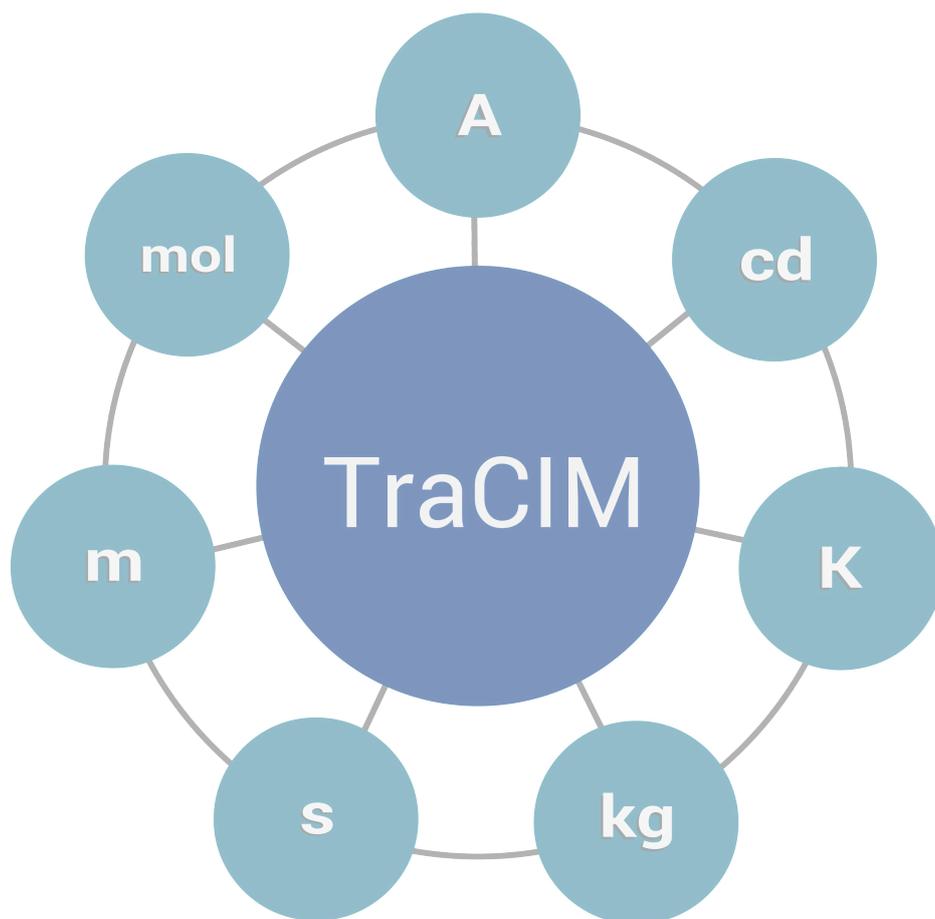


Traceability for Computationally-Intensive Metrology

Validation of Three-Rosette Method Algorithm Implementation



Contents

1	Introduction	2
2	Test result evaluation	2
3	Client-server communication	2
3.1	Development of a client application	3
3.2	Public sample data for client testing	4
4	XML message content	4
4.1	TraCIM XML schemata	4
4.2	Test data	5
4.3	Test results	6
4.4	Test report message	8
4.5	TraCIM error message	8
5	How to get support	8
A	TraCIM and Three-Rosette Method XML schemata	9
A.1	General TraCIM schema	9
A.2	Test schema for Three-Rosette Method	10
A.3	Result schema for Three-Rosette Method	11

1 Introduction

The Three-Rosette Method test is provided by the PTB TraCIM Online System (tracim.ptb.de) and can be ordered by registered customers. To perform a test the customer will get an XML file with data sets containing synthetically generated data in a specified format. Task of the customer is to process this data according to the Computational Aims [1] for this task and send back the results to the TraCIM system. The TraCIM system automatically compares the customer results with its own reference values and sends a report with the test conclusion. Evaluation procedures are described in Section 2.

