POINTING ERROR ENGINEERING FRAMEWORK FOR HIGH POINTING ACCURACY MISSIONS

Massimo Casasco(1), Sohrab Salehi(2), Sven Weikert(3), Jochen Eggert(3), Marc Hirth(3), Haifeng Su(4), and Thomas Ott (5)

(1)*ESA - ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands*

(2)*Rhea System for ESA, ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands, +31(0)715655126, Sohrab.Salehi@esa.int*

(3)*Astos Solutions GmbH, Meitnerstrasse 8, 70569 Stuttgart, Germany*

(4)*Institute of Flight Mechanics and Control, University of Stuttgart, Pfaffenwaldring 7a, 70569 Stuttgart, Germany*

(5)*Airbus Defence and Space, D-88039 Friedrichshafen, Germany*

Abstract: *The publication of the ECSS Control Performance Standard and the ESA Pointing Error Engineering Handbook forms part of the recent standardization effort in Europe to define a clear pointing error engineering methodology for ESA projects. This is complemented by the development of the Pointing Error Engineering Tool, a prototype software, released under the ESA Software Community License and intended to support dissemination of this pointing error engineering methodology. This paper describes the mathematical framework and the steps of the methodology for pointing error budgeting that is implemented in the Pointing Error Engineering Tool and highlights the benefits that this tool can provide with respect to traditional conservative approaches, particularly for high pointing accuracy missions. Finally, the perspective activities to be promoted by ESA in the area of tools for pointing error engineering are outlined.*

Keywords: *Error budgeting, software tool, high accuracy pointing*

1. Introduction

Pointing error engineering covers the engineering process of establishing system pointing error requirements, their systematic analysis throughout the design process and eventually the verification of compliance. For technical as well as historical reasons, pointing error engineering in the European space community has been implemented on the basis of engineering practices that were often tailored on a case-by-case basis and no standard practice was in place. This has changed through the initiative of the European Cooperation for Space Standardization (ECSS). The ECSS Control Performance Standard [1] now provides a solid and exact mathematical basis for constructing performance error budgets. The Pointing Error Engineering Handbook (PEEH) is an ESA applicable document [2], containing a step-by-step process with guidelines, summation rules, recommendations and examples. It provides ESA projects with a clear pointing error engineering methodology.

ESA PEEH users are supported by the Pointing Error Engineering Tool (PEET). System engineers and control engineers use it for the pointing requirements allocation activities, during the early phases of a project, and for the pointing error budget verification activities, in later phases. ESA initiated and coordinated the development of a PEET prototype, carried out by Astos Solutions GmbH with the support of the Institute of Flight Mechanics and Control of the University of Stuttgart. This approach is intrinsically capable of minimizing the margins and uncertainties in pointing budgets and is expected to prove extremely valuable especially for high pointing accuracy missions, where accurate analytical results obtained using PEET could make the difference in taking the correct design decisions.

PEET has since been extended for relative position error budgeting and is released under the ESA Software Community License. Operational software, targeting use in Phases B and C, is under development for ESA to be released under the same license type as PEET. This will permit processing complex calculations for high accuracy pointing, frequency domain techniques introduced in the handbook, provision of traceability and a common platform for exchange of information between the various entities. ESA is using PEET for the pointing error engineering analyses of a number of missions, including Euclid [6], MetOp-SG [7], and Proba-3 [8].

The paper provides a detailed summary of the pointing error analysis and evaluation method according to the PEEH. The implementation and features of the PEET prototype are described.

2. Pointing Error Analysis and Evaluation Methodology

While the ECSS Control Performance Standard E-ST-60-10C provides normative clauses with clear mathematical elements for control performance analysis in general, which apply to all disciplines involving control engineering and at different levels, ranging from equipment to system level, the ESA PEEH embeds the elements of the ECSS standard in a step-by-step engineering process for the specific case of satellite pointing errors. The four analysis steps for pointing error analysis and evaluation are described in Sections 2.3-2.6. More details on the theoretical background of the ESA PEEH are available in [4] and [5].

2.1. Nomenclature and Definitions

A pointing error e can be considered as the response of a system to external or internal physical phenomena affecting the system's pointing performance as illustrated in Figure 1.

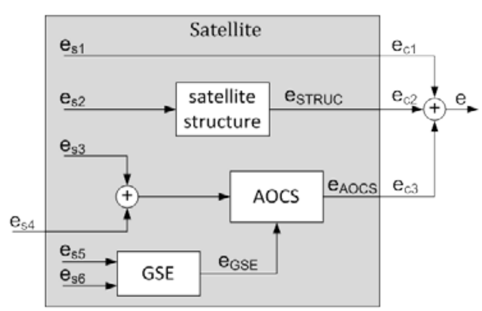


Figure 1. Pointing error source transfer

Each such physical phenomenon is referred to as a pointing error source (PES) and denoted by e_s. A PES is categorized as being either constant in time (time-constant), random in time (time-random) and/or random in its realization (ensemble-random).

A pointing error contributor (PEC), denoted by e_c, represents the contribution of one or more pointing error sources e_s to the overall pointing error e. A PES becomes a PEC through undergoing a pointing system transfer, such as a coordinate frame transformations, control system, or structural transfer function. In order to analyze pointing performance, a pointing system is broken down into subsystems with individually controlled (active or passive) transfer properties. The pointing error e is the sum of the different PECs.

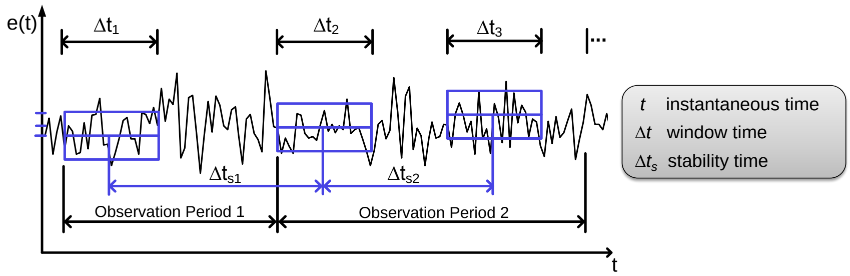


Figure 2. Time dependency of pointing errors

Several types of time dependencies of pointing errors can be distinguished. These are graphically illustrated in Figure 2 for the observation requirements of a pointing system, e.g. a satellite and its payload [3]. Pointing errors can depend on:

- Instantaneous time t: pointing error at any point in time t during system lifetime or a defined observation period.
- Window time Δt: pointing error within a time window Δt, where the time window can occur at any point in time t during system lifetime or a defined observation period.
- Stability time Δt_s: pointing error describing stability, thus the relative error, among pointing errors in time-windows of length Δt. The time-windows are separated by a time difference of length Δt_s, and can occur at any point in time t during system lifetime or a defined observation period.

