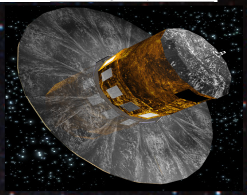
A new generic way to define astrometric calibration for Gaia data processing.

E. Joliet*, U. Lammers*

* European Space Astronomy Centre of ESA, Madrid, Spain



Abstract

Gaia is ESA's ambitious space astrometry mission with a foreseen launch date in early 2012. Its main objective is to perform a stellar census of the 1000 Million brightest objects in our galaxy (completeness to V=20 mag) from which an astrometric catalog of micro-arcsec level accuracy will be constructed. A key element in this endeavor is the Astrometric Global Iterative Solution (AGIS) --- the mathematical and numerical framework for combining the ~70 available observations per star obtained during Gaia's 5yr lifetime into a single global astrometric solution. The fundamental working principles of AGIS was shown (O4.1) at last year's ADASS XVIII. This time we present a new generic astrometric calibration scheme recently implemented in AGIS. For the development of the data processing software, the traditional astrometric calibration scheme is a heavy task as each new change in the model produces changes in the code, a need for new simulation data, new validations tests, etc. The new scheme allows the calibration of the astrometric instrument to be specified in a more generic and flexible manner. The entire model is defined with an external configuration file that can be modified at any time with no or only minimal impacts on the software. The implementation results in acceptable run time overheads compared to the direct approach with a fixed hard-coded calibration model. This new approach can be a starting point to convert other fixed and hard coded scheme into this more analytical breakdown solution.

The problem

Only the effects sufficiently understood and known today to be of relevance for the astrometric core solution have been studied in some details. The analytical forms have been suggested and implemented in AGIS:

- Geometric calibration of the focal plane
- A preliminary crude flux-dependent calibration as a zeroth order approximation to radiation-induced CTI effects
- A preliminary crude spectrum-dependent calibration as a zeroth order approximation to chromatic shifts

Radiation effect model may change even after launch and during operation phase for instance.

From the data processing software development perspective, this is particularly unsatisfactory as the new change in the model implies adapting the code, a need for new simulation data, new unit and validations tests, etc.

Calibration Model Algebraic Details

The core astrometric solution as implemented by AGIS optimizes all unknown, relevant quantities source, satellite attitude, and global parameters (s, a, g) such that the field angle (η,ζ) residuals are minimized in a least-square sense.

$$R_l^{\eta} = \eta_l^{obs} - \eta_l^{calc} = \eta_l^{obs} - F_{\eta}(t_l^{obs}, s, a, g)$$
$$R_l^{\zeta} = \zeta_l^{obs} - \zeta^{calc} = \zeta_l^{obs} - F_{\zeta}(t_l^{obs}, s, a, g)$$

Where Fη/ζ is the used observational model that predicts the field angle η/ζ at a given observation time for the current set of (s, a, g). The astrometric calibration enters this scheme through the observed field angles which are modeled as:

$$\eta_l^{obs} = \eta_n^0 + \Delta\eta_x + \delta\eta_y$$
$$\zeta_l^{obs} = \zeta_n^0 + \Delta\zeta_x$$

Here η/ζ are the nominal AL/AC position of CCD n are the nominal AL/AC position of CCD n with which observation l was made. The astrometric calibration parameters Δη (AL large-scale), δη (AL small-scale) and Δζ (AC large-scale) are corrections to the nominal values with each parameter value representing one distinct calibration effect. The symbols x and y stand for a combination of indices that depend on observation index *l*.

Past dataset processing tests shown correlation problems calculating the geometric calibration parameters in the absence of any non-geometric calibration effects (flux- or spectrum dependent effects). Outlined because of the size of the normal matrix, a rigorous direct determination of the calibration parameters using least-squares and normal equations may be revisited. Normal equations can be set up and solved separately for each CCD/gate combination which reduces the total matrix size by a large factor and is ideal for distributed processing.