The data exchange with the TraCIM system is provided by a RESTful web service. All data is exchanged in XML format. For the communication with the TraCIM system, it is recommended to use a client application which also takes care of the (de)serialization of the XML strings. In Section 3.1 support for developing such a client can be found. The XML schemata used for data exchange are described in Section 4.

Tests can be bought in a web shop (tracim.ptb.de) by registered customers. A test consists of ten test data sets, each representing the data for a pitch deviation measurement according to the error separating method described in [1]. After successful purchase of a test the customer will get an order key that allows him to request test data from the TraCIM system with his client application.

In order to check the functionality of the customer's TraCIM client, a charge free test with public sample data is provided. A registered customer is allowed to request a test with these data sets from the TraCIM system at any time in order to evaluate the correct function of his client-server communication. See Section 3.2 on how to request sample test data sets.

The PTB TraCIM system was developed within the frame of an EURAMET project founded by the European Union (EMRP project NEW06: Traceability for computationally-intensive metrology, www.ptb.de/emrp/tcim.html) and is under strict quality control of TraCIM e.V. Association.

2 Test result evaluation

For the validation, the test results calculated by the customer software are compared to the associated reference results in the TraCIM system data base. For a positive test conclusion all deviations between customer values and reference values must be smaller than the maximum permissible error values (MPE). This MPE, which is the same for all test values, is specified by the customer.

- If the customer specifies an MPE equal to zero, the default value $MPE = 10^{-6}$ mm is used.
- Admissible non-zero values specified by the customer must satisfy

$$10^{-12} \text{ mm} \leq MPE \leq 10^{-6} \text{ mm}. \quad (2.1)$$

Other values for the MPE are not accepted and cause an error message.

3 Client-server communication

The test procedure is highly automated using internet based data exchange by client-server communication via a REST interface. Using a REST client application the user is able to obtain test data sets and send back the calculated results.

The data exchanged between client and server is encapsulated within XML. A proper specification of the applied XML data schemata is given in Section 4. The messages send between the client application and the TraCIM system server are treated as plain character strings.

3.1 Development of a client application

Configuration of the HTTPS connection: The communication between the TraCIM server and the client application is established via HTTPS (Hypertext Transfer Protocol Secure = encrypted HTTP) connection that allows to send and receive content in the form of character strings containing messages in XML format. Each HTTPS connection is created from a specific URL (Uniform Resource Locator). The following configurations are necessary:

- Set the request method “POST” (request comprising input and output)
- Set connection property “Content-Type” to “application/xml”
- Set connection property “Accept” to “application/xml”

Packages for creation and configuration of an HTTPS connection are available for different programming languages, e.g.:

- Java: java.net API
- C/C++: Microsoft C++ REST SDK or similar
- C#: .NET Framework (System.Net.Http)

POST request for obtaining test data sets: The URL of an HTTPS connection for the POST request to obtain test data sets is

`https://tracim.ptb.de/tracim/api/order/<VERROVER_ORDER_KEY>/test` (3.1)

where `<VERROVER_ORDER_KEY>` has to be replaced by the order key for the Three-Rosette Method test purchased at the TraCIM web shop. When the TraCIM system receives the client message it will create a unique process key associated with this test process. Test data and process key are then returned to the client. In case of an incorrect order key the TraCIM server will send an error message. Details about the format of the delivered test data are described in Section 4.2. Error messages are addressed in Section 4.5.

Example 3.1. *A simple way to communicate with the TraCIM server is using the command line tool `cURL` (<https://curl.haxx.se>). To get test data for the order key `<VERROVER_ORDER_KEY>` and save the response to the file `testdata.xml` one can use the following command:*

```
curl -X POST https://tracim.ptb.de/tracim/api/order/<VERROVER_ORDER_KEY>/test
-o testdata.xml
```

(In case one gets an error about unknown CA certificates or similar, one may need to download the certificate bundle `cacert.pem` from <https://curl.haxx.se/docs/caextract.html> and use additionally the option `--cacert cacert.pem`.)

