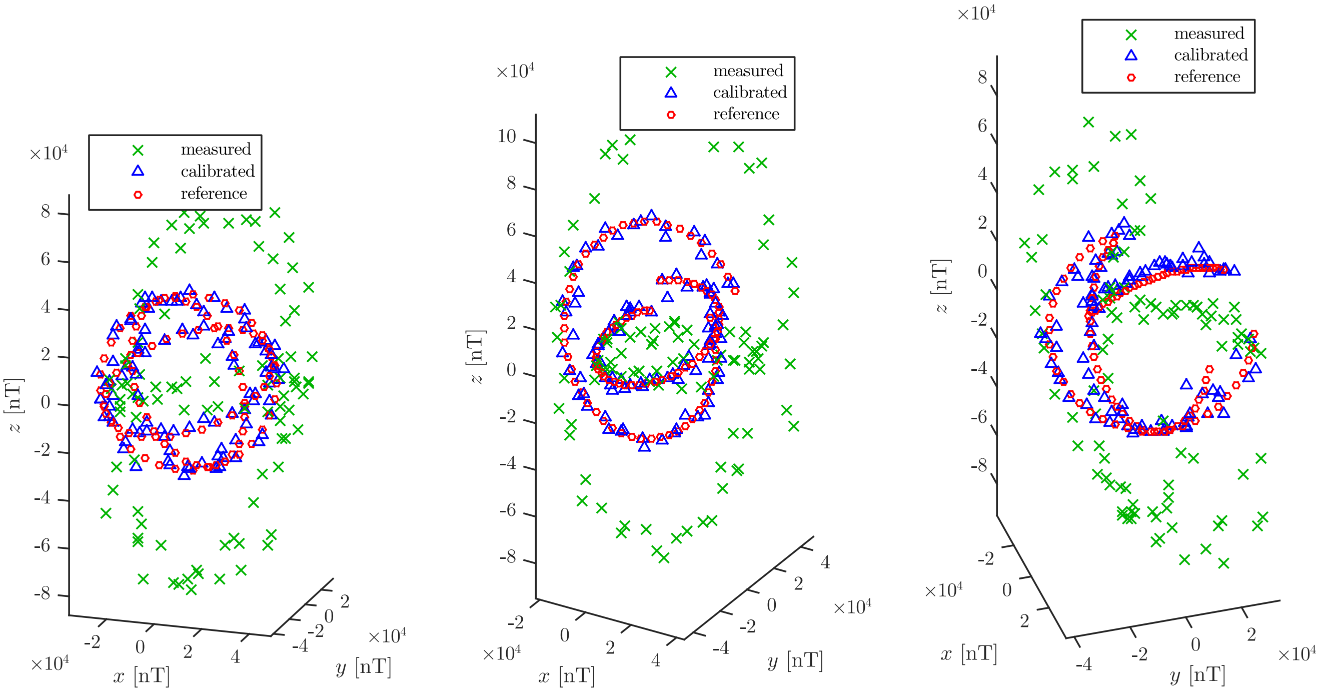
The PSO algorithm is evaluated using the magnetometer model from (2a) under a simulated environmental condition that will be experienced by Aalto-1 satellite [32]. Aalto-1 planned altitude is around 500–900 km, and IGRF12 model [33] shows a geomagnetic field variation of approximately 17 000–48 000 nT on the average altitude of 700 km. Thus, the algorithm is tested with the following model variations:

1) three full circle measurement loci with constant $\lvert\boldsymbol{b}\rvert = 35\,000$ nT and standard deviation $\delta = 2000$ nT in the magnetometer measurements. This is the nominal magnetometer noise level emulating Aalto-1 magnetometer;
2) same as above except with higher noise $\delta = 5500$ nT. This model is used for verifying the algorithm performance under very high noise;
3) same as the first model except the data is contained only in two full circle measurement loci;
4) two full circle measurement loci with varying $\lvert\boldsymbol{b}\rvert$ in the range of 17 000–48 000 nT (linear increase from the first measurement until the last one) and nominal noise level; and
5) three incomplete circle measurement loci with constant $\lvert\boldsymbol{b}\rvert = 35\,000$ nT and nominal noise level. All models simulate 100 magnetometer measurements and the rotation axis information contains no error (perfect gyroscope measurement).

