Analysis of post-processing methods of IMU MEMS cluster of autonomous navigation of ground transport systems

Fedor Sorokin\textsuperscript{1,2}, Roman Kiselev\textsuperscript{2*}, and Alena Golovleva\textsuperscript{3}

\textsuperscript{1}Applied Mechanics Department, Bauman Moscow State Technical University, Moscow, 105005, Russia
\textsuperscript{2}Theory of Mechanisms and Machines Department, Bauman Moscow State Technical University, Moscow, 105005, Russia
\textsuperscript{3}Nuclear Reactors and Installations Department, Bauman Moscow State Technical University, Moscow, 105005, Russia

Abstract. The work is devoted to the description of algorithms and schemes of blocks of sensitive elements (IMU MEMS cluster), applicable for navigation of ground autonomous systems. The work contains the principal description of schemes and methods applicable to a cluster consisting of 32 inertial navigation systems of rough accuracy and the study of the effectiveness of each method and each scheme by simulation mathematical modeling. 3 schematic diagrams of building an INS MEMS cluster consisting of 32 sensors are described. 3 methods of post-processing of excessive inertial information are described: the method of linear averaging, quadratic averaging and the method of interpolation. The mathematical regularities underlying each of the methods are described. A mathematical model is described that allows evaluating the effectiveness of postprocessing schemes and methods. Based on the mathematical model, conclusions are drawn about the operation of methods and schemes of the INS MEMS cluster of 32 sensors. Recommendations are given for choosing the scheme and methods of postprocessing redundant information when building a cluster solution.

1 Introduction

The operation of modern automated systems is impossible without the use of body position sensors in inertial space. Among the huge variety of inertial sensors, microelectromechanical systems (MEMS) have been increasingly distinguished in recent years [1]. MEMS sensors are inertial information sensors that combine a mechanical component (sensitive elements of gyroscopes and accelerometers) and electronics [2].

MEMS sensors have found wide application in various fields of technology: portable devices [3], precision machines [4, 5], unmanned aerial vehicles [6, 7] etc. MEMS sensors have found wide application due to their advantages: low cost, small dimensions,
adaptability, high frequency of information transmission. At the same time, MEMS sensors also have a number of significant disadvantages: low accuracy (relative to analogues, especially in the field of angular velocity sensors), high dependence of accuracy characteristics on external factors, accumulation of error over time.

As already mentioned, the main problem of MEMS sensors is relatively low accuracy. As a result, the main direction of research in the field of MEMS sensors is to increase the accuracy characteristics. There are many approaches to solving this problem. Shkel, et. al. offers completely new materials for the manufacture of sensitive elements of MEMS sensors [8]. In [9], it is proposed to change the topology of the sensitive elements of MEMS gyroscopes and accelerometers according to the criterion of reducing energy dissipation. In the work [10], a new inverse connection system is proposed that allows compensation of sensor errors from external influences. In the works [11], it is proposed to use additional sources of inertial information (global Positioning System, gravity maps of the terrain) for the purpose of dynamic calibration of sensors. Relatively recently, a fundamentally new approach to solving the task has appeared – combining several MEMS sensors into a single system in order to obtain excessive inertial information [12]. In recent years, many works have been devoted to this issue [13, 14]. The accuracy of such systems is increased by processing excessive inertial information. However, the question of algorithms for the operation of such systems is currently poorly understood.

In this paper, the main aspects of processing excessive inertial information were considered. Inertial measurement systems (IMU) with MEMS sensors and clusters based on them were chosen as the object of research. Autonomous ground-transport systems are selected as the object of application: autonomous transport, wheeled robots, etc. The basic algorithms of post-processing of information about the operation of IMU MEMS clusters are considered. Their methodology is described, a mathematical model of their work is compiled. On the basis of a mathematical model, the effectiveness of the described methods of post-processing of excessive inertial information is evaluated. Conclusions are drawn about the effectiveness of the considered methods. In the last decade, inertial measurement unit (IMU) have become widespread in the field of short-term navigation [15]. The main purpose of the IMU is to determine the coordinates of an object in inertial space. The advantages of the IMU include: autonomy (especially outside the scope of GPS), high protection from external influences, high frequency of data transmission [16].

Many devices require the use of small-sized IMU with low power consumption, but do not require very high measurement accuracy or require short-term high accuracy (up to an hour). Such devices include: sensors on the links of machines and manipulators, mobile devices, unmanned aerial vehicles, cruise missiles and other systems. The requirements of such devices are met by IMU based on micromechanical systems (MEMS), such IMU have the following advantages: low cost, small dimensions, low power consumption, the possibility of compact placement of components in one housing. However, IMU MEMS have high random errors of the output signals, which accumulate over time of operation, and a pronounced dependence of the errors on temperature, pressure, time and external influences.