POST request for sending test results and obtaining the certificate: When the customer has processed the test data sets according to the Computational Aims, he has to prepare the XML content with his results as described in Section 4.3. The URL of the HTTPS connection for a POST request in order to send the prepared XML string is

`https://www.tracim.ptb.de/tracim/api/test/<PROCESS_KEY>` (3.2)

where `<PROCESS_KEY>` has to be replaced by the individual process key that was returned by the TraCIM system together with the test data. The TraCIM system will then evaluate the content

and generate a test report. It states whether the test is passed, or otherwise contains a list of the values which didn't pass the test for all elements V01-V10. (In case of a free sample test the report is not signed and does not contain the PTB seal.) The report is encoded in XML and sent to the customer as return message to the POST request (3.2). See Section 4.4 for a description of the XML content returned by the TraCIM system. As soon as the test report is sent to the customer, the associated process key cannot be used again, irrespective of whether the test was passed or not.

Example 3.2. *To send the file `results.xml` with the calculated results for the process with key `<PROCESS_KEY>` to the TraCIM server and save the server response to the file `validation.xml` with the command line tool `cURL` use*

```
curl -d @calculated_result.xml -H "Content-Type: application/xml"
-X POST https://tracim.ptb.de/tracim/api/test/<PROCESS_KEY>
-o validation.xml
```

(As in Example 3.1 one may also need to use the option `--cacert cacert.pem`.)

For the case of an improper process key or an malformed or incomplete XML content the TraCIM system returns an error message, see Section 4.5 for a description. In this case, the associated process key is still valid, so the customer can fix the problems in the XML content and send the results again.

Remark 3.3. *After receiving an order key a customer has a total of 200 days for performing the Three-Rosette Method test. The TraCIM system will occasionally send warning messages to the customers e-mail address stating the remaining time for the order. A final information message is sent three days before the order expires.*

3.2 Public sample data for client testing

Any registered customer can order sample test data free of charge with unlimited request amount for backtracking errors within the client application that could compromise a commercial test. In comparison to a test with commercial test data the certificate returned by the server is not countersigned by PTB as legally valid certificate. Sample order keys are available in the webshop (tracim.ptb.de) for registered customers.

4 XML message content

4.1 TraCIM XML schemata

The XML schemata which specify the format of the XML files for the data exchange with the TraCIM system can be obtained from the following URLs:

- General TraCIM report schema
<https://tracim.ptb.de/tracim/api/schema/tracim.xsd>
- Three-Rosette Method test data schema
https://tracim.ptb.de/tracim/api/schema/PTB_MATH_VERROVER_v1_test.xsd
- Three-Rosette Method result data schema
https://tracim.ptb.de/tracim/api/schema/PTB_MATH_VERROVER_v1_result.xsd

All messages delivered by the TraCIM server have the root element `tracim` of type `tracimMessage` defined in the general TraCIM report schema. These messages may contain order or process information, test data sets according to the Three-Rosette Method test data schema, test conclusions, or error messages.

The only XML messages send by the customer are the test results according to the Three-Rosette Method result data schema with root element `verRoVerResultPackage`.

For the reader's convenience these schemata are included in Appendix A. It is recommended to make use of this schemata during client development, i.e. to automatically build Java, C++ or C# classes.

4.2 Test data

Test data returned by the TraCIM system are composed of the three major elements order identification, process identification and test data sets. The order element contains the order key, the date of the creation of the order and the date for the expiration of the order.

The process element contains the process key associated with the test data request. Finally, the test element contains the test data for the Three-Rosette Method test, consisting of ten test data sets with the IDs V01-V10. For each test data set, the toothlength, the number of relative positions, the relative positions and the data for the cumulative pitch deviations for the relative positions are provided.