The simulation results (and the model reference for comparison) are shown in Table II, while the graphical representations of the measured, calibrated, and true magnetic field vector locus for some model variations (selected for brevity) are given in Fig. 5. Table II provides numerical data of the estimated calibration parameters that can be compared with the included reference parameters. Additionally, estimated parameters using extended Kalman filter (EKF) described in [1] is included as a performance comparison under model 1. Since the calibration parameters are a combination of scaling matrix and offset parameters, these numbers can be translated into geometrical representation depicted in Fig. 5, where the calibrated data should be positioned as close as possible to the reference data.

Table II shows that the new algorithm successfully estimated the calibration parameters for different simulation models, although the performance varies with different modeled scenarios—Fig. 5 also shows that the measurement points are successfully calibrated close to the reference points. In more detailed analysis, the complete data also shows that the error between the corrected and modeled measurement vectors still falls within the noise distribution level.

Under the simulated magnetometer, which takes into account misalignment and cross-axis soft iron errors, the new algorithm accuracy definitely improved from EKF, whose estimated calibration matrix is limited to symmetrical matrix. This can be seen from the difference in fitness values: the major contributor is the large value of F_2 in the EKF estimates, which is 0.036 compared to the PSO’s 0.00003, while for F_1 they are quite similar. This is because F_2, as



(a)

(b)

(c)

Fig. 5. 3-D plot of magnetic field vectors from simulation comparing the measured, calibrated, and true magnetic field vector data points. All modeled with magnetometer noise standard deviation $\delta = 2000\,\text{nT}$. (a) Model 1: Three circle loci, constant ambient magnetic field magnitude. (b) Model 4: Two circle loci, gradually increasing ambient magnetic field magnitude. (c) Model 5: Three incomplete circle loci, constant ambient magnetic field.

described in (9), measures the error from rotational ambiguity based on the known rotation axis, while F_1 measures the fitness using scalar checking—The large difference in the final fitness value is simply caused by the weighting constants from (10).

On the other hand, under an ideal model (symmetrical, triangular, or diagonal calibration matrix), a perfectly adjusted EKF (good initial estimate, perfectly known measurement noise covariance) can perform accurately with much less computational cost, and its recursive algorithm can provide real-time estimation result from the first measurement point. However, it requires a significantly larger number of measurement points for the solutions to converge: For a simulated symmetrical calibration matrix under the same condition with model 1, the EKF typically stabilize at the 1500th measurement point, while the PSO in this paper is typically run for 500 iterations in all simulation cases with 100 measurement points. Similar performance can be achieved using 100–200 iterations, assisted with one or two refinement procedures. PSO performance does improve with increasing data points, but the tradeoff in computational costs is heavier than faster algorithm such as EKF because PSO is a batch algorithm.

Extensive testing on the PSO algorithm also shows that the models with varying ambient magnetic field magnitude and incomplete circle loci show less accurate results in some cases, especially when the data contains large noise level and less data points. An example of this can be observed in Fig. 5(c) (a case of incomplete circle loci), where the loci curves are shorter and cover less area of the sphere surface, and the solution falls into a local minimum (poor estimate of z-axis offset values in Table II). This is expected since all of these conditions relate to low data quality and information availability, which are insufficient for the scalar checking objective. On the other hand, the model with $\delta = 5500\,\text{nT}$ noise level (the highest noise level in Table II), yields the worst fitness value. However, the F_g values themselves are not comparable between different models, as indicated by the incomplete circle loci model that yields better fitness value but produce less accurate parameters estimate.

