MEASUREMENT SCIENCE REVIEW, **20**, (2020), No. 5, 210–217

sciendo

ISSN 1335-8871

MEASUREMENT SCIENCE REVIEW

Journal homepage: https://content.sciendo.com

Using a Gauge Block for Derivation of Parameters of Four Laser Triangulation Sensors in a Local Coordinate System

Ján Buša[1], Miroslav Dovica[2], and Mikhail Zhabitsky[3]

[1]*Laboratory of Information Technologies, Joint Institute for Nuclear Research, Joliot-Curie st, 6, 141980 Dubna, Moscow Region, Russia, busaj@jinr.ru*
[2]*Institute of Special Engineering Processes, Department of Biomedical Engineering and Measurement, Faculty of Mechanical Engineering, Technical University in Košice, Letná, 9, 04001 Košice, Slovakia, miroslav.dovica@tuke.sk*
[3]*Dzelepov Laboratory of Nuclear Problems, Joint Institute for Nuclear Research, Joliot-Curie st, 6, 141980 Dubna, Moscow Region, Russia, mikhail.zhabitsky@jinr.ru*

The paper discusses the derivation of an accurate coordinate measuring system consisting of two, three, or four sensors based on the records of four fixed laser triangulation sensors done for a gauge block in movement. Three-dimensional case is considered. In the simulations, using a set of distances quadruplets, parameters of sensors in a local sensors coordinate system are determined through a least squares minimization process using the Differential Evolution approach. The influence of the measurement and rounding inaccuracy on the identification accuracy using numerical simulation methods are assessed.

Keywords: Metrology, laser triangulation sensor, gauge block, sensors coordinate system, minimization, Differential Evolution.

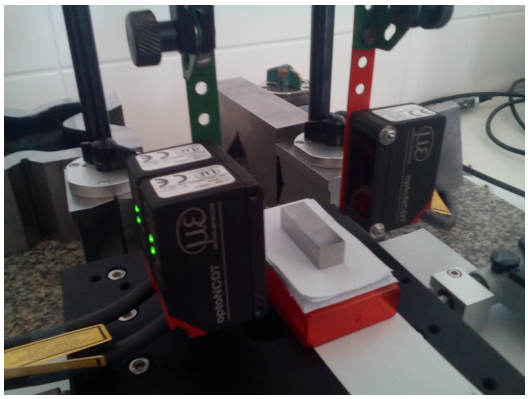

Fig.1. Three sensors and a gauge block

1. Introduction

The standard approach to the length measurements in coordinate metrology assumes a moving measuring system and a fixed workpiece [1]. In the present paper a quantitative characterization of the movement of a workpiece of interest is given based on records provided by a system of two, three, or four fixed laser triangulation sensors (LTS) [2, 3] (see Fig. 1) using the laser triangulation principle described in [4, 5]. Measurement setups using one LTS were already presented, e.g., a system for the real-time gauge measurement in a train [6] or a system for the contactless dynamic displacement measurement [7]. A measuring system for on-line quality control of car engine block has been presented in [8]. The measuring unit is considered as a set of several measuring modules where each of them acts like a single bore gauge and is arranged in four LTSs, which are installed on different positions and in opposite directions.

To our aim, we assume that the output of a sensor interface is provided by the distance to the point target along the sensor laser beam (see Figs. 2 and 3).

The usage of stationary sensor system records asks for the accurate knowledge of the positions of the sensors and the corresponding directions of their laser beams. In this paper we describe a simple method for the identification of sensor parameters in a fixed sensor coordinate system bound to the sensors based on a set of measurements done for different positions of a ***gauge block*** [9] – *a metal or ceramic block involving two parallel opposing faces the distances w between which is known with very high accuracy* (Figs. 1, 2, and 3). To our best knowledge such a task has not been considered before. However, the calibration of four LTSs presented in [8] is based on a similar idea using the standard engine block, whose objective parameters have been measured by a high-accuracy coordinate measuring machine (CMM). The authors also used the Differential Evolution (DE) method [10] to find the intrinsic parameters of LTSs. We used a different improvement of the DE method described below.

