AIAA Guidance, Navigation, and Control (GNC) Conference
August 19-22, 2013, Boston, MA
AIAA 2013-5251

Pointing Error Budgeting for High Pointing Accuracy Mission using the Pointing Error Engineering Tool

Massimo Casasco[1] and Gonzalo Saavedra Criado.[2]
European Space Agency - ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands

Sven Weikert[3] and Jochen Eggert[4]
Astos Solutions GmbH, Office Stuttgart, Meitnerstrasse 8, 70569 Stuttgart, Germany

and

Marc Hirth[5], Thomas Ott[6], and Haifeng Su[7]
Institute of Flight Mechanics and Control, University of Stuttgart, Pfaffenwaldring 7a, 70569 Stuttgart, Germany

Recent standardization efforts in Europe have led to the publication of the ECSS Control Performance Standard and the ESA Pointing Error Engineering Handbook, which are instrumental in defining a clear pointing error engineering methodology for ESA projects. To complement and support dissemination of this pointing error engineering methodology, the prototype of a software tool called Pointing Error Engineering Tool has been developed and released under the ESA Software Community License. This paper describes the mathemathical 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. An application example of the use of the Pointing Error Engineering Tool for a representative high pointing accuracy mission is provided to discuss the specific practical issues that may arise during the pointing error engineering process. Finally, the perspective activities to be promoted by ESA in the area of tools for pointing error engineering are outlined.

Nomenclature

A	=	amplitude of periodic signal
AKE	=	Absolute Knowledge Error
AOCS	=	attitude and orbit control system
APE	=	Absolute Performance Error
AST	=	analysis step
CRV	=	time-constant random variable
CTF	=	coordinate transformation
ECSS	=	European Cooperation on Space Standardization
ESA	=	European Space Agency
e_c	=	time-constant PEC

[1] GNC System Engineer, Control Systems Division, European Space Technology Center, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands, AIAA Member, E-mail: massimo.casasco@esa.int
[2] Concurrent Design Facility System Engineer, Systems and Cost Engineering Division, European Space Technology Center, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands
[3] Head of Development, Office Stuttgart, Meitnerstrasse 8, 70569 Stuttgart, Germany.
[4] Software Developer, Office Stuttgart, Meitnerstrasse 8, 70569 Stuttgart, Germany.
[5] PhD Candidate, Institute of Flight Mechanics and Control, Pfaffenwaldring 7a, 70569 Stuttgart, Germany.
[6] PhD Candidate, Institute of Flight Mechanics and Control, Pfaffenwaldring 7a, 70569 Stuttgart, Germany.
[7] PhD Candidate, Institute of Flight Mechanics and Control, Pfaffenwaldring 7a, 70569 Stuttgart, Germany.

e_{index} = instantaneous error
e_k = knowledge error signal
$e(k,t)$ = pointing error depending on the ensemble of realization index k and time t
$e_k(t)$ = pointing error realization with index k
$\{e_k(t)\}$ = ensemble of pointing error realizations $e_k(t)$
e_p = performance error signal
e_r = pointing error requirement
e_s = time-constant PES
D = drift
d_{nom} = nominal slew amplitude
f = frequency in [Hz]
G_{ee} = single-sided power spectral density
GSE = gyro-stellar estimator
GUI = Graphical User Interface
$H(j\omega)$ = linear time-invariant transfer function
h_{eff} = effective image angular size
h_{nom} = nominal image angular size
IR = infra-red
KDE = Knowledge Drift Error
KRE = Knowledge Reproducibility Error
k = index of specific ensemble realization
LTI = linear time-invariant
M = number of pointing scenes in vertical direction
MIMO = multiple-input, multiple-output
MKE = Mean Knowledge Error
MPE = Mean Performance Error
N = number of pointing scenes in horizontal direction
N_{thr} = number of thrusters used for attitude control
n_p = confidence coefficient
P_c = level of confidence
P_{ee} = linear spectral density
PDE = Performance Drift Error
PDF = probability density function
PEEH = Pointing Error Engineering Handbook
PEET = Pointing Error Engineering Tool
PES = Pointing Error Source
PEC = Pointing Error Contributor
PID = proportional, integral, derivative
PRE = Performance Reproducibility Error
PSD = power spectral density
p = probability density function
RKE = Relative Knowledge Error
RP = random process
RPE = Relative Performance Error
RV = time-random variable
S_{ee} = double-sided power spectral density
SISO = single-input, single-output
$T_{channel}$ = time required for image acquisition of a pointing scene in all channels
T_{slew} = time required for slewing between pointing scenes
t = instantaneous time
t_s = sampling time
XML = Extensible Markup Language
WN = white noise
Δt = window time
Δt_s = stability time
$\delta(f-f_0)$ = Dirac delta function at frequency f_0