The discrepancy between fitness values and actual estimation quality shows that fitness values alone are unreliable especially in real experiment data where the error model of the magnetometer and its environment is not completely known. However, fitness values can be useful for determining the relative accuracy between different PSO runs where the data still contain identical error model. Moreover, detecting large error in the calibration parameters estimate is possible, even for experimental data where the true model is not known, by inspecting the graphical representation of the calibrated measurement locus (e. g., in cases similar with Fig. 4, where the calibrated measurement locus is extremely scaled and displaced).

B. Experimental Test

The experimental test was performed for the engineering model (EM) of Aalto-1 nanosatellite at a magnetic test facility operated by Finnish Meteorological Institute

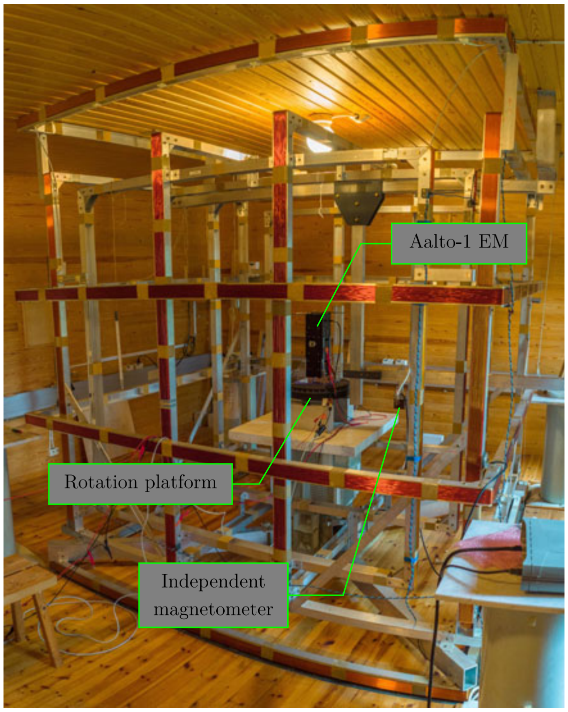


Fig. 6. Magnetic test facilities with Helmholtz cage setup in Nurmijärvi Geophysical Observatory, operated by Finnish Meteorological Institute. The specifications of the test setup are available in [34].

located in Nurmijärvi, Finland [34], as depicted in Fig. 6. The test setup includes

1) Aalto-1 EM;
2) a three-axis Helmholtz cage setup;
3) a one-axis manual rotation platform; and
4) a custom LEMI-CLE fluxgate magnetometer (part number N9512)—referred to as *independent magnetometer* from this point. This independent magnetometer is used for calibrating the Helmholtz coil output.

Using the known angular reference from the initial orientation data (note that this is unfiltered data), it can be calculated that the heading error of the magnetometer improved from 5.24°to 13.24° before calibration to 1.9°–7.3° after calibration.

Another aspect to analyze in this preflight test is whether any inconsistency in the calibration parameters appears when the magnetometer is calibrated in different ambient magnetic field magnitude. This is due to the fact that, as given in (1), the proposed magnetometer model parameters is not dependent on the ambient magnetic field magnitude itself. This means that any inconsistency in the calibration parameters when the spacecraft is subjected to different magnitude of ambient magnetic field might indicate some problems, e. g., the presence of noise in the data affecting the accuracy of the calibration algorithm solution and/or the presence of hysteresis effect from ferromagnetic materials, which is not included in the magnetometer model in (1). Thus, the measurements taken from the magnetometer calibration test consist of four sets of data from the combination of two ambient magnetic field magnitudes $|\boldsymbol{b}|$ (20000 and 50000 nT) set by the Helmholtz cage and two operation modes of the spacecraft (*nominal operating mode* and *full-power operating mode*).