DOI: 10.2478/msr-2020-0026

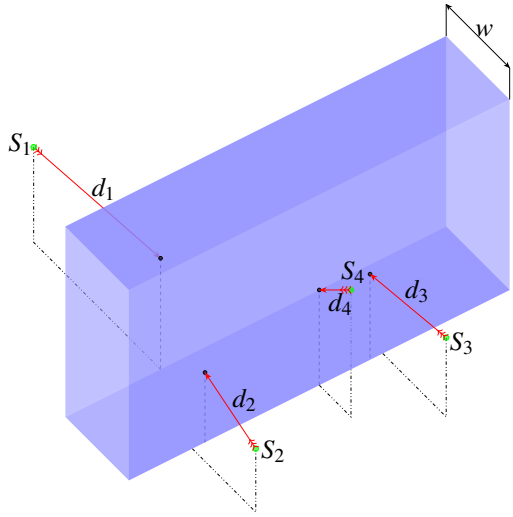

Fig.2. Sensor distances to the points on a gauge block – case 1

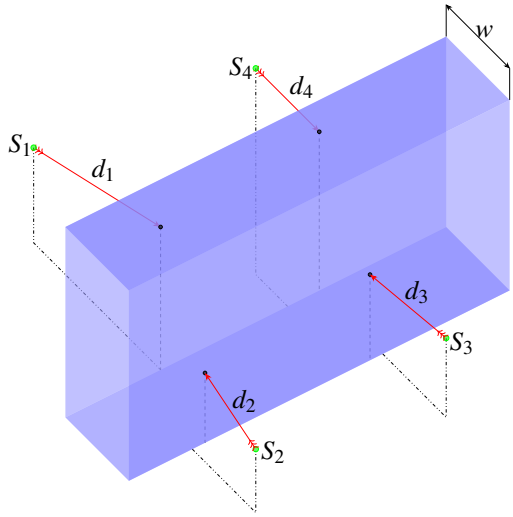

Fig.3. Sensor distances to the points on a gauge block – case 2

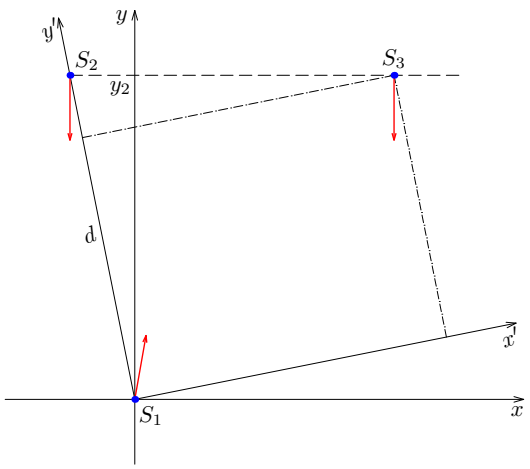

Fig. 4. Sensors positions and orientations in two local coordinate systems in 2D case

If three sensors are placed at the same vertical position, and the corresponding laser beams are all directed "horizontally" as in Fig. 1, the problem could be considered as two-dimensional (see Figs. 1 and 2 in [11]). In fact, the authors of [8] were using the assumption of four planar LTSs with beams lying in a common plane. In [11] a 6 parameters identification problem for the statement of a local coordinate system for three sensors in a plane has been considered. For this purpose a set of distance triplets (see Fig. 2 in [11]) for different (unknown) positions of a "gauge block" (rectangle in a plane) has been used.

However, the assumption formulated above is not realistic. In this paper we consider a solution of the 3D problem. Unlike the two-dimensional case, in that case the laser beam's direction is characterized by two (not one) angles which correspond to the longitude and latitude of a point on a sphere.

The discussion is divided into four parts: the problem formulation (Sec. 2), the problem solution (Sec. 3), numerical case studies – the results of simulations (Sec. 4), and conclusions (Sec. 5).

2. Formulation of the Problem

Given the set of four fixed sensors S_1, S_2, S_3, and S_4, with predefined but unknown positions and angular directions of their laser beams, and a certain position of the gauge block (like in Figs. 2 and 3), a foursome of distances $(d_1;d_2;d_3;d_4)$ measured along the laser beam from each sensor to the block, is recorded and stored. The N times iteration of the measuring process for different positions of the rectangular *gauge block of width w* provides the input of the problem, $(d_{1_i},d_{2_i},d_{3_i},d_{4_i})$, $i = 1, 2, \ldots, N$.

We have to determine the parameters (positions and the laser beams directions) of sensors S_1, S_2, S_3, and S_4 in some local coordinate system using the set of distances above.

2.1. Uniqueness of the Solution

Remark 2.1. *If one will try to solve the problem using only gauge block positions with a common vertical axis, then it is evident that independent vertically shifted sensors will give the same distances. So, in such case the solution cannot be unique. For more general but unknown positions of a gauge block, it will not be possible to determine the block face plane. Therefore four sensors should be used.*

Remark 2.2. *Due to the* unknown rectangles – "gauge block" – positions *the 2D problem does not have a unique solution – if one rotate and/or shift the coordinate system, the same set of distances will correspond to some rectangle's positions.*

Fixing the origin at the sensor S_1 and adding an additional condition, e.g., common x or y coordinate for two sensors (see Fig. 4) will result in a unique solution. Corresponding results of least squares minimization for "exact" and "rounded" triplets data for a case of common y coordinate of sensors S_2 and S_3 are presented in [11].

In 3D case we will consider two additional conditions together with fixed origin at point S_1:

1. we will consider the plane $z = 0$ to be a common plane for sensors S_1, S_2, and S_3;
2. we will take $x_2 = 0$, so $y_2 = d > 0$, where d will be the distance between S_1 and S_2.

Problem formulation. Given a set of N distances foursomes for unknown gauge block positions, determine 14 unknown parameters of four sensors in a local coordinate system

$$\begin{aligned} S_1 &= (0;0;0;\alpha_1;\beta_1), & S_2 &= (0;y_2;0;\alpha_2;\beta_2),\\ S_3 &= (x_3;y_3;0;\alpha_3;\beta_3), & S_4 &= (x_4;y_4;z_4;\alpha_4;\beta_4). \end{aligned} \tag{1}$$

Remark 2.3. *The angles α in (1) represent the azimuthal angle in the x–y plane of the local spherical coordinate system (for the x-axis this angle is 0, for the y-axis it is equal to $\pi/2$), the angles β represent the corresponding polar angle which is equal to $-\pi/2$ for the points lying on the negative half axis z, to $\pi/2$ on the positive half axis z, and 0 for points on the x–y plane. So α and β correspond to the longitude and latitude of a point on a sphere, respectively. The origin of the local coordinate system is placed at the sensor S_1, the y axis is determined by sensors S_1 and S_2, and the coordinate x–y plane and the z-axis are determined by the sensors S_1, S_2, and S_3 (see Figs. 2 and 3).*