ε_{RPE}	=	pointing variation in RPE window time
μ	=	mean
σ	=	standard deviation
ω	=	frequency in [rad/s]
ω_c	=	low-pass filter cut-off frequency

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

I. Introduction

POINTING error engineering covers the engineering process of establishing system pointing error requirements, their systematic analysis throughout the design process and eventually compliance verification. For technical as well as historical reasons, pointing error engineering in the European space community has long 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 situation has been changed by the initiative of the European Cooperation for Space Standardization (ECSS): the ECSS Control Performance Standard E-ST-60-10C[1], published in November 2008, provides solid and exact mathematical elements to build up a performance error budget. However, an additional document that provides guidelines and summation rules based on the top level clauses gathered in this ECSS-E-ST-60-10C standard was considered necessary by ESA to provide ESA projects with a clear pointing error engineering methodology. This methodology is the basis for a step-by-step process with guidelines, recommendations and examples consistent with and complementing the ECSS standard.

The answer to this necessity is the ESA Pointing Error Engineering Handbook[2,3] (PEEH) that was published in 2011 as ESA applicable document with the reference ESSB-HB-E-003. The ESA PEEH builds upon previous analytical work[4,5] that introduced a methematically well-founded approach to pointing error engineering. An accompanying SW tool that supports the ESA PEEH users in compiling pointing budgets was also considered beneficial in promoting dissemination and enhancing application of the Handbook. Such Pointing Error Engineering Tool (PEET) was envisaged to assist system engineers and control engineers both for the pointing requirements allocation activities, typically taking place in early phases of a project, and for the pointing error budget verification activities, taking place in later phases.

Astos Solutions GmbH, with the support of the Institute of Flight Mechanics and Control of the University of Stuttgart and under the supervision of ESA, has led the development of a PEET prototype, with the objective of implementing the step-by-step methodology proposed in the PEEH for early feasibility studies (Concurrent Design Facility-type pre-phase A studies) and phase A studies. This approach is intrinsically capable of minimizing the margins and uncertainties in pointing budgets and therefore is expected to prove extremely valuable especially for high pointing accuracy missions. Indeed, it is for this class of missions that the accurate analytical results that can be obtained using PEET could make the difference in taking the correct design decisions. ESA is currently using PEET for the pointing error engineering analyses of a number of missions, including Euclid[6], MetOp-SG[7], and Proba-3[8].

This paper provides a detailed summary of the pointing error analysis and evaluation methodology according to the PEEH in section II. Section III describes the implementation and features of the PEET prototype. An application of the use of PEET in pointing error budgeting activities for high pointing accuracy missions is presented in section IV. Conclusions and an overview of the foreseen future developments are finally provided in section V.

II. 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.

Nomenclature and definitions used in the ESA PEEH are introduced in section II.A, followed by the definition of the pointing error engineering methodology framework (section II.B) and of the four analysis steps for pointing error analysis and evaluation (sections II.C to II.F).

More details on the theoretical background of the ESA PEEH are available in Ref. 4 and Ref. 5.

A. Nomenclature and Definitions

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

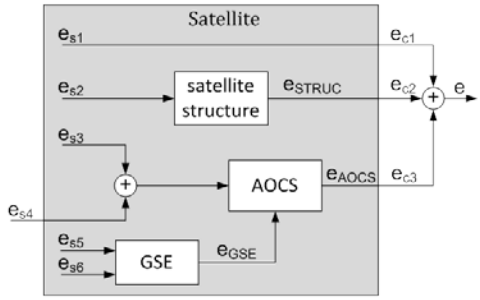


Figure 1. Pointing error source transfer.

Physical phenomena affecting pointing performance and thus the pointing error *e*, before undergoing any system transfer, are referred to as pointing error source (PES) and are denoted as e_s. A PES is 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 as e_c, represents the actual contribution of one or more pointing error sources e_s to the overall pointing error *e*.