TABLE III
Estimated Aalto-1 EM Magnetometer Model Parameters Under Different Ambient Magnetic Field $|\boldsymbol{b}|$ for Nominal Operating Mode

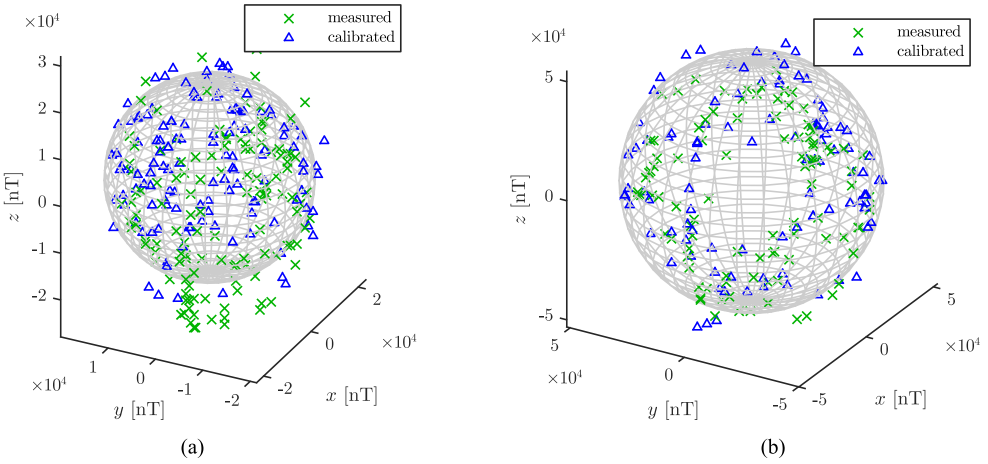
PARAMETER	20 000 [nT], $n_l = 3, n_s = 130$	50 000 [nT], $n_l = 3, n_s = 102$	50 000 [nT], $n_l = 3$, $n_s = 102$ using EKF
off_{m_c1} [nT]	–5145.6	–5226.9	–5961.3
off_{m_c2} [nT]	3626.5	3350	2200.2
off_{m_c3} [nT]	1803.5	2395.6	3601.3
S_{m_c11}	0.8748	0.8617	0.8426
S_{m_c22}	–0.8929	–0.9146	–0.9072
S_{m_c33}	–1.1777	–0.9329	–0.9035
S_{m_c12}	–0.0369	–0.0367	–0.0192
S_{m_c13}	–0.0170	0.0036	0.0351
S_{m_c23}	–0.0027	0.0058	–0.0266
S_{m_c21}	–0.0258	–0.0306	–0.0192
S_{m_c31}	0.0154	–0.0068	0.0351
S_{m_c32}	0.1032	–0.0355	–0.0266
F_g	349.92	375.52	430.42

The estimated calibration parameters, including results using EKF for comparison, are given in Table III. It shows that the parameters converge to similar values, well within the variation demonstrated in the simulation. This is also apparent in the fitness values $F_g \approx 350$ for all setups, which are in the range of simulated data accuracy under similar number of measurement data (simulation uses 100 measurements, while the experiment uses 100–130 measurements). The variation in the values is also contributed by the noise level of Aalto-1 magnetometer, which becomes relatively high compared to the signal, especially for the 20 000 nT ambient magnetic field case. This can be seen from the measurement locus compared with the 50 000 nT case, both presented in Fig. 7. For brevity, the data in Fig. 7 and Table III shows the experiment results under the spacecraft nominal operating mode only. Results for full-powered mode of the spacecraft behaved similarly under different ambient magnetic field, although a difference in the offset vector was observed, caused by the change in the electrical loads surrounding the magnetometer.

Comparing the results from EKF, the difference in the results is contributed from the fact that the EKF can only estimate symmetrical calibration matrix using scalar checking, which also appeared in the simulation tests. However, unlike the simulated model, the fitness value (F_2 in particular) for EKF is not too different from the PSO estimates, since the nondiagonal elements of the magnetometer parameters are much closer to zero.