Remark 2.4. *Solving the Problem, it is possible to use the results for the coordinate system statement for cases of only two or three (like on Fig. 1) sensors. For example, if one would like to use only two sensors for measurement, he/she could add temporarily two additional sensors, then measure distances for different positions of a gauge block, find the parameters, and in the following measurement use only sensors S_1 and S_2.*

3. Solution of the Problem

We consider two different sensor configurations. Case 1: sensor S_1 is placed on the opposite side of the gauge block as the other sensors S_2, S_3, and S_4 (see Fig. 2). Case 2: sensors S_1 and S_2 are placed on the opposite side as the sensors S_3 and S_4 (see Fig. 3). Under the above constraints on the definition of the coordinate system, the only *invariant feature associated to* any position *of the gauge block* follows from the fact that the distance from the target T_1 to the plane $\sigma(T_2,T_3,T_4)$ defined by three other targets for Case 1 (see Fig. 2) and the distance between two skew straight lines $\overline{T_1T_4}$ and $\overline{T_2T_3}$ for Case 2 (see Fig. 3) should equate w for all block placements.

In the subsection 3.2 the precision of the target location under the inaccuracies of the sensor parameters and measured distances is discussed.

We tried to solve the minimization problems (3) and (4) formulated below in Sec. 3.1 using "classical" (gradient, conjugate-gradient, the Newton) methods, however, we did not succeed. Relevant reasonable solution was achieved using a variant of the Differential Evolution algorithm, briefly described in the subsection 3.3 below.

3.1. Formulation of the Minimization Problem

Given the sensor parameters S_1, S_2, S_3, and S_4 (1) defined in Sec. 2, the values $\widetilde{w}_i = \widetilde{w}_i(S_1,S_2,S_3,S_4)$ (which denote the calculated distances from T_1 to the plane $\sigma(T_2,T_3,T_4)$ for Case 1 and the distance between two skew straight lines $\overline{T_1T_4}$ and $\overline{T_2T_3}$ for Case 2) calculated for the input foursomes $(d_{1_i},d_{2_i},d_{3_i},d_{4_i})$, $i = 1, 2, \ldots, N$, will be different from the true distance w and will depend on the accuracy of measurement of these foursomes.

So the deviations

$$r_i = \widetilde{w}_i(S_1,S_2,S_3,S_4) - w, \qquad i = 1,2,\ldots,N, \tag{2}$$

are introduced, and the problem solution is considered as the solution of the nonlinear least squares problem

$$(S_1^*,S_2^*,S_3^*,S_4*) = \operatorname{argmin} \min_{(S_1,S_2,S_3,S_4)} \sum_{i=1}^{N} r_i^2, \tag{3}$$

or the min-max problem:

$$(S_1^*,S_2^*,S_3^*,S_4*) = \operatorname{argmin} \min_{(S_1,S_2,S_3,S_4)} \max_{i=1,\ldots,N} |r_i|. \tag{4}$$

Remark 3.1. *If the sensor parameters are wrong, corresponding to the* false sensors system*, then significant nonzero deviations are expected. If the number N of the measurements is large enough, the best approximating values of the 14 parameters should be close to the "true" parameters of the original sensors system.*

3.2. Accuracy Issues

Let us consider a sensor $\bar{S}$ with the true parameters, and its approximation $\hat{S}$:

$$\bar{S} = [\bar{x}_S,\bar{y}_S,\bar{z}_S,\bar{\alpha},\bar{\beta}] \quad \text{and} \quad \hat{S} = [\hat{x}_S,\hat{y}_S,\hat{z}_S,\hat{\alpha},\hat{\beta}]. \tag{5}$$

If one knows the true distance $\bar{d}$ to the target point

$$\bar{T} = [\bar{x}_T,\bar{y}_T,\bar{z}_T], \tag{6}$$

then considering three coordinate functions $X(x,d,\alpha,\beta)$, $Y(y,d,\alpha,\beta)$, and $Z(z,d,\beta)$, the target coordinates are the following

$$\begin{aligned} \bar{x}_T &= X(\bar{x}_S,\bar{d},\bar{\alpha},\bar{\beta}) = \bar{x}_S + \bar{d}\cdot\cos\bar{\beta}\cdot\cos\bar{\alpha},\\ \bar{y}_T &= Y(\bar{y}_S,\bar{d},\bar{\alpha},\bar{\beta}) = \bar{y}_S + \bar{d}\cdot\cos\bar{\beta}\cdot\sin\bar{\alpha},\\ \bar{z}_T &= Z(\bar{z}_S,\bar{d},\bar{\beta}) = \bar{z}_S + \bar{d}\cdot\sin\bar{\beta}, \end{aligned} \tag{7}$$

and similar formulae define the coordinates of the approximate target $\hat{T}$ given for the approximate sensor $\hat{S}$ from (5) and the distance $\hat{d}$.