A PES becomes a PEC after pointing system transfer, through e.g. the following transformations:

- coordinate frame,
- control system,
- structural.

In order to analyze pointing performance, a pointing system is broken down into subsystems with individually controlled (active or passive) transfer properties (see Figure 1). The pointing error *e* is the sum of the different PEC.

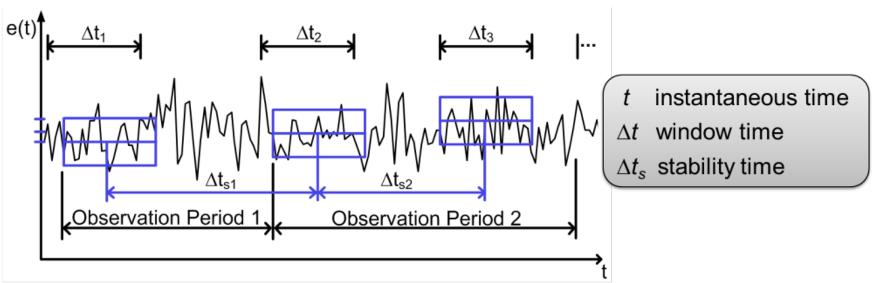


Figure 2. Time dependency of pointing errors

Several types of time dependencies of pointing errors can be distinguished (see Figure 2 for a graphical illustration of time-dependency types), which are dictated by 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 comprehensive set of pointing error indices, categorized in knowledge or performance errors and depending on instantaneous, window and stability time, is formulated in Table 1, where:

- AKE (Absolute Knowledge Error) is defined as the difference between the actual parameter (attitude, geolocation, etc.) and the known (measured or estimated) parameter in a specified reference frame
- APE (Absolute Performance Error) is defined as the difference between the target (commanded) parameter (attitude, geolocation, etc.) and the actual parameter in a specified reference frame
- MKE (Mean Knowledge Error) defined as the mean value of the AKE over a specified time interval Δt
- MPE (Mean Performance Error) defined as the mean value of the APE over a specified time interval Δt

- RKE (Relative Knowledge Error) defined as the 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) defined as the 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) defined as the difference between MKEs taken over two time intervals separated by a specified time Δt_s within a single observation period
- PDE (Performance Drift Error) defined as the difference between MPEs taken over two time intervals separated by a specified time Δt_s within a single observation period
- KRE (Knowledge Reproducibility Error) defined as the difference between MKEs taken over two time intervals separated by a specified time Δt_s within different observation periods
- PRE (Performance Reproducibility Error) defined as the difference between MPEs taken over two time intervals separated by a specified time Δt_s within different observation periods.

Table 1. Mathematical formulation of pointing error indexes.

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

B. Pointing Error Engineering Methodology Framework

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 shows a flow diagram for the pointing error engineering methodology.