Below is an excerpt of an test data XML string delivered by the TraCIM system for the POST request (3.1):

```
<?xml version="1.0" encoding="UTF-8"?>
<tracim:tracim xmlns:tracim="http://tracim.ptb.de/tracim"
               xmlns:verrover="http://tracim.ptb.de/verrover/test"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <tracim:order>
    <tracim:key>[ORDER_KEY]</tracim:key>
    <tracim:creationDate>[...]</tracim:creationDate>
    <tracim:expirationDate>[...]</tracim:expirationDate>
  </tracim:order>
  <tracim:process>
    <tracim:key>[PROCESS_KEY]</tracim:key>
  </tracim:process>
  <tracim:test xsi:type="verrover:verRoVerTestPackage">
    <verrover:testElement>
      <verrover:basicID>V01</verrover:basicID>
      <verrover:toothlength>20</verrover:toothlength>
      <verrover:noRelPos>20</verrover:noRelPos>
      <verrover:relPos>
        <verrover:pos>0</verrover:pos>
        <verrover:pos>1</verrover:pos>
        [...]
      </verrover:relPos>
      <verrover:measurementResults>
        <verrover:column>
          <verrover:pitchDev> 0.0041675583826800</verrover:pitchDev>
          <verrover:pitchDev> 0.0058374338702700</verrover:pitchDev>
          [...]
        </verrover:column>
        <verrover:column>
          <verrover:pitchDev> 0.0051613447037000</verrover:pitchDev>
          <verrover:pitchDev> 0.0079450089121300</verrover:pitchDev>
```

```

        [...]
    </verrover:column>
</verrover:measurementResults>
</verrover:testElement>
<verrover:testElement>
    <verrover:basicID>V02</verrover:basicID>
    [...]
</verrover:testElement>
</tracim:test>
</tracim:tracim>

```

The process key [PROCESS_KEY] is created by the TraCIM system when the test data is delivered. This key has to be used by the customer when sending the test results as described in the next section.

The test data contains 10 Three-Rosette Method test data sets V01-V10. For each test data set the `toothlength`, the number of relative positions (`noRelPos`), and the relative positions (`relPos`) are given. Finally, `measurementResults` contains the (synthetic) data for the test. For each relative position there is one column of cumulative pitch deviations of length `toothlength`, so `measurementResults` can be seen as `toothlength` \times `noRelPos` matrix. The unit of the pitch deviations is mm.

4.3 Test results

The calculated results of the software under test must be send to the TraCIM system as XML string compliant with the XML schema for the result data (see 4.1, result data schema). The client has to specify the following information:

- [PROCESS_KEY]: process key received with the test data (section 4.2)
- [SOFTWARE_VENDOR]: software vendor (can be different from customer name)
- [SOFTWARE_NAME]: name of software under test
- [SOFTWARE_VERSION]: version of software under test
- [SOFTWARE_REVISION]: revision of software under test (optional)
- [MPE]: MPE (in mm) for all test values (set the MPE to 0.0 to use the default values)

This information is followed by elements that contain the results of the software under test for the basic IDs V01-V10:

- `cumulativePitchDevA`: cumulative pitch deviations of the gear (divided circle *A*)
- `cumulativePitchDevB`: cumulative pitch deviations of divided circle *B*
- `cumulativePitchDevC`: cumulative pitch deviations of divided circle *C*
- `singlePitchDevA`: single pitch deviations of the gear (divided circle *A*)
- `singlePitchDevB`: single pitch deviations of divided circle *B*
- `singlePitchDevC`: single pitch deviations of divided circle *C*
- `sigma`: standard deviation of the single measurements

- `sigmaCumulativePitchDevA`: maximum standard deviation of the cumulative pitch deviations of the gear
- `sigmaSinglePitchDevA`: standard deviation of the single pitch deviations of the gear