Using the Lagrange theorem for the function X we have

$$\hat{x}_T = X(\hat{x}_S,\hat{d},\hat{\alpha},\hat{\beta}) \overset{L}{=} X(\bar{x}_S,\bar{d},\bar{\alpha},\bar{\beta}) +$$

$$+\frac{\partial X}{\partial x}\bigg|_{[\bar{x}_S,\tilde{d},\tilde{\alpha},\tilde{\beta}]}\cdot(\hat{x}_S-\bar{x}_S)+\frac{\partial X}{\partial d}\bigg|_{[\bar{x}_S,\tilde{d},\tilde{\alpha},\tilde{\beta}]}\cdot(\hat{d}-\bar{d})+$$
$$+\frac{\partial X}{\partial \alpha}\bigg|_{[\bar{x}_S,\tilde{d},\tilde{\alpha},\tilde{\beta}]}\cdot(\hat{\alpha}-\bar{\alpha})+\frac{\partial X}{\partial \beta}\bigg|_{[\bar{x}_S,\tilde{d},\tilde{\alpha},\tilde{\beta}]}\cdot(\hat{\beta}-\bar{\beta})= \quad (8)$$
$$=\bar{x}_T+(\hat{x}_S-\bar{x}_S)+\cos\tilde{\alpha}\cos\tilde{\beta}(\hat{d}-\bar{d})-$$
$$-\tilde{d}\sin\tilde{\alpha}\cos\tilde{\beta}(\hat{\alpha}-\bar{\alpha})-\tilde{d}\cos\tilde{\alpha}\sin\tilde{\beta}(\hat{\beta}-\bar{\beta}).$$

For the y coordinate we get

$$\hat{y}_T=Y(\hat{y}_S,\hat{d},\hat{\alpha},\hat{\beta})\overset{L}{=}$$
$$\overset{L}{=}\bar{y}_T+(\hat{y}_S-\bar{y}_S)+\sin\tilde{\alpha}\cos\tilde{\beta}(\hat{d}-\bar{d})+ \quad (9)$$
$$+\tilde{d}\cos\tilde{\alpha}\cos\tilde{\beta}(\hat{\alpha}-\bar{\alpha})-\tilde{d}\sin\tilde{\alpha}\sin\tilde{\beta}(\hat{\beta}-\bar{\beta}).$$

Finally, for the z coordinate we have

$$\hat{z}_T\overset{L}{=}\bar{z}_T+(\hat{z}_S-\bar{z}_S)+\sin\tilde{\beta}(\hat{d}-\bar{d})+\tilde{d}\cos\tilde{\beta}(\hat{\beta}-\bar{\beta}). \quad (10)$$

The values $\tilde{d}$, $\tilde{\alpha}$, and $\tilde{\beta}$ in (8), (9), and (10) are, in general, different. We will use these equations in the next section, when we discuss the precision of the numerical solution.

3.3. *Brief Description of the Asynchronous Differential Evolution Method*

Differential Evolution (DE) — is an efficient method to solve global optimization problems [10, 12, 13]. The method uses *a population* of N_p vectors $\vec{s}_i$, which represents candidate solutions in the search domain Ω. A steady-state DE variant — Asynchronous Differential Evolution with Adaptive Correlation Matrix (ADE-ACM) [14] is used in this work to iteratively improve a candidate solution by applying evolutionary operations of *mutation, crossover and selection* over population members.

In steady-state strategy one can pick-up a random member from the population as *a target vector* $\vec{s}_i$ [15]. *A mutant vector* $\vec{v}_i$ is formed by simple arithmetic combination of three randomly selected distinct population members (rand/1 DE-strategy):

$$\vec{v}_i=\vec{s}_r+F(\vec{s}_p-\vec{s}_q), \quad (11)$$

where *a scale factor* F is sampled for each iteration from a Cauchy distribution $C_F(\mu_F,\gamma_F=0.1)$. The location parameter μ_F is updated according to values of the scale factor, which results in a population's improvement.

Canonical Differential evolution [12] uses *uniform crossover*: *a trial vector* $\vec{u}_i$ is created from the target vector $\vec{s}_i$ by replacing several of its coordinates by corresponding coordinates from a mutant vector (11). An average number of replaced coordinates (C_rN_p) depends on *crossover rate* $C_r\in[0,1]$. At least one coordinate is replaced. Optimal value of crossover rate depends on the problem: $C_r\to 0$ leads to faster convergence for separable problems, while $C_r\to 1$ should be used for non-separable objective functions.

ADE-ACM [14] uses a different approach to perform crossover: the crossover operator is modified to take into account *pairwise dependencies* between the variables. The current population is used to calculate *a sample correlation matrix* $\mathscr{C}$:

$$\mathscr{C}_{jk}=\frac{q_{jk}}{\sqrt{q_{jj}q_{kk}}};\quad q_{jk}=\frac{1}{N_p-1}\sum_{i=0}^{N_p-1}(s_{ij}-\langle s\rangle_j)(s_{ik}-\langle s\rangle_k), \quad (12)$$

where $\langle s\rangle_j$ denotes the population averaging of the j-th variable of population vectors $\vec{s}_i$, $i=0,1,\ldots N_p-1$. The approximation of the correlation matrix is cumulatively adapted through successful evolutionary steps:

$$C\leftarrow(1-l)C+l\mathscr{C},\quad l=0.01. \quad (13)$$

Based on the acquired *adaptive correlation matrix*, groups of dependent variables can be identified and crossover is applied in the corresponding subspace of the original domain Ω. Thanks to this crossover scheme, the ADE-ACM competitively solves a wide range of global optimization problems, both separable and non-separable [16]. For partially-separable problems ADE-ACM shows both better convergence probability and faster convergence rate than DE variants with uniform crossover.

A greedy algorithm is used for *selection*: a trial vector $\vec{u}_i$ replaces the target vector $\vec{s}_i$, if it results in improvement of the objective function value.

During successive operations ADE controls spreads of population members in each coordinate and in the function values. If any spread is less than a small predefined value, *an independent restart* is initiated [17]. The ADE is started with $N_p=20$, at each restart the size of population is doubled.

Minimization is performed until the predefined maximal number of function evaluations is reached. To ensure convergence to the global minimum, 10 independent trials are used. The 7 best results are used to estimate uncertainty in the value of the found minimum. If the precision is not sufficient, the procedure is repeated with a larger budget of maximal number of function evaluations.