The time-dependent pointing errors are summarized in Table 1.

Table 1. Definition of pointing error indices

Index	Name	Definition
AKE	Absolute Knowledge Error	difference between the actual parameter (attitude, geolocation, etc.) and the known (measured or estimated) parameter in a specified reference frame
APE	Absolute Performance Error	difference between the target (commanded) parameter (attitude, geolocation, etc.) and the actual parameter in a specified reference frame
MKE	Mean Knowledge Error	mean value of the AKE over a specified time interval Δt
MPE	Mean Performance Error	mean value of the APE over a specified time interval Δt
RKE	Relative Knowledge Error	difference between the AKE at a given time within the time interval Δt and the MKE over the same time interval
RPE	Relative Performance Error	difference between the APE at a given time within the time interval Δt and the MPE over the same time interval
KDE	Knowledge Drift Error	difference between MKEs taken over two time intervals separated by a specified time Δt_s within a single observation period
PDE	Performance Drift Error	difference between MPEs taken over two time intervals separated by a specified time Δt_s within a single observation period
KRE	Knowledge Reproducibility Error	difference between MKEs taken over two time intervals separated by a specified time Δt_s within different observation periods
PRE	Performance Reproducibility Error	difference between MPEs taken over two time intervals separated by a specified time Δt_s within different observation periods.

The comprehensive set of pointing error indices, categorized as knowledge or performance errors and depending on instantaneous time, window time and stability time, is formulated in Table 2.

Table 2. Mathematical formulation of pointing error indices

Pointing Error Indices	
index	***instantaneous***
$e_{APE}(t)$	$= e_P(t)$
$e_{AKE}(t)$	$= e_K(t)$
$e_{MPE}(t, \Delta t)$	$= \overline{e_P}(t, \Delta t)$
$e_{MKE}(t, \Delta t)$	$= \overline{e_K}(t, \Delta t)$
$e_{RPE}(t, \Delta t)$	$= e_P(t) - \overline{e_P}(t, \Delta t)$
$e_{RKE}(t, \Delta t)$	$= e_K(t) - \overline{e_K}(t, \Delta t)$
$e_{PDE}(t, \Delta t_1, \Delta t_2, \Delta t_S)$ $e_{PRE}(t, \Delta t_1, \Delta t_2, \Delta t_S)$	$= \overline{e_P}(t, \Delta t_1) - \overline{e_P}(t + \Delta t_s, \Delta t_2)$
$e_{KDE}(t, \Delta t_1, \Delta t_2, \Delta t_S)$ $e_{KRE}(t, \Delta t_1, \Delta t_2, \Delta t_S)$	$= \overline{e_K}(t, \Delta t_1) - \overline{e_K}(t + \Delta t_s, \Delta t_2)$

Δt	*window time*	e_{index}	*instantaneous error*
Δt_s	*stability time*	$e_K(t)$	*knowledge error signal*
		$e_P(t)$	*performance error signal*

time average: $\bar{e}(t, \Delta t) = \langle e(t) \rangle_{\Delta t} = \frac{1}{\Delta t} \int_{t-\Delta t/2}^{t+\Delta t/2} e(t)\ dt$

instantaneous time

window time

stability time

The frequency domain classification of time windowed and windowed stability errors allows an exact evaluation of the metrics defined Table 2. They can be used to determine the contribution of the PEC signal PSD to the different time windowed and windowed stability errors. For this the PES PSD needs to be characterized or at least approximated with reasonable assumptions.

The frequency-domain pointing error metrics are specific PSD weighting functions F_{metric}. In order to perform analysis, rational approximations, $\tilde{F}_{metric}$, of the weighting functions are given in [5] and summarized in Table 3 such that $F_{metric}(\omega) \sim |\tilde{F}_{metric}(j\omega)|^2$ and with $s=j\omega$. The metrics can be understood as a function by which the PEC signal power, described by its PSD, is weighted. The weighting function corresponds to a low pass, a high pass or a combination of both. As can be seen in Table 3 the weighting functions have the form of a sinc-function. This is due to the fact that the windowing in the time domain is equivalent to filtering the time signal by a rectangular function, which has the sinc-function as frequency domain equivalent.

Table 3. Pointing error metrics - frequency domain

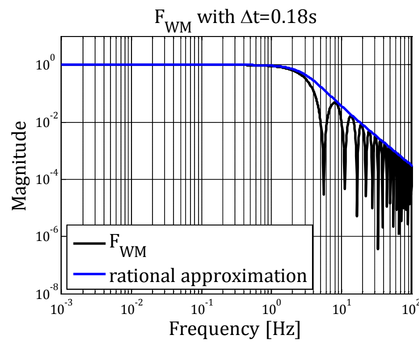
Pointing Error Metric Weighting Functions F_{metric}

Windowed Mean (WM)

$$F_{WM}(\omega,\Delta t)=\frac{2(1-\cos(\omega\Delta t))}{(\omega\Delta t)^2}$$

rational approximation:

$$\tilde{F}_{WM}(s,\Delta t)=\frac{2(s\Delta t+6)}{(s\Delta t)^2+6(s\Delta t)+12}$$

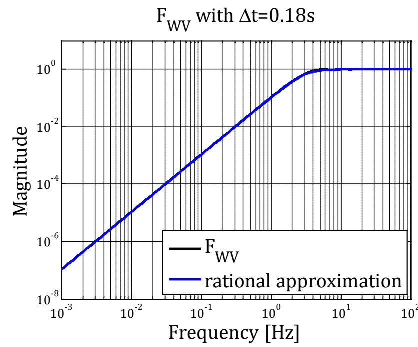


Windowed Variance (WV)

$$F_{WV}(\omega,\Delta t)=1-\frac{2(1-\cos(\omega\Delta t))}{(\omega\Delta t)^2}$$

rational approximation:

$$\tilde{F}_{WV}(s,\Delta t)=\frac{s\Delta t\left(s\Delta t+\sqrt{12}\right)}{(s\Delta t)^2+6(s\Delta t)+12}$$