Due to its pronounced shortcomings, it was long believed that IMU MEMS would not find wide application and would occupy only a small niche of mobile devices. However, in recent years, many papers have been written [17, 18] proving the possibility of widespread use of IMU MEMS due to advanced achievements in the field of micromechanics.

2 Methodology for improving accuracy characteristics based on IMU

One of the methods of increasing the accuracy characteristics of the MEMS IMU is to combine $N$ IMU into one system in order to obtain excessive inertial information about the
position of an object in inertial space, which allows to increase the accuracy characteristics up to $\sqrt{N}$ times [19].

Modern calibration methods make it possible to eliminate systematic errors with high accuracy. Many imperfections, such as temperature deformations, axis distortions, static drift, and others, are eliminated by calibration and feedback in the system. Random errors are difficult to compensate for, and cluster solutions are used to reduce their level in order to increase accuracy up to the level of $D_R/\sqrt{N}$, where $D_R$ is the variance of sensor readings caused by random errors.

This work is devoted to the description of various methods of processing inertial information of a cluster solution based on inertial navigation systems. As an illustration for each of the methods, we will use a cluster system of 32 IMU.

Three variants of multi-level cluster schemes will be considered for each of the methods (Figure 1). The letters "D" denotes detector from 1 to 32, the letters "C" denote controllers that process incoming inertial information. The simplest circuit is the one in Figure 1.a, all sensors are connected directly to the controller. Figure 1.b shows a two-level circuit, in which intermediate controllers C1, C2, C3 and C4 are added. Figure 1.b shows a three-level scheme.

![Fig. 1. Diagrams of clusters of 32 sensors (a) 1st order, (b) 2nd order and (3) 3rd order.](image)

### 2.1 Data processing

As part of the work, the following assumptions were made:
1. Sensor readings are random independent quantities;
2. For \( N \to \infty \), the nature of the distribution of random variables tends to the normal distribution;
3. Sensor readings do not depend on time.

The whole logic of building a cluster solution and its processing is primarily based on GOST R 8.736-2011. In accordance with it, the following must be carried out for data processing:
1. Eliminate systematic errors (calibration, inverse connection);
2. Calculate the arithmetic mean of the observation results;
3. Evaluate the mean square deviation of the sensor readings;
4. Check the assumption about the normal law of the distribution of quantities;
5. Calculate the confidence interval of random measurement errors;
6. Calculate the boundaries of the non-excluded systematic measurement error;
7. Calculate the confidence limits of the measurement error.

Data processing on the above points is necessary not so much for implementation into the averaging algorithm, as for evaluating the effectiveness of eliminating systematic errors and the quality of sensor operation in principle. This mathematical evaluation should be carried out after calibration and establishment of feedbacks. Feedbacks in this paper mean dynamic calibration of functional dependences of quantities on temperature, pressure, vibrations and other measurable factors of external and internal influences.

### 2.2 Measurement calculation methodology

To obtain the measurement result according to GOST R 8.736-2011, it is assumed that gross errors are excluded in accordance with the measurement calculation method. Since there is no single evaluation methodology for the operation of cluster systems, we introduce the following evaluation algorithm:
1. Calculate the arithmetic mean of the measurements:
   \[
   \bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}
   \]  
   (1)
2. Calculate the mean square deviation:
   \[
   S = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N(N-1)}}
   \]  
   (2)
3. For each measurement, we calculate the deviation of the measurements from the arithmetic mean:
   \[
   V_i = \frac{|x_i - \bar{x}|}{S}
   \]  
   (3)
4. Gross errors are excluded:
   1) if \( 20 < N < 50 \), then at \( V_i > 3 \) then \( x_i \) should be recognized as a miss and excluded from consideration;
   2) if \( N < 20 \), the measurement \( x_i \) should be considered a miss, such that \( V_i > 2 \), or use the Chauvenet’s criterion [20];
   3) for a small number of measurements, the following criteria should be applied:
      \[
      V_i > \begin{cases} 
      1.6N = 3, 4 \\
      1.7N = 5, 6, 7 \\
      1.9N = 8, 9 \\
      2N = 10, 11 
      \end{cases}
      \]  
      (4)
   4) since classical clusters have \( 4 < N < 50 \), consideration of methods for estimating misses for \( N > 50 \) is not required.

Suppose, based on the measurement results, we dropped \( K \) statistically insignificant measurements, then the number of measurements decreased to \( \tilde{N} = N - K \). If \( K > \frac{N}{2} \), then
either the significance criterion was chosen incorrectly, or the sensors require repeated calibration.