4. Numerical Case Studies

In this section we present the results of numerical simulations. Two sensor systems $\mathscr{S}_1$ (14) and $\mathscr{S}_2$ (15) for Case 1 and one system $\mathscr{S}_3$ (16) for Case 2 have been considered:

$$S_1=\left(0;0;0;\tfrac{\pi}{3};\tfrac{\pi}{4}\right),\quad S_2=\left(0;50;0;\tfrac{5\pi}{3};0\right),$$
$$S_3=\left(20;50;0;\tfrac{3\pi}{2};0\right),\quad S_4=\left(-20;50;20;\tfrac{3\pi}{2};0\right), \quad (14)$$

$$S_1=\left(0;0;0;\tfrac{\pi}{3};\tfrac{\pi}{4}\right),\quad S_2=\left(0;50;0;\tfrac{5\pi}{3};\tfrac{\pi}{6}\right),$$
$$S_3=\left(40;45;0;\tfrac{5\pi}{4};\tfrac{\pi}{6}\right),\quad S_4=\left(20;50;30;\tfrac{3\pi}{2};-\tfrac{\pi}{6}\right), \quad (15)$$

$$S_1=\left(0;0;0;\tfrac{\pi}{2};0\right),\quad S_2=\left(0;50;0;\tfrac{3\pi}{2};0\right),$$
$$S_3=\left(20;50;0;\tfrac{3\pi}{2};0\right),\quad S_4=\left(20;0;10;\tfrac{\pi}{2};0\right). \quad (16)$$

The first three coordinates are in millimeters (mm), two angular coordinates are in radians.

We used randomly modified parameters using the uniform distribution (within radius of 1 mm for the first three coordinates and 0.0175 radian $\approx$ 1 degree for angular coordinates)

to model the inaccuracy of parameter determination. The parameters of three systems which we considered as "original" are listed below (here and in the following text the values are rounded to fit the text width). Parameters of particular sensors are presented in the columns.

```
   S1           S2           S3           S4
 0            0          19.98883825 -19.99255062
 0           49.95691998  49.95906517  50.05072563
 0            0           0           19.91418540
 1.57832969   4.70825814   4.70321121   4.71034922
-0.00177239   0.00936077   0.00670605  -0.00376562

   S1           S2           S3           S4
 0            0          40.06409233  19.91224330
 0           50.05905562  44.92675448  50.08682815
 0            0           0           29.95563246
 1.03162095   5.24669058   3.93927376   4.69646645
 0.78408640   0.51227262   0.52808819  -0.53736586

   S1           S2           S3           S4
 0            0          20.05295680  19.97718986
 0           49.97694430  49.93759188  -0.04898805
 0            0           0           9.94829850
 1.57834019   4.70170318   4.70150254   1.58741101
-0.01258636  -0.00769893  -0.01512428  -0.00716478
```

4.1. Simulation Data Generation

In our numerical experiments we used the data sets with different number N of a "gauge block" position. To identify 14 parameters it is evident that N cannot be a small number. For $N = 27$ the results were unsatisfactory, good results were obtained, e.g., for $N = 135$. Below in Sec. 4.3 the results for $N = 54$ will be presented.

For distance generation and also for some estimates of the resulting precision we used GNU Octave [18] – an open source alternative to the proprietary programming language MATLAB [19]. Our scripts and functions could be, of course, used in MATLAB as well. For the minimization this program was too slow, and we prepared a C++ package briefly presented below in Sec. 4.4.

For this purpose we generated data for a "gauge block" position using the next Octave code:

```
% 54 distances foursomes generation
for k=1:3, % 3 alpha
  for m=1:3, % 3 beta
    for l=1:3, % 3 x,y,z
      nd=nd+1; % distances counter
      alpha=pi/2+pi/12*(2-k);
      beta=pi/12*(2-m);
      cpoint=[0;20;0]+(2-l)*[0;15;0];
      ... distances foursame calculation
      alpha=pi/2+pi/12*(2-k);
      beta=pi/6*(2-m);
      cpoint=[15;18;0]+(2-l)*[0;8;0];
      ... distances foursame calculation
    end
  end
end
```

A "gauge block" is placed in 6 positions (using 2 basic positions and two shifts of each one) and in each position it is rotated around the center in two angular directions – for each direction 3 different angles are used.

Using the same principle, several data sets have been considered. For each position of the "gauge block" all 4 distances to the sensors are calculated, rounded (see Sec. 4.3), and stored.

4.2. Double Precision Distances Data

In this section only one result for the system $\mathscr{S}_1$ will be presented. $N = 54$ "gauge block" positions have been used for generation of foursomes of distances. The distances stored in the double precision (further considering as exact) have been used to determine the parameters of the sensor system. The system (14) is considered as "ideal", and its perturbed system above as "original".