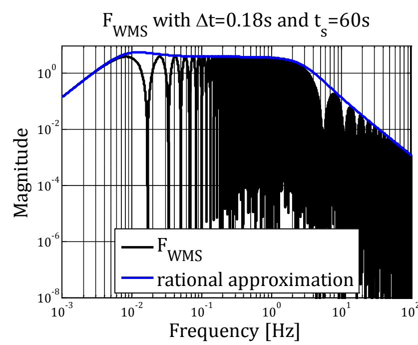


Windowed Mean Stability (WMS)

$$F_{WMS}(\omega,\Delta t,\Delta t_s)=F_{STA}\times F_{WME}=\frac{4(1-\cos(\omega\Delta t_s))^2}{(\omega\Delta t)^2}$$

rational approximation:

$$\tilde{F}_{WMS}(\omega,\Delta t,\Delta t_s)=\tilde{F}_{STA}\times\tilde{F}_{WM}$$

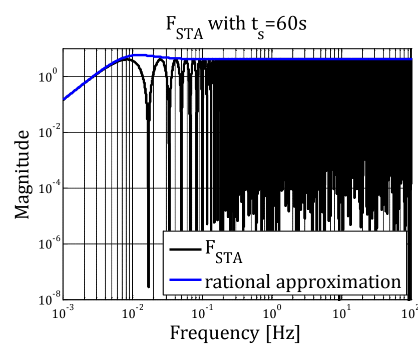


Stability (STA)

$$F_{STA}(\omega,\Delta t_s)=2(1-\cos(\omega\Delta t_s))$$

rational approximation:

$$\tilde{F}_{STA}(s,\Delta t_s)=\frac{2s\Delta t_s(s\Delta t_s+6)}{(s\Delta t_s)^2+6(s\Delta t_s)+12}$$



2.2. Pointing Error Engineering Methodology Framework

Figure 3 shows a flow diagram for the pointing error engineering methodology. The awareness of the whole pointing error engineering cycle is key for pointing error engineering activities from requirement specification to performance verification: indeed for specification of pointing error requirements relevant analysis and verification methods have to be identified and vice versa.

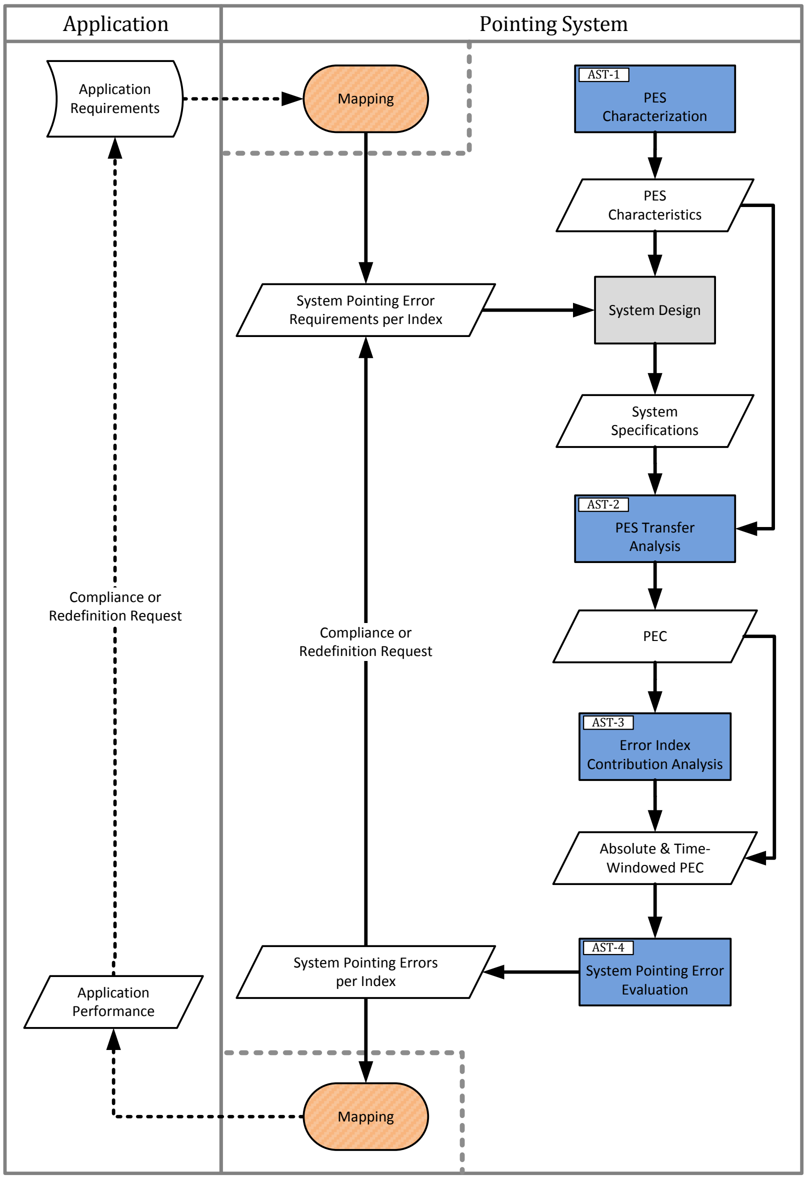


Figure 3. Pointing error engineering methodology workflow

The process starts with mapping the user specified Application Requirements into unambiguous System Pointing Error Requirements, formulated according to the classification in Table 2. The compliance of the system pointing error requirements is analyzed by estimating and combining the different occurring error sources in four analysis steps, AST-1 to AST-4. Note that the mapping process is not treated in the PEEH because it is application specific.

The analysis steps AST-1 to AST-4 from Figure 3 should then be applied to each subsystem of the pointing system. For complex cases, the pointing system can be broken down into several subsystems. Then step AST-4 is performed again at pointing system level in order to compile and evaluate the overall pointing error budget.

This methodology can be implemented via two different main analysis methods:

a) simplified statistical method: analysis with standard deviation, σ, and mean, μ, and their summation per ECSS pointing error index under the assumption of the central limit theorem.
b) advanced statistical method: analysis by joint PDF characterization via convolution of different error probability density functions (PDF), *p…(e)*, and evaluation of level of confidence for required ECSS pointing error indices.

The currently available PEET prototype implements the simplified statistical method. It is however foreseen that future releases of PEET will also include the implementation of the advanced statistical method.