Below is an excerpt of the result XML string which has to be send back to the TraCIM server using the POST request (3.2).

```
<?xml version="1.0" encoding="UTF-8"?>
<verRoVerResultPackage xmlns="http://tracim.ptb.de/verrover/result">
  <processKey>[PROCESS_KEY]</processKey>
  <softwareVendor>[SOFTWARE_VENDOR]</softwareVendor>
  <softwareName>[SOFTWARE_NAME]</softwareName>
  <softwareVersion>[SOFTWARE_VERSION]</softwareVersion>
  <softwareRev>[SOFTWARE_REVISION]</softwareRev>
  <mpe>[MPE]</mpe>
  <results>
    <basicID>V01</basicID>
    <cumulativePitchDevA>
      <A>0.003016577163048</A>
      <A>0.005842443714513</A>
      [...]
    </cumulativePitchDevA>
    <cumulativePitchDevB>
      <B>0.00108168971475</B>
      <B>0.0021203602717245</B>
      [...]
    </cumulativePitchDevB>
    <cumulativePitchDevC>
      <C>-0.000345608355143</C>
      <C>-0.0016770575558685</C>
      [...]
    </cumulativePitchDevC>
    <singlePitchDevA>
      <a>0.003016577163048</a>
      <a>0.002825866551465</a>
      [...]
    </singlePitchDevA>
    <singlePitchDevB>
      <b>0.00108168971475</b>
      <b>0.0010386705569745</b>
      [...]
    </singlePitchDevB>
    <singlePitchDevC>
      <c>-0.000345608355143</c>
      <c>-0.0013314492007255</c>
      [...]
    </singlePitchDevC>
    <sigma>0.00034996010615617789</sigma>
    <sigmaCumulativePitchDevA>0.00011066710256478359</sigmaCumulativePitchDevA>
    <sigmaSinglePitchDevA>0.00011066710256478359</sigmaSinglePitchDevA>
  </results>
  <results>
    <basicID>V02</basicID>
    [...]
  </results>
</verRoVerResultPackage>
```

4.4 Test report message

After evaluation of the test results the TraCIM system returns the validation conclusion as XML string according to the general TraCIM XML schema. It comprises the three elements:

- **passed**: true, if the software passed the test successfully, else false
- **report**: character string with short report on the test evaluation
- **reportPDF**: character string with the test report PDF (Base64-encoded)

In order to create the test report PDF document as PDF file, the Base64-encoded character string contained in the **reportPDF** tag has to be decoded and written to a new file with the proper **.pdf** file name extension. Below is an excerpt of an XML string with the test conclusion:

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<tracim:tracim xmlns:tracim="http://tracim.ptb.de/tracim"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <tracim:validation>
    <tracim:passed>true</tracim:passed>
    <tracim:report>Congratulations! You passed the test.</tracim:report>
    <tracim:reportPDF>JVBERi0xLjQKJWlsZX[...]</tracim:reportPDF>
  </tracim:validation>
</tracim:tracim>
```

4.5 TraCIM error message

The following XML code is sent by the server in case of errors, where [error code] and [error description] are replaced with values for the particular error.

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<tracim:tracim xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:tracim="http://tracim.ptb.de/tracim">
  <tracim:error>
    <tracim:code>[error code]</tracim:code>
    <tracim:description>[error description]</tracim:description>
  </tracim:error>
</tracim:tracim>
```

In case an error message is received, the error description may contain useful hints what caused the error.

5 How to get support

- technical support for client software: support.tracim@ptb.de
- support regarding computational aims: info.tracim@ptb.de
- Provider support (contact, fees, etc.): info.tracim@ptb.de

A TraCIM and Three-Rosette Method XML schemata

In the following, the XML schematas relevant for the Three-Rosette Method test are presented.