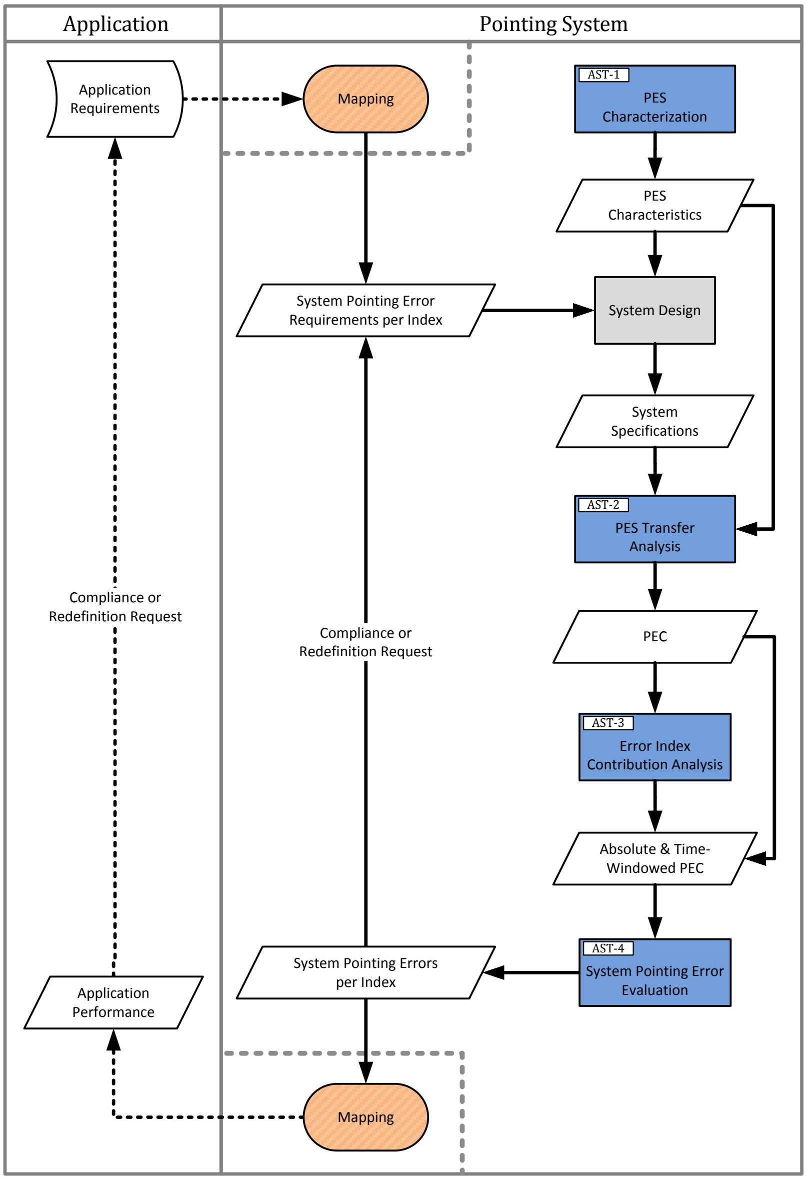


Figure 3. Pointing error engineering methodology workflow

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

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

This methodology in 4 analysis steps 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), $p_{\ldots}(e)$, and evaluation of level of confidence for required ECSS pointing error indices.

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

C. AST-1: Characterization of Pointing Error Source

Depending on the PES error data characteristics as well as on the type of available PES error data, the eligible mathematical elements for PES characterization are selected in AST-1. Depending on the maturity of the available data a PES can be described according to its fundamental properties:

- ensemble-randomness
- time-randomness
- power spectrum

The available PES error data is firstly classified based on their nature as either bias, periodic, Gaussian random, uniform random or drift. Subsequently, the decision tree in Figure 4 provides guidelines for systematically selecting a suitable PES description method.

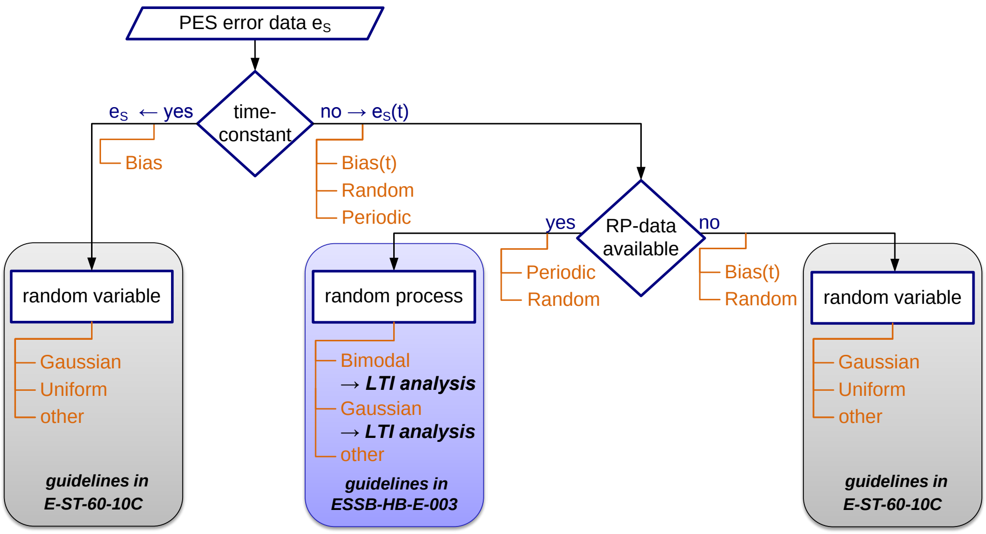


Figure 4. PES classification based on error data properties

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 both are also possible, meaning that a PES can be time-random and time-constant. 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 $\{\ldots\}$ 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.

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/$\sqrt{}$(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, Ref. 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 Ref. 3.