2.3. Step AST-1: Characterization of Pointing Error Source

Selecting the eligible mathematical elements for PES characterization are selected in step AST-1, depends on the PES error data characteristics as well as the type of available PES error data. Figure 4 shows the categorization of a PES as time-constant or time-random, and its description as random variable or random process.

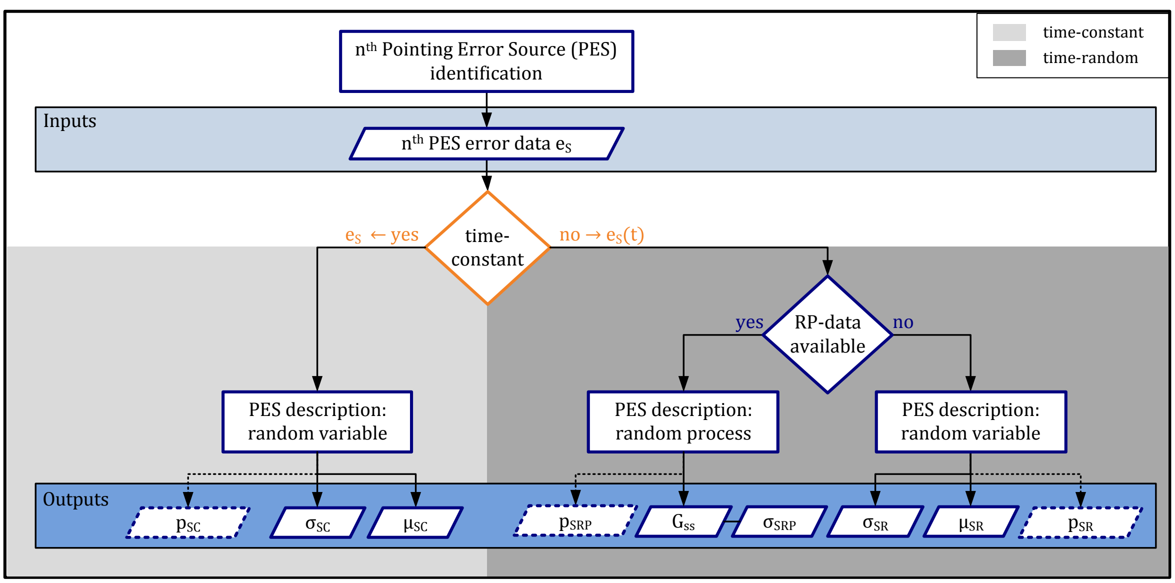


Figure 4. Characterization method

Depending on the maturity of the available data, a PES can be described according to its fundamental properties:

- ensemble-randomness
- time-randomness
- power spectrum

In the decision tree, a PES is categorized depending on its fundamental properties. The first decision is whether a PES is time-constant or time-random. Time-constant PES do not vary randomly with time, but in their ensemble of realizations. On the other hand, time-random PES have a magnitude that varies randomly in time. Of course a combination of both is also possible, meaning that a PES can have time-random and time-constant properties (e.g. a periodic signal whose amplitude varies over different realizations). A time-constant PES is described as a random variable in line with the mathematical elements provided in the PEEH. A time-random PES is ideally described as a stationary random process if sufficient information about the underlying process is available. This is because describing a PES as stationary random process has the advantage that exact window time and stability time properties of the PES are captured.

A random pointing error process $\{e_k(t)\}$ is an ensemble of k sampling function realizations that are random in time t (time-random) and random in its ensemble of realizations (ensemble-random). The ensemble is the set {…} of all realizations k of the random pointing error $e_k(t)$. The probability properties of a random process are described by the ensemble statistical quantities (e.g. mean or variance) at fixed values of t, where $e_k(t)$ is a random variable over the index k. In general, the statistical quantities are different at different times t. If the statistical quantities are equal for all t the random process is said to be stationary. A stationary random process is described by its PDF $p(e)$. In practice most stationary random processes have a Gaussian PDF and thus are completely defined by their mean value and covariance respectively [2].

The frequency domain characteristics of a random stationary process are described by means of its power spectral density (PSD). This becomes important when considering time-windowed pointing errors because windowing in time domain is equivalent to low-pass-filtering in frequency domain. This enables mathematically exact analysis of time dependent pointing errors. The PSD is a powerful formalism to describe random stationary noise processes. The double-sided PSD of $e_k(t)$ in [unit^2/(rad s^{-1})] is defined as $S_{ee}(\omega)$, based on which the single-sided PSD is given as $G_{ee}(\omega)=2S_{ee}(\omega)$, with ω being the frequency in [rad s^{-1}]. The single-sided PSD is commonly also defined in [unit/√(rad s^{-1})], in which case it is referred to as $P_{ee}(\omega)= \sqrt{G_{ee}(\omega)}$.

If time series data is not available, [1] provides guidelines for an approximate random variable description. Examples of application of the decision tree for the random process and random variable cases are provided in [3].

2.4. Step AST-2: Transfer Analysis

The description of the PES is given with respect to its point of origin. In order to evaluate a pointing error requirement, the transfer of a PES from its origin to the point of interest needs to be analyzed in step AST-2 to determine the pointing error contributor (PEC). This can be done by decomposing the pointing system into subsystems, as exemplified in Figure 1. These PECs are obtained by a transformation, which depends on the system under evaluation. The transfer characteristics of each system are tunable to a certain extent and thus can be used to perform trade-offs with the aim of making pointing errors compliant with their requirement. The input (PES) and output (PEC) parameters of the transfer analysis are shown in Figure 5.

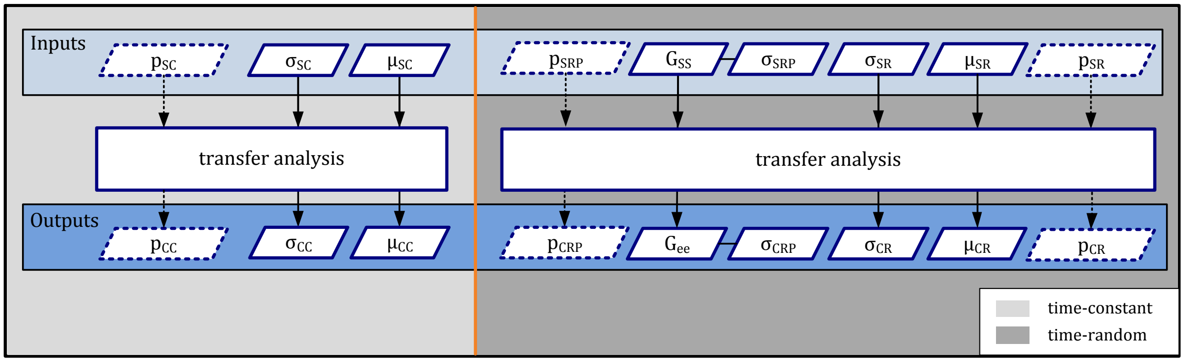


Figure 5. Transfer analysis

There are various techniques in the frequency domain and the time domain for the system transformation of time-random PES described as random processes. The analytic methods are based on linear transformation of statistical properties, whereas the numerical methods rely on simulations and experimental results. In the case of a time-constant PES, the system transformation analysis is simply the multiplication of the bias/mean value with the system steady-state gain.