Three matrices are presented below. The first is the difference between the "original" and "ideal" system $\mathscr{S}_1$ (let us recall that the first three lines are in mm). The second matrix is the result of the calculations. Indeed, the average of the 7 from 10 best solutions is used and presented here. This matrix could be compared with the "original" system $\mathscr{S}_1$ above. Finally, the third matrix represents the difference between the resulting matrix and the "original". For the x, y, and z coordinates the worst case is 0.0000217 mm $\approx$ 22 nm for the y coordinate of the sensor S_4. The maximum angular difference 0.00000044 rad $\approx$ 0.000025 degree.

```
   Sensors differences ideal-original
 0.00000000   0.00000000   0.01116175  -0.00744938
 0.00000000   0.04308002   0.04093483  -0.05072563
 0.00000000   0.00000000   0.00000000   0.08581460
-0.00753336   0.00413084   0.00917777   0.00203976
 0.00177239  -0.00936077  -0.00670605   0.00376562

   Averaged optimal sensors
 0.00000000   0.00000000  19.98884170 -19.99255112
 0.00000000  49.95692049  49.95908683  50.05074732
 0.00000000   0.00000000   0.00000000  19.91419721
 1.57832953   4.70825807   4.70321122   4.71034925
-0.00177224   0.00936059   0.00670571  -0.00376606

   Sensors differences optimal-original
 0.00000000   0.00000000   0.00000345  -0.00000050
 0.00000000   0.00000051   0.00002166   0.00002170
 0.00000000   0.00000000   0.00000000   0.00001181
-0.00000016  -0.00000007   0.00000001   0.00000003
 0.00000015  -0.00000018  -0.00000034  -0.00000044
```

4.3. Distances Data Rounded to 0.1 μm

In this subsection we will present the results of numerical simulations using distance foursomes rounded to 0.1 μm. Only results for selected data sets of size $N = 54$ will be shown.

At first let us present the result for one simulation concerning the system $\mathscr{S}_1$.

```
   Averaged optimal sensors
 0.00000000   0.00000000  19.98862833 -19.99080958
 0.00000000  49.95700254  49.96434449  50.05016400
 0.00000000   0.00000000   0.00000000  19.92080476
```

Table 1. Upper bounds in mm for target inaccuracies for system $\mathscr{S}_1$

	S_1	S_2	S_3	S_4
Δx_T	0.0009661	0.0011004	0.0086499	0.0044867
Δy_T	0.0000620	0.0001490	0.0053898	0.0025396
Δz_T	0.0000078	0.0000060	0.0000695	0.0066510

Table 2. Upper bounds in mm for target inaccuracies for system $\mathscr{S}_2$

	S_1	S_2	S_3	S_4
Δx_T	0.0023199	0.0018423	0.0065804	0.0028651
Δy_T	0.0026404	0.0026216	0.0088654	0.0035003
Δz_T	0.0017003	0.0008761	0.0012513	0.0018348

Table 3. Upper bounds in mm for target inaccuracies for system $\mathscr{S}_3$

	S_1	S_2	S_3	S_4
Δx_T	0.0003852	0.0003079	0.0059452	0.0024507
Δy_T	0.0000699	0.0001060	0.0015320	0.0012659
Δz_T	0.0000091	0.0000142	0.0001895	0.0084260

```
 1.57831038   4.70823615   4.70322171   4.71036112
-0.00175725   0.00933902   0.00656800  -0.00397468

   Sensors differences optimal-original
 0.00000000   0.00000000  -0.00020992   0.00174103
 0.00000000   0.00008256   0.00527933  -0.00056163
 0.00000000   0.00000000   0.00000000   0.00661936
-0.00001931  -0.00002200   0.00001049   0.00001190
 0.00001514  -0.00002176  -0.00013805  -0.00020906
```

The solution presented in the matrix above could be compared with the parameters of the "original" system. The differences presented in the bottom matrix show a very good precision for the second sensor S_2, the worst precision is about 5.3 μm for the y coordinate of the sensor S_3.

It is important to have the upper bounds of the inaccuracies by the determination of a target point T for some sensor S and corresponding measured distance d between S and T. For example, using (8), for the inaccuracy of the x coordinate of the target T

$$|\hat{x}_T - \bar{x}_T| \le |\Delta x_S| + |\cos\tilde{\alpha}\cos\tilde{\beta}| \cdot |\Delta d| + \tilde{d}|\sin\tilde{\alpha}\cos\tilde{\beta}| \cdot |\Delta\alpha| + \tilde{d}|\cos\tilde{\alpha}\sin\tilde{\beta}| \cdot |\Delta\beta|.$$

Using, in this case, $|\Delta d| \le 0.00005$, and

$$|\cos\tilde{\alpha}| \le 0.01,\ |\sin\tilde{\alpha}| \le 1,\ |\cos\tilde{\beta}| \le 1,\ |\sin\tilde{\alpha}| \le 0.003$$

for the first sensor S_1 we get the upper bound

$$|\hat{x}_T - \bar{x}_T| < 0.00097 \text{ mm} = 0.97\ \mu\text{m}.$$

In Table 1 the upper bounds for all three coordinate inaccuracies for target T at distance $d < 50$ mm from each of sensors $S_1 - S_4$ are given. These upper bounds are taken as the worst cases from results calculated for 3 different data sets of size $N = 54$.

These results could be improved if we consider distances d smaller then 50 mm, and also using more accurate upper bounds for sin and cos function values.

We would like to note that the target location precision for the first two sensors S_1 and S_2 is much better than the results for the sensors S_3 and S_4. The same applies to the accuracy of the sensor S_1 and S_2 parameters themselves.

For the next two cases we will give the corresponding results without the detailed description in the text.