Next, the question arises of finding the true value of the measured value. According to GOST R 8.736-2011, the measurement result should be the arithmetic mean of the corrected measurements:

\[ \bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \]  

(5)

However, this is only one of the methods of estimating the true value. With a small number of sensors, the arithmetic mean can give a very large deviation from the true value, in which case, for example, a standard estimate can be used, which reduces the contribution of the readings furthest from the true value:

\[ \bar{x} = \sqrt{\frac{\sum_{i=1}^{N} (x_i)^2}{N}} \]  

(6)

With a large number of sensors, interpolation methods can be used, for example, the principle of least squares. Interpolation methods are universal and allow obtaining sufficiently accurate values even with an abnormal distribution of random variables.

Thus, for 32 sensors, at least 3 circuits can be made, for which there are at least 3 methods for processing. A logical question arises about which scheme and method are the most effective for a set of 32 sensors.

### 3 Research of a mathematical model by generating random normal values

To evaluate the operation of each scheme and method, we will construct a normal distribution of quantities. Let the true value of \( x_{tr} = 10 \) (it is also a mathematical expectation), and the standard deviation \( \sigma = 1 \). The normal distribution function has the form:

\[ f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-x_{tr})^2}{2\sigma^2}} \]  

(7)

The graph of the function for the given \( \sigma \) and \( x_{tr} \) is shown in Figure 2. According to this normal law, 32 random variables were generated, theoretically corresponding to the readings of 32 sensors. Next, calculations were carried out using 3 algorithms for three schemes, the results of calculations for are shown in Table 1.

![Graph of the normal distribution function](image)

**Fig. 2.** The normal distribution of random variables at \( \sigma = 1 \) and \( x_{tr} = 10 \).
Table 1. Sensor readings obtained by generating random variables.

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
<th>x8</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.55</td>
<td>11.17</td>
<td>11.21</td>
<td>9.15</td>
<td>10.43</td>
<td>7.24</td>
<td>8.33</td>
<td>11.35</td>
</tr>
<tr>
<td>x9</td>
<td>x10</td>
<td>x11</td>
<td>x12</td>
<td>x13</td>
<td>x14</td>
<td>x15</td>
<td>x16</td>
</tr>
<tr>
<td>x17</td>
<td>x18</td>
<td>x19</td>
<td>x20</td>
<td>x21</td>
<td>x22</td>
<td>x23</td>
<td>x24</td>
</tr>
<tr>
<td>10.73</td>
<td>9.57</td>
<td>11.72</td>
<td>12.64</td>
<td>10.9</td>
<td>10.45</td>
<td>8.75</td>
<td>10.14</td>
</tr>
<tr>
<td>x25</td>
<td>x26</td>
<td>x27</td>
<td>x28</td>
<td>x29</td>
<td>x30</td>
<td>x31</td>
<td>x32</td>
</tr>
<tr>
<td>8.89</td>
<td>11.13</td>
<td>9.21</td>
<td>9.75</td>
<td>10.84</td>
<td>9.16</td>
<td>12.04</td>
<td>10.35</td>
</tr>
</tbody>
</table>

To carry out the calculations, a program was written in C++, which generates the values of random parameters and performs calculations for the selected cluster scheme using three methods, the calculation results are shown in Table 2, where \( x_{ij} \) is the result of the cluster of the \( i \)-th order, by the \( j \)-th method (\( j = 1 \) corresponds to the linear averaging method, \( j = 2 \) the mean quadratic averaging method and \( j = 3 \) the interpolation method).

Table 2. Results of calculations according to the parameters of Table 1.

<table>
<thead>
<tr>
<th>( x_{11} )</th>
<th>( x_{12} )</th>
<th>( x_{13} )</th>
<th>( x_{21} )</th>
<th>( x_{22} )</th>
<th>( x_{23} )</th>
<th>( x_{31} )</th>
<th>( x_{32} )</th>
</tr>
</thead>
</table>

The average value of the sensor readings is \( \bar{x} = 10.1799 \). In this particular example, the 1st order scheme with averaging of readings after removing statistically insignificant sensor readings, as described above, proved to be the most effective. However, it is impossible to judge the effectiveness of the schemes and the method by only one sample of sensor readings, therefore, \( 10^6 \) generations were made for normal distributions with \( x_t = 10 \) and standard deviation \( \sigma = 4 \) to evaluate the methods. Since at the moment the interpolation method is not applicable in practice due to the specifics of the operation of microcontrollers (lack of computing power), the interpolation method was not considered. The calculation results are shown in Tables 3, 4 and 5. \( \Delta x_{\text{minj}} \) is the smallest deviation of the cluster solution from the true value obtained by the \( j \)-method, \( \Delta x_{\text{maxj}} \) is the largest deviation, \( x_{avj} \) is the average value of the cluster solution, \( k_{\text{impj}} \) is the coefficient of cluster accuracy improvement (expected improvement \( k_{\text{impj}} = \sqrt{32} \)). The efficiency coefficient can be used to judge the rationality of the scheme and method. In real systems, it is expected that the efficiency coefficient is 20-30% less than the theoretical one. This is due to the correlation between individual sensor systems, which is not taken into account by the theoretical model.