D. 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 AST-2 to determine the pointing error contributor (PEC). This can be done by breaking down the pointing system into subsystems, as exemplified in Figure 1 (or, equivalently, in Figure 9 of section III.C). These PEC's 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.

Different techniques in frequency and time domain exist for system transformation of time-random PES described as random processes. It can be distinguished between analytic methods, which are based on linear transformation of statistical properties, and numerical methods, which rely on simulations and experimental results. Time-constant PES system transformation analysis is a simple 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 in order 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 Ref. 1, but these approaches are not implemented in the PEET prototype.

E. AST-3: Pointing Error Index Contribution

Determination of the contribution of PEC's to the pointing error indices is performed in 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 (i.e. APE and AKE) that only depend on the instantaneous time, and not on the window time or stability time.

As highlighted in Figure 4, guidelines for evaluating the pointing error index contribution for the random variable description of time-random PEC's are provided in Ref. 1, in the form of tables that quantify the contribution for a number of different error probability distribution functions. These tables are reproduced and used in the PEET prototype when the random variable description of a PEC is selected.

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

A worked example that illustrates the application of the process in frequency domain that makes use of the weighting functions described above can be found in Ref. 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 statistical interpretation is done 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. To make clear which property and corresponding probability characteristic is described, it is necessary to define one of the three statistical interpretations[1]:

- mixed
- ensemble
- 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).

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

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).

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

F. AST-4: Pointing Error Evaluation

Pointing error evaluation is carried out in two steps: the time-constant and time-random error contributors are first combined together separately taking into account possible correlations between errors and the probability with the applicable confidence level is computed. Thereafter, the total pointing error is computed per error index from both intermediate results.

The rules for summing the means and variances of error contributors are the following:

- the means are summed linearly,
- the uncorrelated variances are summed quadratically,
- for the correlated variances, upper bound estimation is used.

The error index is then computed for the applicable confidence level. First the standard deviation is multiplied by n_p, where n_p is a positive scalar such that for a Gaussian distribution the n_p confidence level encloses the probability P_c, as specified in the requirement. Then the standard deviation is summed with the mean values.

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).

III. The Pointing Error Engineering Tool (PEET)

The Pointing Error Engineering Tool (PEET) implements the pointing error engineering methodology presented in the ESA PEE Handbook (PEEH), with a special focus on pre-phase A and phase A activities. PEET uses the computational power of MATLAB in combination with a specialized Java GUI to provide a system for pointing error evaluations. An interface to Excel is available in order to import data from Excel sheets and export data to

Excel files. Also data from the MATLAB workspace can be accessed by PEET. PEET is released under the ESA Software Community License.

A short overview over the system structure will be given in section III.A. The graphical user interface is shown in section III.B and the computational capability of the PEET core is explained in section III.C.

A. Overview

In this section, a short overview over the architectural structure of PEET is given (see Figure 5). PEET was designed to be fully integrated into the MATLAB environment. It consists of a graphical user interface (GUI) written in Java Swing and 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. For the communication between the Java GUI and the MATLAB core, the *Java Matlab Interface* (JMI) is used. This interface is provided by MATLAB and enables Java code to access the computational power of MATLAB. With the exception of the Control System Toolbox, PEET relies on a standard MATLAB installation.

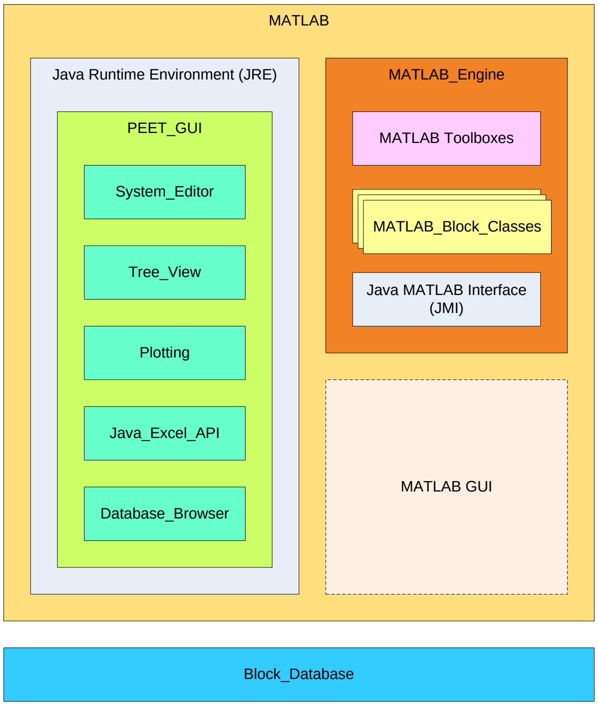


Figure 5. System overview

The user interface is very similar to Simulink. To build up pointing systems, the user can add various building blocks and connections between them. PEET provides a database which contains predefined building blocks applicable to pointing error engineering. This database is implemented as a single XML file which maps between the block types used in the GUI and the MATLAB classes available in the core.