The frequency domain approach relies on the observation that if the input error signal of a system, the PES, is known and the system can be represented by a linear time-invariant (LTI) transfer function $H(j\omega)$, being stable and strictly proper, the output error signal, the PEC, can be determined. The variance of a PES described as random processes is related to its PSD as shown in Eq. 1.

$$\sigma_{SRP}^2(e) = \frac{1}{2\pi}\int_0^\infty G_{ss}(\omega)d\omega \tag{1}$$

The PSD G_{ss} of the input error signal $e_s(t)$ is transformed by the system according to the well-known relations in Eq. 2 (SISO case) and Eq. 3 (MIMO case).

$$G_{ee}(\omega) = |H(j\omega)|^2 G_{ss}(\omega) \tag{2}$$

$$\mathbf{G}_{ee}(\omega) = \mathbf{H}^*(j\omega)\mathbf{G}_{ss}(\omega)\mathbf{H}(j\omega) \tag{3}$$

The variance of the output error signal $e_c(t)$ is thus computed from its PSD G_{ee} via Eq. 4.

$$\sigma_{CRP}^2(e) = \frac{1}{2\pi}\int_0^\infty G_{ee}(\omega)d\omega \tag{4}$$

This transfer can be analyzed by various methods [9][10]. The advantage of this analytical approach, fully implemented in the PEET prototype, over numerical methods is that it can be used to tune the system transfer function H based on signal and system norms. Guidelines for transfer analysis based on simulations and experimental results are provided in [1], but these approaches are not implemented in the PEET prototype.

2.5. Step AST-3: Pointing Error Index Contribution

The contribution of PECs to the pointing error indices is determined in step AST-3. This step can be skipped for random time-constant PES because it does not depend on time. In addition, AST-3 can also be skipped for the analysis of errors, such as APE and AKE, which only depend on the instantaneous time, and not on the window time or stability time. The contribution analysis is shown in Figure 6 with input and output parameters.

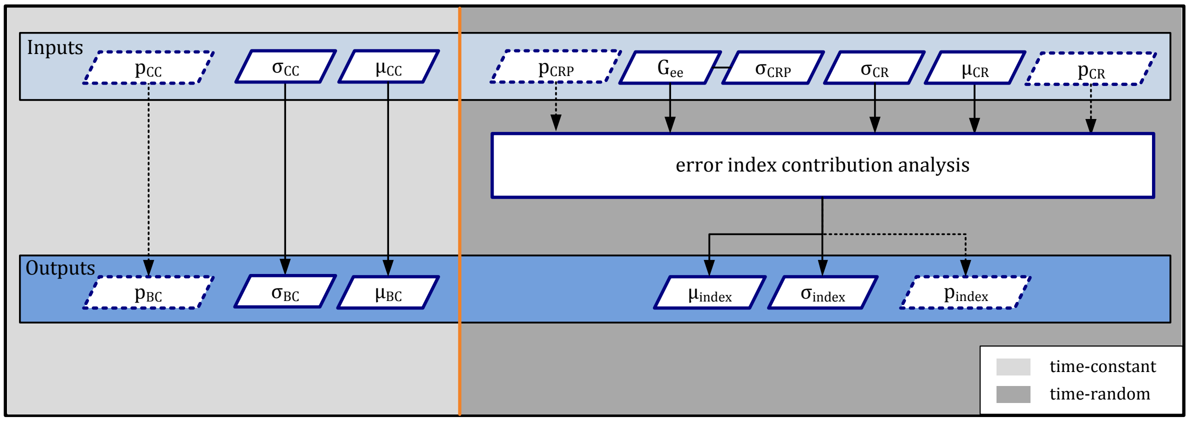


Figure 6. Pointing error index contribution analysis

Guidelines are provided in [1], for evaluating the pointing error index contribution of a random variable description of time-random PECs. These take the form of tables that quantify the contribution for a number of different error probability distribution functions. These tables are used in the PEET prototype when the random variable description of a PEC is selected.

On the contrary, for PECs described as random processes, an exact evaluation of the contribution to an error index is possible by evaluating the integral of the PSD associated with the stationary random-process and applying a suitable spectral weighting function [2][3]. A summary of the exact expressions and rational approximations for the weighting functions can be found in [2]. The PEET prototype includes algorithms that implement and perform all the calculations needed to evaluate the error index contributions for PECs described as stationary random processes.

A worked example, that illustrates the application of the process in the frequency domain makes use of the weighting functions described above, can be found in [3].

A PEC needs to be interpreted with respect to the required statistical property in line with statistical interpretation guidelines provided in the PEEH. If a PES is described as random process, the statistics is interpreted for each pointing error index contributor at the end of AST-3. On the other hand, if a PES is described by a random variable, an equivalent mean and variance is determined based on the statistical interpretation already in AST-1. In the following a short summary is given on the statistical interpretation.