Now, let us present the results for the sensor system $\mathscr{S}_2$.

```
   Sensors differences ideal-original
 0.00000000   0.00000000  -0.06409233   0.08775670
 0.00000000  -0.05905562   0.07324552  -0.08682815
 0.00000000   0.00000000   0.00000000   0.04436754
 0.01557660  -0.01070282  -0.01228295   0.01592253
 0.00131176   0.01132616  -0.00448941   0.01376709

   Original sensors
 0.00000000   0.00000000  40.06409233  19.91224330
 0.00000000  50.05905562  44.92675448  50.08682815
 0.00000000   0.00000000   0.00000000  29.95563246
 1.03162095   5.24669058   3.93927376   4.69646645
 0.78408640   0.51227262   0.52808819  -0.53736586

   Averaged optimal sensors
 0.00000000   0.00000000  40.06564663  19.91403358
 0.00000000  50.05960123  44.92786857  50.08810241
 0.00000000   0.00000000   0.00000000  29.95386548
 1.03159297   5.24672452   3.93927302   4.69644309
 0.78403271   0.51224814   0.52806036  -0.53732810

   Sensors differences optimal-original
 0.00000000   0.00000000   0.00155430   0.00179028
 0.00000000   0.00054561   0.00111409   0.00127426
 0.00000000   0.00000000   0.00000000  -0.00176698
-0.00002798   0.00003394  -0.00000075  -0.00002336
-0.00005369  -0.00002448  -0.00002783   0.00003776
```

In Table 2 the upper bounds for all three coordinate inaccuracies for a target T from each of the sensors $S_1 - S_4$ are given.

Finally, Case 2 sensor system $\mathscr{S}_3$ will be considered.

```
   Sensors differences ideal-original
 0.00000000   0.00000000  -0.05295680   0.02281014
 0.00000000   0.02305570   0.06240812   0.04898805
 0.00000000   0.00000000   0.00000000   0.05170151
-0.00754386   0.01068580   0.01088644  -0.01661468
 0.01258636   0.00769893   0.01512428   0.00716478

   Original sensors
 0.00000000   0.00000000  20.05295680  19.97718986
 0.00000000  49.97694430  49.93759188  -0.04898805
 0.00000000   0.00000000   0.00000000   9.94829849
 1.57834019   4.70170318   4.70150254   1.58741101
-0.01258636  -0.00769893  -0.01512428  -0.00716478

   Averaged optimal sensors
 0.00000000   0.00000000  20.05163295  19.97772896
 0.00000000  49.97690208  49.93893530  -0.05003985
 0.00000000   0.00000000   0.00000000   9.95560399
 1.57833250   4.70169704   4.70155845   1.58744917
-0.01260239  -0.00768075  -0.01502769  -0.00741663

   Sensors differences optimal-original
 0.00000000   0.00000000  -0.00132385   0.00053911
 0.00000000  -0.00004222   0.00134342  -0.00105180
 0.00000000   0.00000000   0.00000000   0.00730549
-0.00000769  -0.00000614   0.00005590   0.00003816
-0.00001603   0.00001818   0.00009659  -0.00025184
```

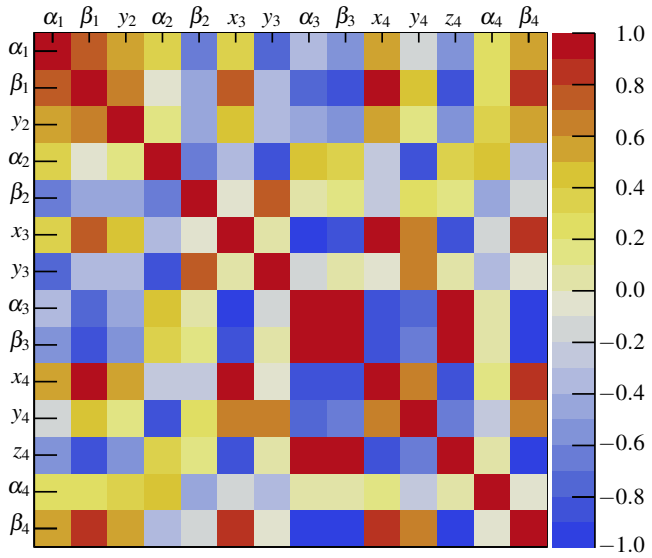

Fig.5. Adaptive correlation matrix (13) near the global minimum

In Table 3 the upper bounds for all three coordinate inaccuracies for target T from each of the sensors S_1 – S_4 are given. In this case, similar to both systems considered before, the difference between the inaccuracies, upper bounds for the first two sensors S_1 and S_2 and the sensors S_3 and S_4 is even more pronounced.

4.4. *C++ Program Description, Implementation, and Performance*

The C++ implementation of the ADE-ACM method was used to find out positions of the sensors. For efficient load of multicore processors, two parallel variants were tested: OpenMP (Open Multi-Processing) and MPI (Message Passing Interface). The program implements asymmetric Primary/Secondary model: multiple secondary processes are used to evaluate objective function values (each candidate solution by a separate process), while the primary process implements the ADE-ACM algorithm, issues candidate solutions and collects results from secondary processes.

Three tasks of size $N = 54$ for which the results were presented above were solved on the 2C/4T Intel(R) Core(TM) i5-420M CPU @ 2.50Ghz system

$\mathscr{S}_1$: `nfe 300 000`, `OMP_NUM_THREADS=1` – 10 s, `OMP_NUM_THREADS=2` – 7 s, `mpirun -np 2` – 12 s.

$\mathscr{S}_2$: `nfe 2 300 000`, `OMP_NUM_THREADS=1` – 68 s, `OMP_NUM_THREADS=2` – 51 s, `mpirun -np 2` – 91 s.

$\mathscr{S}_3$: `nfe 300 000`, `OMP_NUM_THREADS=1` – 11 s, `OMP_NUM_THREADS=2` – 7 s, `mpirun -np 2` – 14 s.