Table 3. Calculation results using scheme 1.

<table>
<thead>
<tr>
<th>( \Delta x_{\text{min1}} )</th>
<th>( \Delta x_{\text{min2}} )</th>
<th>( \Delta x_{\text{max1}} )</th>
<th>( \Delta x_{\text{max2}} )</th>
<th>( x_{av1} )</th>
<th>( x_{av2} )</th>
<th>( k_{\text{imp1}} )</th>
<th>( k_{\text{imp2}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 3.5\times10^{-7} )</td>
<td>( 3.5\times10^{-6} )</td>
<td>4.03</td>
<td>4.11</td>
<td>10</td>
<td>10.07</td>
<td>5.72</td>
<td>5.72</td>
</tr>
</tbody>
</table>

Table 4. Calculation results using scheme 2.

<table>
<thead>
<tr>
<th>( \Delta x_{\text{min1}} )</th>
<th>( \Delta x_{\text{min2}} )</th>
<th>( \Delta x_{\text{max1}} )</th>
<th>( \Delta x_{\text{max2}} )</th>
<th>( x_{av1} )</th>
<th>( x_{av2} )</th>
<th>( k_{\text{imp1}} )</th>
<th>( k_{\text{imp2}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1.4\times10^{-6} )</td>
<td>( 2.5\times10^{-7} )</td>
<td>10</td>
<td>10</td>
<td>9.977</td>
<td>9.986</td>
<td>5.4</td>
<td>5.45</td>
</tr>
</tbody>
</table>

Table 5. Calculation results using scheme 3.

<table>
<thead>
<tr>
<th>( \Delta x_{\text{min1}} )</th>
<th>( \Delta x_{\text{min2}} )</th>
<th>( \Delta x_{\text{max1}} )</th>
<th>( \Delta x_{\text{max2}} )</th>
<th>( x_{av1} )</th>
<th>( x_{av2} )</th>
<th>( k_{\text{imp1}} )</th>
<th>( k_{\text{imp2}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 3.6\times10^{-7} )</td>
<td>( 1.3\times10^{-6} )</td>
<td>10</td>
<td>10</td>
<td>10.06</td>
<td>10.07</td>
<td>5.38</td>
<td>5.33</td>
</tr>
</tbody>
</table>
As you can see from the tables for the random generation of $10^6$ cluster system launches, 1 method applied to 1 scheme behaves most stably. Moreover, it can be noticed that the more we split the scheme into separate cluster sections, i.e. we increase its order, the worse the calculation results become. This is explained by the fact that we do not examine all the received inertial information in its entirety, but examine it in small groups, thereby we do not exclude cases when a separate group has deviated greatly from the true value in one direction and there is a natural loss of significant measurements, for example, if in scheme 3 a group of four sensors takes the following values: 6, 7, 8, 9, if the true value is 10, then we will exclude significant measurements from the solution and get an additional error.

4 Conclusion

So, within the framework of the work, 3 schemes and 3 methods were described that can be used to build a cluster solution based on a block of sensitive elements consisting of 32 sensors. The main mathematical regularities that are necessary for the formation of the algorithm and the working program of the cluster were described. The cluster system operation was simulated by generating $10^6$ cluster system launches under the assumptions made at the beginning of the work. As a result of the work, we can come to the following conclusions:

1. If possible, excessive inertial information should be considered in aggregate, which is a consequence of the greater efficiency of scheme 1 in comparison with scheme 2 and the greater efficiency of scheme 2 in comparison with scheme 3;
2. In cases where it is not possible to connect all sensors to one controller (currently 1 microcontroller can process inertial information from no more than 64 sensors), the scheme in which the smallest order of the cluster system is implemented should be used;
3. The most effective method of averaging sensor readings (an algorithm for postprocessing excess inertial information) is the linear averaging method, the method of root-mean-square averaging has not shown sufficient effectiveness;
4. The most promising method is the interpolation method, however, at the moment the method is not applicable due to the limited RAM of microcontrollers).

The authors express their gratitude to the leadership of the Bauman Moscow State Technical University and The Theory of Mechanisms and Machines Department in particular for their support.

References