The properties of physical phenomena, and thus the pointing errors and their sources, are described in terms of their probability characteristics. Three statistical interpretations [1] for describing the property and statistical characteristics are defined: mixed, ensemble, and temporal

In the mixed interpretation one considers the probability P greater or equal to a level of confidence P_c such that the ensemble of pointing error realizations $\{e_k(t)\}$ or $e(k,t)$ is less than a required error value e_r in its ensemble of realizations k and in time t. This mathematically translates into Eq. 5.

$$Prob[\{|e_k(t)|\} < e_r] \geq P_c \quad or \quad Prob[|e(k,t)| < e_r] \geq P_c \tag{5}$$

In the ensemble interpretation the probability P greater or equal to a level of confidence P_c is considered, such that a realization k of the ensemble of pointing error realizations $\{e_k(t)\}$ or $e(k,t)$ is less than a required error value e_r for all times t. This mathematically translates into Eq. 6.

$$Prob[|e_{max}(k)| < e_r] \geq P_c \;\; with\; e_{max}(k) = \max_t[\{e_k(t)\}] \; or \; \max_t[e(k,t)] \tag{6}$$

In the temporal interpretation the probability P greater or equal to a level of confidence P_c is considered, such that the entire ensemble of pointing error realizations $\{e_k(t)\}$ or $e(k,t)$, or just the worst case realization, with realization index k, is less than a required error value e_r for a fraction of time t. This mathematically translates into Eq. 7.

$$Prob[|e_{max}(t)| < e_r] \geq P_c \;\; with\; e_{max}(t) = \max_k[\{e_k(t)\}] \; or \; \max_k[e(k,t)] \tag{7}$$

2.6. Step AST-4: Pointing Error Evaluation

Pointing error is evaluated in two steps, as shown in Figure 7.

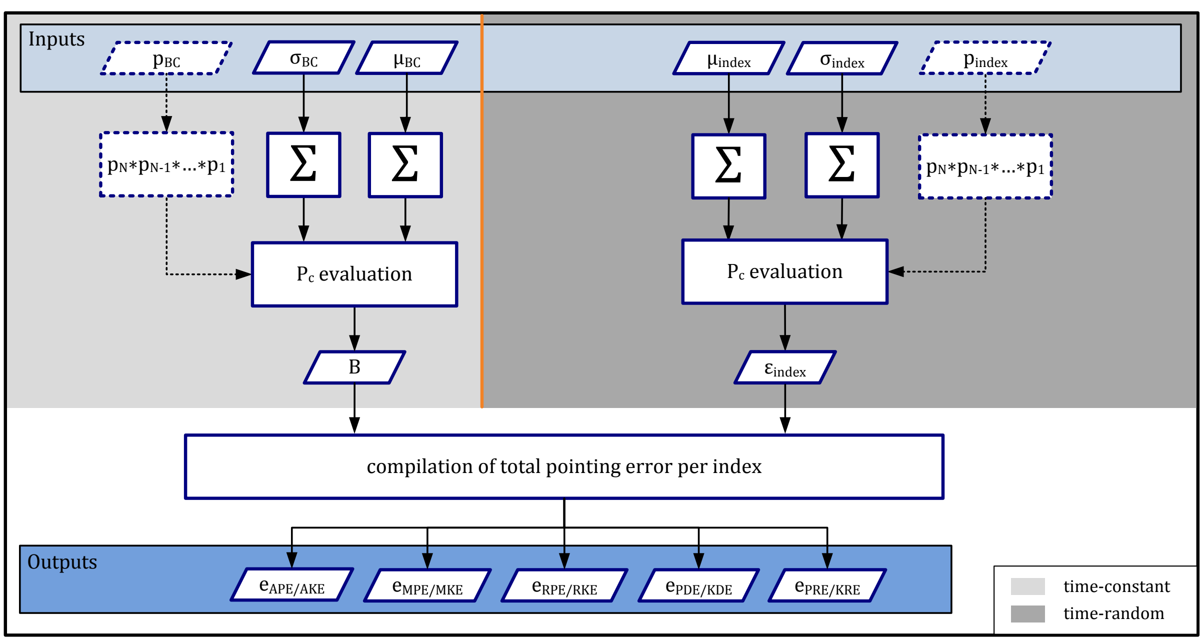


Figure 7. Pointing error evaluation

The time-constant and time-random error contributors are first combined together separately, taking into account possible correlations between errors. The probability with the applicable confidence level is then computed. The total pointing error is subsequently computed per error index from both intermediate results.

The rules for summing means and variances of error contributors are the following: the means are summed linearly, the uncorrelated variances are RSSd, and upper bound estimation is used for the correlated variances.

The error index is then computed for the applicable confidence level. First the standard deviation is multiplied by *np*, where *np* is a positive scalar such that for a Gaussian distribution the *np* confidence level encloses the probability P_c, as specified in the requirement. Then the scaled standard deviation is summed with the mean value. Finally, the time-constant and time-random pointing errors are summed per APE, AKE, MPE and MKE indices. Note that the time-constant error does not contribute to the other pointing error indices.

3. The Pointing Error Engineering Tool (PEET)

The Pointing Error Engineering Tool (PEET) automates the 4-step pointing error engineering methodology presented in the ESA Pointing Error Engineering Handbook (PEEH). The current prototype has a special focus on pre-phase A and phase A activities. PEET is released under the ESA Software Community License.

The architectural structure of PEET is shown in Figure 8.

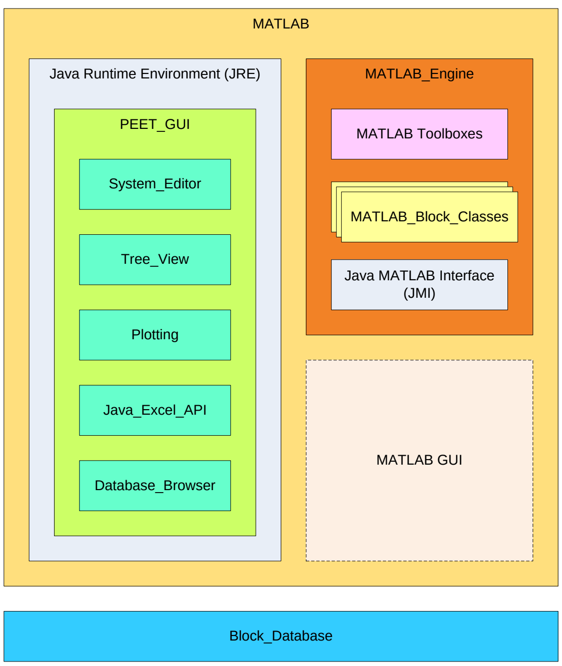


Figure 8. Architectural structure

The graphical user interface is written in Java Swing with a core implemented by MATLAB classes. The Java GUI runs completely within the virtual machine of the MATLAB installation and does not need any additional Java runtime. It communicates with the MATLAB core through MATLAB's *Java Matlab Interface*. With the exception of the Control System Toolbox, PEET relies on a standard MATLAB installation. An additional interface is available for import/export of data with Excel. Data from the MATLAB workspace can also be accessed by PEET.