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<AstroCal xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="astrocal.xsd">
 <DataSpace>
    <Axis id="0" label="CcdRowNo" origin="0" min="1" max="7" delta="1" discrete="true" distributed="true"/>
    <Axis id="1" label="AFCcdStripNo" origin="0" min="3" max="11" delta="1" discrete="true" distributed="true"/>
    <Axis id="2" label="Time" origin="2012-01-01T00:00:00 (TCB)" min="0d" max="1840d" delta="115d"
discrete="false" distributed="false">
       <Discretization id="1" uniform="true" scale="4"/>
       <Discretization id="2" uniform="true" scale="2"/>
    </Axis>
    <Axis id="3" label="Telescope" origin="0" min="1" max="2" delta="1" discrete="true" distributed="false"/>
    <Axis id="4" label="PixelCol" origin="0" min="0" max="11" delta="1" discrete="true" distributed="false">
       <Discretization id="1" uniform="true" scale="2"/>
       <Discretization id="2" uniform="true" scale="3"/>
    </Axis>
 </DataSpace>
 <FuncsCollection>
    <Func id="0" classcal="gaia.cu3.agis.algo.gis.calibration.model.L0" description="1"/>
    <Func id="1" classcal="gaia.cu3.agis.algo.gis.calibration.model.L1" description=" 2(mubar-1/2)"/>
    <Func id="2" classcal="gaia.cu3.agis.algo.gis.calibration.model.L2" description="6(mubar-1/2)^2-1/2"/>
    <Func id="3" classcal="gaia.cu3.agis.algo.gis.calibration.model.G0" description="(G-G0)"/>
    <Func id="4" classcal="gaia.cu3.agis.algo.gis.calibration.model.Nu0" description="(nu-nu0)"/>
 </FuncsCollection>
 <ALEffectsCollection>
    <Effect id="0" use="true" update="true" description="AL large-scale" dep="*,*,0,*,-" funcs="0,1,2"
constraints="gaia.cu3.agis.algo.gis.calibration.model.AlLargeScaleConstraints"/>
    <Effect id="1" use="true" update="true" description="AL small-scale" dep="*,*,-,-,0" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
    <Effect id="2" use="true" update="true" description="Radiation damage" dep="*,*,1,-,1" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
    <Effect id="3" use="true" update="true" description="Chromaticity" dep="*,*,2,*,-" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
    <Effect id="4" use="true" update="true" description="Effect X" dep="*,*,-,*,2" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
    <Effect id="5" use="true" update="true" description="Effect Y" dep="*,*,2,*,2" funcs="0"
constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
 </ALEffectsCollection>
 <ACEffectsCollection>
    <Effect id="0" use="true" update="true" description="AC large-scale" dep="*,-,-,*,-"
 funcs="0" constraints="gaia.cu3.agis.algo.gis.calibration.model.NoConstraints"/>
 </ACEffectsCollection>
</AstroCal>
```

The generic solution

The basic idea is to formulate the calibration in terms of general "analytical calibration functions" whose specific form and number are treated as variable in the coding phase.

Compared to the past approach with a fixed hard-coded calibration model, the new generic and flexible manner considers the entire model to be defined into an (XML) external configuration file that can be modified at any time with no or only minimal impacts on the software.

We define a direct generalization in terms of calibration effects and functions:

$$\text{AL}: \quad \eta_l^{obs} = \eta_n^0 + \sum_{i=0}^{N_{AL}-1} E_i^{AL}(l)$$
$$\text{AC}: \quad \zeta_l^{obs} = \zeta_n^0 + \sum_{i=0}^{N_{AC}-1} E_i^{AC}(l)$$

Each of the *E* represents one basic *calibration effect* and is composed of a linear combination of *calibration functions f and calibration units* $c_{i,j}$:

$$E_i(l) = \sum_{j=0}^{K_i-1} c_{i,j} \cdot f_{i,j}(l) \quad ; \quad c_{i,j} = c_{i,j}(\kappa_0, \ldots \kappa_{L_{i,j}-1})$$

The functions f receive the observation index l and it is assumed that this index suffices to compute from it all parameters needed to evaluate the corresponding function/effect for this observation. Examples of these parameters are:

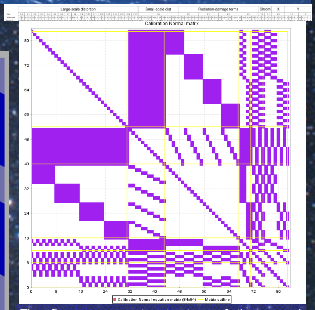
- telescope index (K_3)
- CCD row number (K_0)
- CCD strip number (K_1)
- pixel column (K_4)
- time (K_2), etc (K_j).

The Singular Value Decomposition is used to solve an *M ×M* system of linear equations for the unknowns *c(i,j) that minimize field angle residuals.*

All calibration units c (together with the corresponding functions) constitute the calibration of the astrometric instrument.

- <AstroCal>: Root element and schema validation
- <DataSpace>: axis space definition
  - <Axis>: continuous / discrete space axis with min, max and delta values
    - <Discretization>: define the discretization regarding the scale factor between the base axis. Non-*uniformed* discretization not considered (yet).
- <FuncsCollection>: group of functions (ids) used to define calibration effects
  - <ALEffectCollection>: AL calibration effects
  - <ACEffectCollection>: AC calibration effects
- <Effect>: define the effect based on a function "id" and a constraint class instantiated on runtime.

The effect dependencies ("dep") on the space axis are represented with a list of characters in a given order such as it follows the axis definition - the order is the same as the list of *<Axis>*: "0" means the axis discretization id=0, "*" means that the effect depends on this axis, "-" means no dependency.



The figure represents structure of the normal matrix for an assumed realistic set of calibration effects and parameters discretizations per CCD.

Datacube implementation

The list of parameters K_k, $0 < k < L_{i,j}$ that the calibration functions may depend on can be regarded as forming a general data space of dimensionality $L_{i,j}$. Each parameter represents an axis of this space with the following assumptions:

- Each of the axes is discrete and finite in both directions. Example: CCD number [1,2,3,4,5,6,7].
- For intrinsically real-valued parameters a meaningful discretization is assumed to exist, i.e. the time. A corresponding calibration effect to first discretization (covering 5 year of nominal operation time) is time-independent and this dimension of the parameter data space can effectively be collapsed.

In this approach, changing the calibration model, i.e. changing existing calibration functions and/or include new effects with maximum ease requires just editing the corresponding entries in an XML file prior to running the software.

Each calibration data AL and AC are represented in a one-dimensional vector. A class provides the mapping between a datacube (one per calibration effect) and the calibration unit in the 1D data vector.

Although untested with large-scale tests, the complete framework is fully implemented. The Java implementation was done using the Java XML API (JAXB) standard in JDK 1.6 to bind validate and load the calibration model definition xml file based on a schema (XSD format file) that reflects the datacube implementation.

For more information on Gaia, visit http://www.rssd.esa.int/Gaia