A.1 General TraCIM schema

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  version="1.0" targetNamespace="http://tracim.ptb.de/tracim"
  xmlns:tracim="http://tracim.ptb.de/tracim"
  xmlns:tns="http://tracim.ptb.de/tracim"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<xs:element name="tracim" type="tns:tracimMessage"/>
<xs:complexType name="tracimMessage">
  <xs:sequence>
    <xs:element name="order" type="tns:order" minOccurs="0"/>
    <xs:element name="process" type="tns:process" minOccurs="0"/>
    <xs:element name="test" type="tns:test" minOccurs="0"/>
    <xs:element name="validation" type="tns:validation" minOccurs="0"/>
    <xs:element name="error" type="tns:error" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="order">
  <xs:all>
    <xs:element name="key" type="xs:string"/>
    <xs:element name="creationDate" type="xs:dateTime"/>
    <xs:element name="expirationDate" type="xs:dateTime"/>
  </xs:all>
</xs:complexType>
<xs:complexType name="process">
  <xs:sequence>
    <xs:element name="key" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="test" abstract="true">
  <xs:sequence/>
</xs:complexType>
<xs:complexType name="validation">
  <xs:all>
    <xs:element name="passed" type="xs:boolean"/>
    <xs:element name="report" type="xs:string"/>
    <xs:element name="reportPDF" type="xs:string"/>
  </xs:all>
</xs:complexType>
<xs:complexType name="error">
  <xs:all>
    <xs:element name="code" type="xs:string"/>
    <xs:element name="description" type="xs:string"/>
  </xs:all>
</xs:complexType>
<xs:complexType name="result" abstract="true">
  <xs:sequence/>
</xs:complexType>
</xs:schema>
```

A.2 Test schema for Three-Rosette Method

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified" version="1.0"
  targetNamespace="http://tracim.ptb.de/verrover/test"
  xmlns:verrover="http://tracim.ptb.de/verrover/test"
  xmlns:tns="http://tracim.ptb.de/verrover/test"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tracim.ptb.de/tracim"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <xs:import namespace="http://tracim.ptb.de/tracim"/>

  <xs:element name="verRoVerTestPackage" type="tns:verRoVerTestPackage"/>
  <xs:complexType name="verRoVerTestPackage">
    <xs:complexContent>
      <xs:extension base="ns1:test">
        <xs:sequence>
          <xs:element name="testElement" type="tns:verRoVerTest" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="verRoVerTest">
    <xs:sequence>
      <xs:element name="basicID" type="xs:string"/>
      <xs:element name="toothlength" type="xs:int"/>
      <xs:element name="noRelPos" type="xs:int"/>
      <xs:element name="relPos" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="pos" type="xs:int" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="measurementResults" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="column" type="tns:column" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="column">
    <xs:sequence>
      <xs:element name="pitchDev" type="xs:double" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A.3 Result schema for Three-Rosette Method

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema attributeFormDefault="unqualified"
  elementFormDefault="qualified" version="1.0"
  targetNamespace="http://tracim.ptb.de/verrover/result"
  xmlns:verrover="http://tracim.ptb.de/verrover/result"
  xmlns:tns="http://tracim.ptb.de/verrover/result"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="http://tracim.ptb.de/tracim"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <xs:import namespace="http://tracim.ptb.de/tracim"/>

  <xs:element name="verRoVerResultPackage" type="tns:verRoVerResultPackage"/>
  <xs:complexType name="verRoVerResultPackage">
    <xs:complexContent>
      <xs:extension base="ns1:result">
        <xs:sequence>
          <xs:element name="processKey" type="xs:string"/>
          <xs:element name="softwareVendor" type="xs:string"/>
          <xs:element name="softwareName" type="xs:string"/>
          <xs:element name="softwareVersion" type="xs:string"/>
          <xs:element name="softwareRev" type="xs:string" nillable="true"/>
          <xs:element name="mpe" type="xs:double"/>
          <xs:element name="results" type="tns:verRoVerResult" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="verRoVerResult">
    <xs:sequence>
      <xs:element name="basicID" type="xs:string"/>
      <xs:element name="cumulativePitchDevA" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="A" type="xs:double" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="cumulativePitchDevB" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="B" type="xs:double" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="cumulativePitchDevC" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="C" type="xs:double" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="singlePitchDevA" minOccurs="0">
        <xs:complexType>
```

```

    <xs:sequence>
      <xs:element name="a" type="xs:double" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="singlePitchDevB" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="b" type="xs:double" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="singlePitchDevC" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="c" type="xs:double" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="sigma" type="xs:double"/>
<xs:element name="sigmaCumulativePitchDevA" type="xs:double"/>
<xs:element name="sigmaSinglePitchDevA" type="xs:double"/>
</xs:sequence>
</xs:complexType>
</xs:schema>

```

References

- [1] Computational Aims: Reduced error separating method for pitch deviation measurements

Computational Aims

Reduced error separating method for pitch deviation measurements

Contents

1	Language	2
2	Title	2
3	Keywords	2
4	Mathematical area	2
5	Dependencies	2
6	Input parameters	2
6.1	Input parameter 1	2
6.2	Input parameter 2	2
6.3	Input parameter 3	3
7	Output parameters	3
7.1	Output parameter 1	3
7.2	Output parameter 2	3
7.3	Output parameter 3	3
7.4	Output parameter 4	3
7.5	Output parameter 5	3
7.6	Output parameter 6	4
7.7	Output parameter 7	4
7.8	Output parameter 8	4
7.9	Output parameter 9	4
8	Mathematical model	4
8.1	Cumulative pitch deviations	4
8.2	Single pitch deviations	5
8.3	Standard deviations	5
9	Signature	5
10	References	5
11	History	6

1 Language

English (en)

2 Title

Reduced error separating method for pitch deviation measurements

3 Keywords

- Gear measurement
- Pitch deviations
- Three-rosette method
- Error separation method

4 Mathematical area

Linear least squares.