C. Discussion

Using simulated data (presented in Table II), the improved PSO is proven to be capable of estimating calibration parameters under varying ambient magnetic field magnitude as well as incomplete circle measurement loci. The improvement of the new algorithm over the standard scalar checking method that estimates triangular or diagonal scaling matrix is the ability to estimate misalignment



(a)

(b)

Fig. 7. 3-D plot of magnetic field vectors from experiment showing calibrated and measured magnetic field vector loci under different ambient magnetic field $|\boldsymbol{b}|$ with the spacecraft in nominal operating mode. (a) $|\boldsymbol{b}| = 20\,000$ [nT], $n_l = 3$, $n_s = 130$. (b) $|\boldsymbol{b}| = 50\,000$ [nT], $n_l = 3$, $n_s = 102$.

and cross-axis errors of the magnetometer using the knowledge of the magnetometer rotation axis. As such, the new algorithm can be used to accurately calibrate magnetometers without strict alignment procedure and full knowledge of the system's attitude. This advantage might be beneficial when the spacecraft is already on orbit and the calibration parameters need to be updated as the sensor characteristic (or magnetic characteristic of the spacecraft) changes, although more testing is required to conclude the algorithm performance under real flight data.

The estimation accuracy itself is affected by data noise and incomplete information (e. g., small number of data points, unbalanced loci in one side of the sphere). This inaccuracy showed up as inconsistency of the estimated parameters from the experimental test (especially the bias in z-axis; see Table III), where the measurement locus, shown in Fig. 7(a), around the negative z-axis is particularly noisy. One approach to minimize this is by using multisampled measurements, where each measured magnetic field vector is averaged from multiple, high-frequency reading of the magnetometer to eliminate most of the noise. This is an approach used in another experimental test using different magnetometer setup in [23], where a magnetometer was calibrated for validating magnetorquer output.

The current implementation of the new feature, however, still has some limitations:

1) individual measurement error factors (i. e., scale factor, nonorthogonality, misalignment, soft iron error, hard iron error, and magnetometer offset error) are still not uniquely solvable inside $\boldsymbol{K}_m$ and $\boldsymbol{k}_m$ (this is also true for other scalar checking algorithm);
2) the measurement data has to be dividable into at least two segments ($n_l \geq 2$) where the measurement locus of each segment l can be averaged into a single plane with unique normal direction $\mathring{\boldsymbol{n}}_l$ (implemented in this work with orthogonal distance regression plane fitting) and the average rotation axis of the magnetometer $\mathring{\boldsymbol{\omega}}_l$ when the measurement was taken for that segment is known;
3) the requirement for each measurement loci to be contained in one rotation plane means that ideally the spacecraft is equipped with active attitude control instruments such as reaction wheels or thrusters;
4) in this paper, the rotation axis information for the experiment was known from the test procedure, while for the simulation it is taken directly from the model. For calibration using flight data, it can be obtained from other sensors such as gyroscopes or star trackers. However, more analysis is required to investigate the effect of errors in the rotation axis information;
5) the algorithm cannot differentiate the measurement locus caused by the actual rotation of the spacecraft in inertial space and the one caused by the varying magnetic field direction, where the latter case will appear in a long flight period in polar orbit. Thus, for on-orbit application, one needs to make sure that no significant ambient magnetic field direction change occurs in individual locus when the data is segmented; and
6) the rotation axis fitting method also provide a significant change in determining the estimation accuracy from existing methods, since the calibration parameters will try to align the data with the reference axes of the sensor that provide the rotation axis information. This could become a disadvantage if the sensor itself is not properly aligned with the spacecraft body axis. However, since it is possible to derive rotation axis information from other vector measurement sensors [29], this approach opens up the possibilities to align the magnetometer directly with imaging payloads on board the spacecraft.

For practical use of on-orbit application, extensive test is needed for validating the algorithm performance from flight data, especially concerning factors that affect the validity of rotation axis information (e. g., measurement errors and the change in magnetic field direction especially near the geomagnetic poles). This will help in identifying key factors that influence the flight data quality such as the number of measurement points, minimum length of each locus curve, ideal spacecraft spin rate, and sensor sampling rate.