3.1.Graphical User Interface

The user interface is very similar to Simulink. To build up pointing systems, the user can add various building blocks and connections between them, using predefined building blocks applicable to pointing error engineering, provided in the PEET database.

The user first constructs the pointing system by dragging building blocks, such as pointing error sources (PES) or transfer systems, into the System Editor (see Figure 9). Each type of block has a mask through which the parameters of the block can be configured. The block is then connected to other blocks.

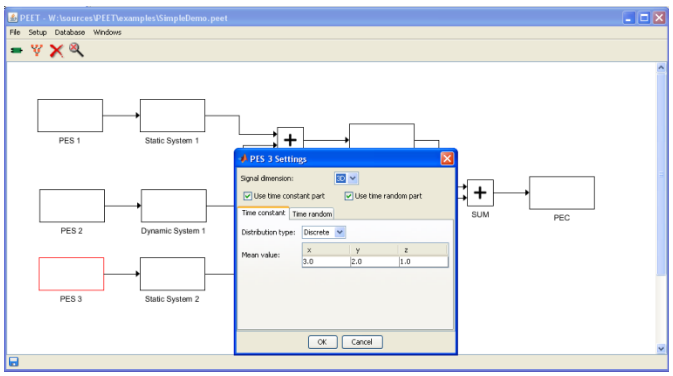


Figure 9. System editor and block mask

The user then defines the error indices applicable to the pointing scenario, as well as any correlation between error sources or correlation between axes of a single error source. PEET currently supports uncorrelated and fully correlated error sources. The user then runs PEET and can inspect the results of the error computations in the Tree View (see Figure 10).

The pointing system is shown in a tree-like structure, on the left side of the screen. Selecting a particular block causes the error information of its input and output signals to show up on a number of tabs, on the right side of the screen. The error signals are split up into components (e.g. random variable part, drift part, etc.) to give a better overview about the error contribution at this point of the pointing system. The final block shown at the bottom of the tree has a special tab called *Pointing error*. On this tab, all the information about the final pointing error and the line-of-sight error is presented to the user.

MATLAB® is a registered trademark of MathWorks.
Excel® is a registered trademark of Microsoft.

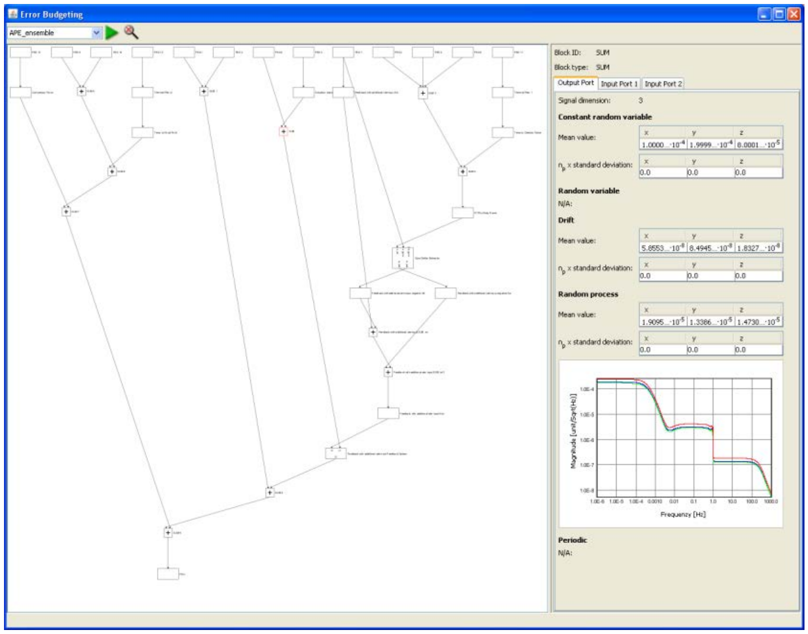


Figure 10. Tree View

Additionally PEET has a sensitivity analysis capability, which can be used for the identification of error drivers in the pointing system. In general, the sensitivity value expresses the change of the components of the final pointing error with respect to a change in some scalar parameter value. Figure 11 shows an example of the sensitivity analysis. In this case, changing the value of the parameter’s upper left matrix element will have a large influence on the y-axis of the final pointing error.

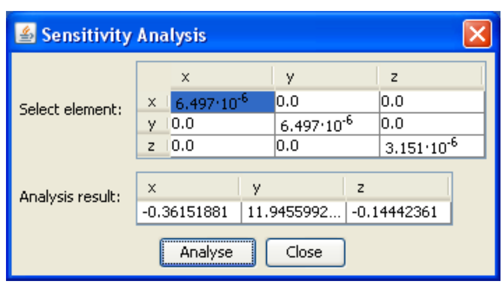


Figure 11. Sensitivity analysis manager

3.2. PEET Core

PEET supports a wide range of pointing error sources which can be either 1-dimensional or 3-dimensional. Depending on the available data, each PES can be described by its statistical properties or by a PSD (see Section 2.3). The time-correlation of the noise, i.e. its coloring, can be fully taken into account by using the PSD representation. PEET automatically converts PES signal representations into an eligible format using a structure with 5 signal types (time-constant random variable, time-random random variable, PSD, drift, and periodic). Each PES signal type is internally mapped either to a PSD or to a covariance matrix, as the underlying treatment in the system transfer is different for each signal type.

PEET provides tools and models to support the user in defining the system transfer from PES to contributors. PEET provides parameterized blocks for the most common models. The following models [11] are currently supported by PEET:

- Pointing Error Sources
- Mapping from 1D signals to 3D signals
- Coordinate Transformations
- Flexible and Rigid Plant Models
- Static Systems (e.g. matrix)
- Dynamics Systems (e.g. transfer function, state-space, zero-pole-gain, etc.)
- Feedback Systems
- PID Controllers
- Gyro-Rate Noise (parametric model based on [12])
- Gyro-Stellar Estimators
- Star Tracker Noise (parametric model for sensor field-of-view and pixel noise)
- Reaction Wheel Microvibrations

The system transfer, depending on the signal type, is automatically performed either via Eq. 3 as most accurate way to regard the propagation of time-correlation, or in terms of covariance via covariance propagation. The correlation of signals is fully considered in this computation.

The user can select pointing error indices (among those defined in Table 2) and associate a statistical interpretation (temporal, ensemble and mixed) to each index according to [1][2]. The pointing error indices are automatically computed by PEET, using a globally defined level of confidence.