The main difficulty of the minimization process arises from correlation (or anti-correlation) between parameters (Fig. 5).

In addition to the already described elements of the program, we have implemented a simple (and also another slightly more complicated) test that proves that for the resulting "optimal" sensor system with specified parameters for each rounded distance foursome, there is a position of the "gauge block" for which the exact distances after rounding match the specified ones. The principle of the tests is to check whether within the specified distance precision exist two distance foursomes for which the resulting values r_i (2) are both negative and positive. All 9 resulting systems presented above did pass the simpler test successfully.

5. Discussion / Conclusions

Presented results show the possibility of using the proposed method to solve the sensor parameters identification tasks. Of course, only simulation results were discussed. In case of interest, real measurements can be used instead of simulated distance data sets. In such a situation, further study of the gauge block positions will be necessary.

From the point of view of accuracy, the case of identifying the parameters of the system of two sensors by temporarily supplementing the other two sensors is particularly interesting. For less precise distances measured, corresponding, e.g., to rounding to 1 μm instead of 0.1 μm considered above, evidently using larger values N will be necessary. "Nowadays, it is possible to find LTS with resolution up to 0.01 μm. However, the need of higher resolution comes into conflict with the measuring range of the transducer." – state the authors of [7].

The computation time is not long, if we take into account that the task would be solved once in identifying the parameters of the installed sensor system.

Finally, let us recall that for the measured distance foursomes it is not necessary to determine and coordinate the positions of the gauge block in the local coordination system.

Acknowledgement

Work supported within the project VEGA 1/0224/18 "Research and development of testing and measuring methods in coordinate metrology". The authors are grateful to Gheorghe Adam for fruitful discussion and advice.

References

[1] Weckenmann, A., Kraemer, P., Hoffmann, J. (2007). Manufacturing metrology – state of the art and prospects. In: 9th *International Symposium on Measurement and Quality Control (9th ISMQC)*. IMEKO.

[2] MTI Instruments, Inc. (2020) Laser triangulation sensors. `https://www.mtiinstruments.com/technology-principles/laser-triangulation-sensors/`.

[3] Micro-Epsilon Messtechnik. (2020). Micro-epsilon sensors. `https://www.micro-epsilon.com/`.

[4] Berkovic, G., Shafir, E. (2012). Optical methods for distance and displacement measurements. *Advances in Optics and Photonics*, 4(4), 441–471. `https://doi.org/10.1364/AOP.4.000441`.

[5] MTI Instruments Inc. (2019). An introduction to laser triangulation sensors. https://www.azosensors.com/article.aspx?ArticleID=523.

[6] Zhang, Z., Feng, Q., Gao, Z., Kuang, C., Fei, C., Li, Z., Ding, J. (2008). A new laser displacement sensor based on triangulation for gauge real-time measurement. *Optics & Laser Technology*, 40, 252–255. https://doi.org/10.1016/j.optlastec.2007.04.009.

[7] Soave, E., D'Elia, G., Mucchi, E. (2020). A laser triangulation sensor for vibrational structural analysis and diagnostics. *Measurement and Control*, 53(1-2), 73–82. https://doi.org/10.1177/0020294019877484.

[8] Li, X.-Q., Wang, Z., Fu, L.-H. (2016). A laser-based measuring system for online quality control of car engine block. *Sensors*, 16(11), 1877. https://doi.org/10.3390/s16111877.

[9] Wikimedia Foundation. (2020). Gauge block. https://en.wikipedia.org/wiki/Gauge_block.

[10] Storn, R. M., Price, K. V. (1995). *Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces.* International Computer Science Institute (ICSI), TR-95-012.

[11] Buša, J., Dovica, M., Kačmár, L. (2018). Derivation of a coordinate system of three laser triangulation sensors in a plane. In *Numerical Methods and Applications: 9th International Conference.* Springer, LNCS 11189, 64–71. https://doi.org/10.1007/978-3-030-10692-8_7.

[12] Price, K. V., Storn, R. M., Lampinen, J. A. (2005). *Differential Evolution. A Practical Approach to Global Optimization*. Springer, https://doi.org/10.1007/3-540-31306-0.

[13] Das, S., Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1), 4–31. https://doi.org/10.1109/TEVC.2010.2059031.

[14] Zhabitskaya, E., Zhabitsky, M. (2013). Asynchronous differential evolution with adaptive correlation matrix. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO'13)*. Association for Computing Machinery (ACM), 455–462. https://doi.org/10.1145/2463372.2463428.

[15] Zhabitskaya, E., Zhabitsky, M. (2012). Asynchronous differential evolution. In *Mathematical Modeling and Computational Science: International Conference (MMCP 2011)*. Springer, LNCS 7125, 328–331. https://doi.org/10.1007/978-3-642-28212-6_41.

[16] Zhabitsky, M. (2016). Comparison of the asynchronous differential evolution and jade minimization algorithms. *EPJ Web of Conferences*, 108, 02048. https://doi.org/10.1051/epjconf/201610802048.

[17] Zhabitskaya, E., Zhabitsky, M. (2013). Asynchronous differential evolution with restart. In: *Numerical Analysis and Its Applications: 5th International Conference (NAA 2012)*. Springer, LNCS 8236, 555–561. https://doi.org/10.1007/978-3-642-41515-9_64.

[18] Eaton, J.W. (2020). GNU Octave. https://www.gnu.org/software/octave/.

[19] The MathWorks, Inc. (2020). MATLAB. https://ch.mathworks.com/products/matlab.html.

Received July 6, 2020
Accepted September 30, 2020