V. CONCLUSION

A novel approach was developed to improve the standard PSO implementation for magnetometer calibration parameters estimation problem in magnetic domain without the need of compromising the number of estimated parameters. The improved PSO is designed to estimate a full 3 × 3 calibration matrix $\boldsymbol{K}_m$ and 3 × 1 calibration vector $\boldsymbol{k}_m$ (a total of 12 parameters) by adding a new rotation axis fitting objective definition, which requires the knowledge of the magnetometer rotation axis.

Simulated data shows that the improved algorithm is capable of accurately estimating calibration parameters under varying ambient magnetic field magnitude as well as incomplete circle measurement loci, where its accuracy is affected by data quality, such as small number of data points, unbalanced loci in one side of the sphere, and noise level. Further experimental tests using flight data should be performed to investigate the algorithm accuracy in practical on-orbit application.

The algorithm can be improved further using a more general curve fitting algorithm to fit the measurement locus into the angular velocity information so that the requirements on the measurement locus can be relaxed. The reliability of the calibration parameters might also be improved by including time-varying parameters that corresponds to the electrical loads on the spacecraft subsystems, such as proposed in [5]. Additionally, the algorithm can be improved by implementing a more sophisticated fitness value weighting technique and termination condition to ensure *pareto optimized* result of the two objectives.

REFERENCES

[1] J. L. Crassidis, K.-L. Lai, and R. R. Harman
Real-time attitude independent three-axis magnetometer calibration
J. Guid., Control, Dyn., vol. 28, no. 1, pp. 115–120, 2005.

[2] C. C. Foster and G. H. Elkaim
Extension of a two-step calibration methodology to include nonorthogonal sensor axes
IEEE Trans. Aerosp. Electron. Syst., vol. 44, no. 3, pp. 1070–1078, Jul. 2008.

[3] V. Renaudin, M. H. Afzal, and G. Lachapelle
Complete triaxis magnetometer calibration in the magnetic domain
J. Sensors, vol. 2010, 2010, Art. no. 967245.

[4] L.-Y. Liu *et al.*
The magneto-resistive magnetometer of BCU on the Tatiana-2 satellite
Terrestrial, Atmos. Ocean. Sci., vol. 23, no. 3, pp. 317–326, 2012.

[5] J. C. Springmann and J. W. Cutler
Attitude-independent magnetometer calibration with time-varying bias
J. Guid., Control, Dyn., vol. 35, no. 4, pp. 1080–1088, 2012.

[6] J. R. Wertz
Spacecraft Attitude Determination and Control (Astrophysics and Space Science Library: A series of books on the recent developments of space science and of general geophysics and astrophysics). Berlin, Germany: Springer-Verlag, 1978.

[7] F. L. Markley and J. L. Crassidis
Fundamentals of Spacecraft Attitude Determination and Control (Space Technology Library). Berlin, Germany: Springer-Verlag, 2014.

[8] H. Lühr, F. Yin, and R. Bock
Magnetic properties of CHAMP and their effects on in-orbit calibration
J. Sensors Sensor Syst., vol. 2, no. 1, pp. 9–17, 2013.

[9] Z. Zeng, S. Zhang, Y. Xing, and X. Cao
Robust adaptive filter for small satellite attitude estimation based on magnetometer and gyro
Abstract Appl. Anal., vol. 2014, 2014, Art. no. 159149.

[10] A. Ali, S. Siddharth, Z. Syed, and N. El-Sheimy
Swarm optimization-based magnetometer calibration for personal handheld devices
Sensors, vol. 12, no. 9, pp. 12455–12472, 2012.

[11] D. Gebre-Egziabher, G. H. Elkaim, J. D. Powell, and B. W. Parkinson
Calibration of strapdown magnetometers in magnetic field domain
J. Aerosp. Eng., vol. 19, no. 2, pp. 87–102, 2006.