PEET can be used throughout all design phases by successive modelling refinement. This means that starting with variances, mean values, and simple transfer models, the system description can be refined in later design phases by time-series data originating from simulations or by using PSDs with their cross-spectral densities together with sophisticated transfer models.

The interfaces to Java and analysis algorithms of the PEET core are implemented in an object-oriented manner using MATLAB classes. These classes are suitable for future extension and maintenance. The resulting concise structure inherently improves code re-use. The dedicated internal data structures and the corresponding categorization algorithms are suitable for both GUI-based computations and script-based computations. This enables the user to use batch mode operations and recursive computations. Algorithms for handling the signal transfer analysis history are developed, which support correct computation of signal correlations, and enable extension for user-specified correlation.

The current implementation of PEET assumes the applicability of the central limit theorem, i.e. non-Gaussian distributions are converted into equivalent Gaussian distributions. The user needs to be aware of this when evaluating the confidence level, in case of dominating non-Gaussian pointing error contributors. Furthermore, all PES are assumed to be stationary and systems transfers are treated as LTI models. Accurate computation of cross-correlation is currently limited to only full or no correlation of PES.

4. Applications

An artificial example of an high accuracy pointing satellite mission with typical pointing error sources and system transfers that convert these sources into final pointing error contributors is described in [13][14].

The pointing error engineering methodology and the current prototype PEET tool are already being used for the error analyses of diverse European satellites.

Euclid, in an orbit around the second Sun-Earth Lagrange point (L2), is an ESA mission to map the geometry of the dark Universe. Its AOCS is responsible for high pointing accuracy during science observation. The stringent pointing stability requirement (25 milli-arcsec rms over 500 s, TBC) require high performance AOCS sensors and a cold gas micro-propulsion system. MetOp-SG, in Sun synchronous orbit, forms the space segment of the EUMETSAT Polar System, and aim to collect consistent long-term remotely sensed data for meteorology and climate monitoring, forecasting and operational service provision. It is composed of multiple instruments, on several satellites, with various scanning requirements for pointing and pointing knowledge. PROBA-3 is a low-cost ESA mission for solar coronagraphy and technology demonstration for formation flying of spacecraft in a highly elliptic orbit. The two spacecraft will operate as a single unit, pointing at selectable directions, with a relative position accuracy at the millimeter level. EDRS is a geostationary constellation of GEO satellites intended to relay user data between LEO satellites (as well as UAVs in the future) and ground stations. The major requirements are on pointing knowledge and pointing stability, which is challenging with respect to a typical telecom satellite.

5. Conclusion

The methodology of the ESA PEEH and the accurate calculations supported in PEET can be beneficial in reducing the uncertainties in pointing budgets. This in turn can prove essential in guiding design decisions for high pointing accuracy missions, where comfortable margins with respect to requirements cannot be allocated.

The successful application results are paving the way for the development of an enhanced software framework that will develop specialized modules for Earth Observation, Science, Telecommunication and Navigation missions, which ESA will pursue in the near future. PEET is expected to become the reference tool for pointing error budgeting activities, processing complex calculations for high pointing accuracy and providing traceability in a common platform for exchange of information among the various stakeholders (ESA, industrial primes, industrial subcontractors, and scientific community).

6. References

[1] European Cooperation on Space Standardization, “Control Performance Standard ECSS-E-ST-60-10C.” ESA-ESTEC Requirements & Standards Division, 2008.

[2] ESA Engineering Standardisation Board, "Pointing Error Engineering Handbook ESSB-HB-E-003." ESA-ESTEC Requirements & Standards Division, 2011.

[3] Ott, T. Benoit, A. Van den Braembussche, P. and Fichter, W. "ESA Pointing Error Engineering Handbook.", 8th International ESA GNC Conference. Karlovy Vary, Czech, 2011.

[4] Lucke, R.L. Sirlin, S.W. and San Martin, A.M. "New Definition of Pointing Stability: AC and DC Effects." The Journal of the Astronautical Sciences, Vol. 40, No. 4, pp. 557-576, 1992.

[5] Pittelkau, M.E. "Pointing Error Definitions, Metrics, and Algorithms." American Astronautical Society, AAS 03-559, p. 901, 2003.

[6] Laureijs, R.J. Duvet, L. Escudero Sanz, I. Gondoin, P. Lumb, D.H. Oosterbroek, T. and Saavedra Criado, G. "The Euclid Mission." Proceedings of SPIE 7731. San Diego, USA , 2010.

[7] Kangas, V. D'Addio, S. Betto, M. Barre, H. Loiselet, M. and Mason, G. "Metop Second Generation microwave sounding and microwave imaging missions." Proceedings of the 2012 EUMETSAT Meteorological Satellite Conference. Sopot, Poland, 2012.

[8] Tarabini Castellani, L. Rodriguez, G. Llorente, S. Fernandez, J.M. Ruiz, M. Mastreau, A. Cropp, A. and Santovincenzo, A. "Proba-3 Formation Flying Mission." 7th International Workshop on Satellite Constellations and Formation Flying. Lisbon, Portugal, 2013.

[9] Bayard, D. S. "A State-Space Approach to Computing Spacecraft Pointing Jitter." AIAA Journal of Guidance, Control, and Dynamics, Vol. 27, No. 3, pp. 426-433, 2004.

[10] Bayard, D. S. and Neat, G. "Performance Characterization of a Stellar Interferometer." IEEE Control Systems Magazine, pp. 90-108, October 2007.

[11] Eggert, J. Hirth, M. Ott, T. and Su, H. "Pointing Error Engineering Tool – PEET Software User Manual." Issue 1.7, 26 July 2013. URL: http://peet.estec.esa.int/files/ASTOS-PEET-SUM-001_Iss1.7_PEET_Software_User_Manual.pdf.

[12] IEEE-Std-952-1997, Appendix B (R2008) "IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros."

[13] Hirth, M. Ott, T. and Su, H. "Pointing Error Engineering Tool - PointingSat Definition." Issue 1.3, 8 July 2013, URL: http://peet.estec.esa.int/files/ASTOS-PEET-TN-001_Iss1.3_PointingSat_Definition.pdf.

[14] Casasco, M. Saavedra Criado, G. Weikert, S. Eggert, J. Hirth, M. Ott, T. and Su, H. "Pointing Error Budgeting for High Pointing Accuracy Mission using the Pointing Error Engineering Tool." AIAA GNC Conference. Boston, Massachusetts, 2013.
.