5 Dependencies

None.

6 Input parameters

6.1 Input parameter 1

Symbol: N
Description: toothnumber
Type: \mathbb{N}
Shape: scalar
Constrains: $N \geq 3$

6.2 Input parameter 2

Symbol: $q = (q_1, q_2, \dots, q_R)$
Description: relative positions between divided circles \mathcal{A} and \mathcal{B}
Type: \mathbb{N}
Shape: vector of dimension R
Constrains:

- $3 \leq R \leq N$
- $0 \leq q_r < q_s \leq N - 1$ for all $1 \leq r < s \leq R$
- $\gcd(q_2 - q_1, q_3 - q_1, \dots, q_R - q_1, N) = 1$, where gcd denotes the greatest common divisor

6.3 Input parameter 3

- Symbol:** $W = (W_{i,q_r})$ for $i = 0, 1, \dots, N - 1$, and $r = 1, 2, \dots, R$
Description: measured cumulative pitch deviations of the gear, measured in R different relative positions
Type: \mathbb{R}
Shape: matrix of dimension $N \times R$
Constrains: none

7 Output parameters

7.1 Output parameter 1

- Symbol:** $A = (A_0, \dots, A_{N-1})$
Description: cumulative pitch deviations of the gear (divided circle \mathcal{A})
Type: \mathbb{R}
Shape: vector of dimension N
Constrains: $A_{N-1} = 0$.

7.2 Output parameter 2

- Symbol:** $B = (B_{q_1}, \dots, B_{q_R})$
Description: cumulative pitch deviations of divided circle \mathcal{B}
Type: \mathbb{R}
Shape: vector of dimension R
Constrains: $B_{q_R} = 0$.

7.3 Output parameter 3

- Symbol:** $C = (C_0, \dots, C_{N-1})$
Description: cumulative pitch deviations of divided circle \mathcal{C}
Type: \mathbb{R}
Shape: vector of dimension N
Constrains: $C_{N-1} = 0$.

7.4 Output parameter 4

- Symbol:** $a = (a_0, \dots, a_{N-1})$
Description: single pitch deviations of the gear (divided circle \mathcal{A})
Type: \mathbb{R}
Shape: vector of dimension N
Constrains: $\sum_{j=0}^{N-1} a_j = 0$.

7.5 Output parameter 5

- Symbol:** $b = (b_{q_1}, \dots, b_{q_R})$
Description: single pitch deviations of divided circle \mathcal{B}
Type: \mathbb{R}
Shape: vector of dimension R
Constrains: $\sum_{r=1}^R b_{q_r} = 0$.

7.6 Output parameter 6

Symbol: $c = (c_0, \dots, c_{N-1})$
Description: single pitch deviations of divided circle \mathcal{C}
Type: \mathbb{R}
Shape: vector of dimension N
Constrains: $\sum_{j=0}^{N-1} c_j = 0$.

7.7 Output parameter 7

Symbol: σ
Description: standard deviation of the single measurements
Type: \mathbb{R}
Shape: scalar
Constrains: $\sigma \geq 0$

7.8 Output parameter 8

Symbol: σ_A
Description: maximum value of the standard deviation of the cumulative pitch deviations A
Type: \mathbb{R}
Shape: scalar
Constrains: $\sigma_A \geq 0$

7.9 Output parameter 9

Symbol: σ_a
Description: standard deviation of the single pitch deviations a
Type: \mathbb{R}
Shape: scalar
Constrains: $\sigma_a \geq 0$