[12] Y. X. Liu, X. S. Li, X. J. Zhang, and Y. B. Feng
Novel calibration algorithm for a three-axis strapdown magnetometer
Sensors, vol. 14, no. 5, pp. 8485–8504, 2014.

[13] A. Wahdan, J. Georgy, W. F. Abdelfatah, and A. Noureldin
Magnetometer calibration for portable navigation devices in vehicles using a fast and autonomous technique
IEEE Trans. Intell. Transp. Syst., vol. 15, no. 5, pp. 2347–2352, Oct. 2014.

[14] K. Mohamadabadi
Anisotropic magnetoresistance magnetometer for inertial navigation systems Ph.D. dissertation, Ecole Polytechnique X, Palaiseau, France, 2013. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00946970

[15] R. Eberhart and J. Kennedy
A new optimizer using particle swarm theory
In *Proc. 6th Int. Symp. Micro Mach. Human Sci.*, 1995, pp. 39–43.

[16] J. Kennedy and R. Eberhart
Particle swarm optimization
In *Proc. 1995 IEEE Int. Con. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[17] R. Eberhart and Y. Shi
Particle swarm optimization: Developments, applications and resources
In *Proc. 2001 Congr. Evol. Comput.*, vol. 1, 2001, pp. 81–86.

[18] R. Poli
Analysis of the publications on the applications of particle swarm optimisation
J. Artif. Evol. Appl., vol. 2008, 2008, Art. no. 685175.

[19] Z. Wu, Y. Wu, X. Hu, and M. Wu
Calibration of three-axis strapdown magnetometers using particle swarm optimization algorithm
In *Proc. 2011 IEEE Int. Symp. Robot. Sensors Environ.*, 2011, pp. 160–165.

[20] Z. Wu, Y. Wu, X. Hu, and M. Wu
Calibration of three-axis magnetometer using stretching particle swarm optimization algorithm
IEEE Trans. Instrum. Meas., vol. 62, no. 2, pp. 281–292, Feb. 2013.

[21] D. Bratton and J. Kennedy
Defining a standard for particle swarm optimization
In *Proc. 2007 IEEE Swarm Intell. Symp.*, 2007, pp. 120–127.

[22] E. Carrubba, A. Junge, F. Marliani, and A. Monorchio
Particle swarm optimization for multiple dipole modeling of space equipment
IEEE Trans. Magn., vol. 50, no. 12, Dec. 2014, Art. no. 7028010.

[23] B. A. Riwanto
Cubesat attitude system calibration and testing
M.S. thesis, Aalto University, Espoo, Finland, 2015. [Online]. Available: http://urn.fi/URN:NBN:fi:aalto-201509184426

[24] M. Reyes-Sierra and C. A. C. Coello
Multi-objective particle swarm optimizers: A survey of the state-of-the-art
Int. J. Comput. Intell. Res., vol. 2, no. 3, pp. 287–308, 2006.

[25] M. Clerc and J. Kennedy
The particle swarm—explosion, stability, and convergence in a multidimensional complex space
IEEE Trans. Evol. Comput., vol. 6, no. 1, pp. 58–73, Feb. 2002.

[26] R. C. Eberhart and Y. Shi
Comparing inertia weights and constriction factors in particle swarm optimization
In *Proc. 2000 Congr. Evol. Comput.*, 2000, vol. 1, pp. 84–88.

[27] Y. Shi and R. C. Eberhart
Parameter selection in particle swarm optimization
In *Evolutionary Programming VII*, (Lecture Notes in Computer Science), V. Porto, N. Saravanan, D. Waagen, and A. Eiben Eds. Berlin, Germany: Springer-Verlag, 1998, vol. 1447, pp. 591–600.

[28] S. Jo, Y. Choi, and H. Bang
Optimal angular velocity estimation of spacecraft using only star tracker measurements
J. Guid., Control, Dyn., vol. 38, no. 2, pp. 342–346, 2015.

[29] I. Y. Bar-Itzhack
Classification of algorithms for angular velocity estimation
J. Guid., Control, Dyn., vol. 24, no. 2, pp. 214–218, 2001.

[30] I. Söderkvist
Using SVD for some fitting problems, 2009. [Online]. Available: http://www.ltu.se/cms_fs/1.51590!/svd-fitting.pdf

[31] H. Späth
Orthogonal least squares fitting with linear manifolds
Numerische Mathematik, vol. 48, no. 4, pp. 441–445, 1986.

[32] A. Kestilä *et al.*
Aalto-1 nanosatellite—technical description and mission objectives
Geosci. Instrum., Methods Data Syst., vol. 2, no. 1, pp. 121–130, 2013.

[33] C. C. Finlay *et al.*
International geomagnetic reference field: The eleventh generation
Geophys. J. Int., vol. 183, no. 3, pp. 1216–1230, 2010.

[34] F. M. Institute, Nurmijärvi magnetic calibration and test laboratory, 2015. [Online]. Available: http://en.ilmatieteenlaitos.fi/c/document_library/get_file?uuid=6b00a65f-31d5-4fe4-a8da-410e1059-a65c&groupId=30106

Bagus Adiwiluhung Riwanto received the B.Sc. degree in electrical engineering from the Institut Teknologi Bandung, Bandung, Indonesia, in 2011, where he was involved in a nanosatellite design project in designing magnetorquer for attitude control, and the M.Sc. degree in space science and technology from Aalto University, Espoo, Finland, and Luleå University of Technology, Luleå, Sweden, in 2015, under the Erasmus Mundus' Space Master Programme. During the master's study, he worked on Aalto-1 and Aalto-2 nanosatellite Projects, specifically in the integration and testing of the attitude determination and control system. He is currently working toward the Doctorate degree at Aalto University. His doctoral research includes the development of remote sensing nanosatellites.

Tuomas Tikka received the M.Sc.(tech.) degree from Aalto University in 2012. He is currently working toward the Doctorate degree at Aalto University.

He is currently a Small Satellite Technology Professional. Also he is currently a CEO with five year experience from Aalto University's CubeSat projects. He has been working with project management, quality management, and system engineering tasks in Aalto-1, Aalto-2, Hello World, and ASPECT CubeSat missions. Recently in 2016, he founded a new company Reaktor Space Lab with key personnel from Aalto-2 project, offering space-based services and turnkey small satellite solutions utilizing cost-effective CubeSat platforms. He has also been active in small satellite testing standardization in ISO, and research of development and testing practices for NewSpace Projects in his doctoral thesis.

Antti Kestilä received the Graduate degree from Delft University, The Netherlands and the M.Sc. degree. He is currently working toward the Ph.D. degree in the Department of Radio Science and Engineering, Aalto University School of Science and Technology, Espoo, Finland.

He is currently a Space System Engineer. Also he is currently the co-founder, System Engineer, and Mission Designer of Aalto-1 small satellite, the first Finnish satellite to fly in space, and is involved in several other small satellite projects. His research interests include pulsar star navigation, in-space propulsion, remote sensing technologies, and additive manufacturing in space. He has also worked on aerospace human–machine interfacing at Osaka University.

Jaan Praks (S'98–M'02) was born in Estonia, on November 16, 1971. He received the M.Sc. degree in physics from the University of Tartu, Tartu, Estonia, in 1996 and the D.Sc.(tech.) degree from Aalto University, Espoo, Finland, in 2012.

He is currently an Assistant Professor with Aalto University, where he is currently working on remote sensing instrumentation and nanosatellites with the Microwave Remote Sensing and Space Technology Group. His research interests include SAR polarimetry, SAR interferometry, SAR polarimetric interferometry, space technology, and nanosatellites.

Prof. Praks has participated in numerous remote sensing project and measurement campaigns, and led the development of Aalto-1 and Aalto-2 nanosatellites. He is a member of the ESA Advisory Committee on Education since 2014 and Finnish delegate in European Association of Remote Sensing Laboratories Council.