8 Mathematical model

8.1 Cumulative pitch deviations

The output parameters A, B and C , and an additional scalar K are defined by

$$(A, B, C, K) = \underset{(A', B', C', K')}{\operatorname{argmin}} \sum_{j=0}^{N-1} \sum_{r=1}^R (A'_j + B'_{q_r} + C'_{j-q_r \bmod N} + K' - W_{j, q_r})^2, \quad (8.1)$$

subject to

$$A_{N-1} = B_{q_R} = C_{N-1} = 0. \quad (8.2)$$

Here, A denotes the cumulative pitch deviations of the gear, while B and C are cumulative pitch deviations of the divided circles \mathcal{B} and \mathcal{C} related to the measuring instrument. For example, \mathcal{C} might be the divided circle realized by a rotary table, and \mathcal{B} by the manual rotation of the gear on the rotary table.

Equation (8.1) together with the constrains (8.2) defines a linear least squares problem, hence the solution $x = (A, B, C, K)$ is given by

$$x = G \cdot w \quad (8.3)$$

for a suitable matrix G , where w the vector obtained by writing all columns of W below each other.

8.2 Single pitch deviations

The output parameters a , b , and c are given by

$$a_j = A_j - A_{j-1 \bmod N} \quad \text{for } j = 0, 1, \dots, N-1, \quad (8.4)$$

$$b_{q_1} = B_{q_1} - B_{q_r} \quad \text{and} \quad b_{q_r} = B_{q_r} - B_{q_{r-1}} \quad \text{for } r = 2, \dots, R, \quad (8.5)$$

and

$$c_j = C_j - C_{j-1 \bmod N} \quad \text{for } j = 0, 1, \dots, N-1. \quad (8.6)$$

8.3 Standard deviations

The variance σ^2 of the single measurements is given by

$$\sigma^2 = \frac{1}{(R-2)(N-1)} \sum_{j=0}^{N-1} \sum_{r=1}^R (A_j + B_{q_r} + C_{j-q_r \bmod N} + K - W_{j,q_r})^2. \quad (8.7)$$

As before, denote by w the vector obtained by writing all columns of W below each other. It is supposed that the variance-covariance matrix of w is given by

$$V_w = \sigma^2 E \quad (8.8)$$

with E the identity matrix in \mathbb{R}^{NR} . The variance-covariance matrices V_A and V_a of the vectors A and a can then be calculated according to the rules of error propagation: Since Equation (8.1) defines a linear least squares problem, there is a $N \times RN$ -matrix G_A such that

$$A = G_A \cdot w \quad (8.9)$$

(just take the first N rows of Equation (8.3)). It follows that

$$V_A = G_A V_w G_A^t = \sigma^2 G_A G_A^t. \quad (8.10)$$

The variance σ_A^2 is defined as the maximum value of the diagonal elements of V_A , i.e.

$$\sigma_A^2 = \max_{j=0, \dots, N-1} (V_A)_{jj}. \quad (8.11)$$

Since by Equation (8.4) the vector of the single pitch deviations a depends linearly on A , there is a matrix S such that $a = S \cdot A$. The variance-covariance matrix of a is thus given by $V_a = S V_A S^t$. One can show that all diagonal elements of V_a are identical, thus one can define σ_a^2 by

$$\sigma_a^2 = (V_a)_{00} = (V_a)_{11} = \dots = (V_a)_{N-1, N-1}. \quad (8.12)$$

9 Signature

$$[A, B, C, a, b, c, \sigma, \sigma_A, \sigma_a] = \text{VERROVER}(W, q)$$

10 References

- i.)* F. Keller, M. Stein, and K. Kniel. Reduced error separating method for pitch calibration on gears. In *Advanced Mathematical and Computational Tools in Metrology and Testing XI*, Series on Advances in Mathematics for Applied Sciences, vol. 89, pages 220–228. World Scientific Publishing Co Pte Ltd, 2018.

11 History

- i.*) Created on 2018-06-18 by Frank Keller (PTB, Germany)
- ii.*) Reviewed on 2018-06-20 by Martin Stein (PTB, Germany)
- iii.*) Updated on 2019-02-26 by Frank Keller (PTB, Germany)