B. Graphical User Interface

In a first step, the user has to define the pointing system. This is done in a way similar to Simulink. The user adds typical building blocks like pointing error sources (PES) or transfer systems to the so called *System Editor*. This system editor is shown in Figure 6.

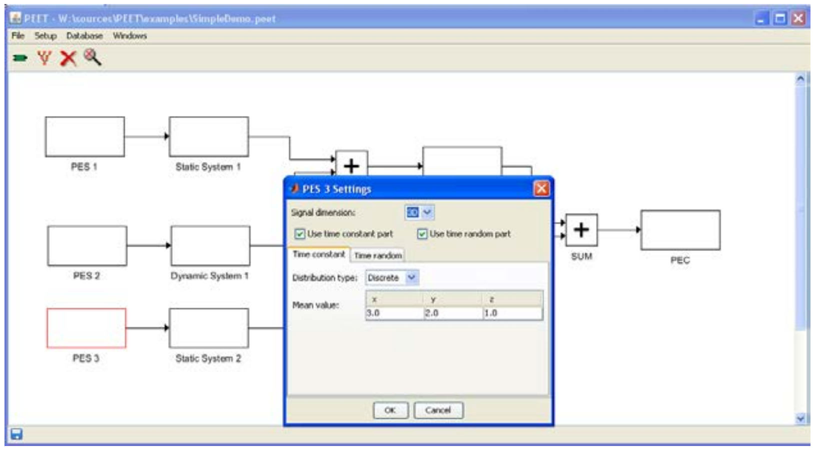


Figure 6. System editor and a block mask

Each block added to the system must be configured and connected to other blocks. For the configuration, each block type provides a block mask, which contains all available parameters for the block type. Figure 6 shows an example for the block mask of a pointing error source.

In addition to the definition of the pointing system, the error indices applicable to the pointing scenario can be defined by the user. Also correlation between error sources and correlation between axes of a single error source can be optionally defined in the first step. PEET supports uncorrelated and fully correlated error sources.

In a second step, the error computations are performed. The results of these computations can be inspected using the so called *Tree View*. An example of the tree view is shown in Figure 7.

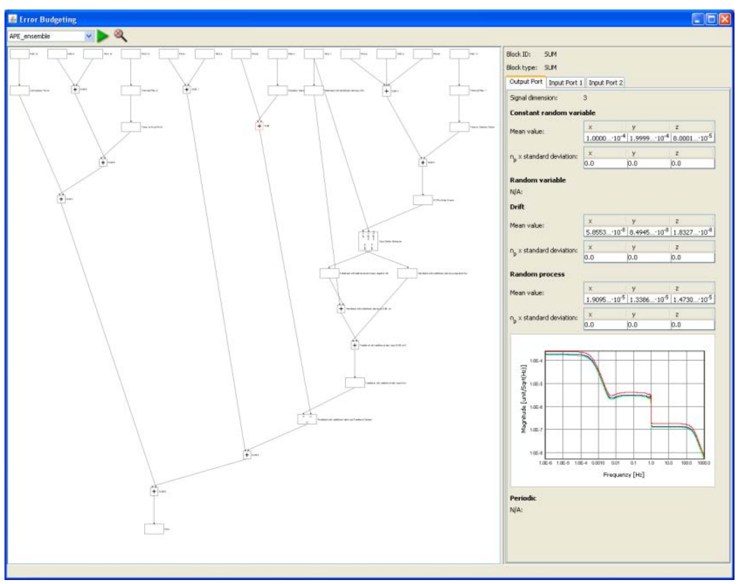


Figure 7. Tree View

On the left side of the tree view, the pointing system is shown in a tree-like structure. By selecting one of the blocks in the tree, the error information of the block's input and output signals shows up on different tabs. 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 owns 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.

A special feature of PEET is the sensitivity analysis. The sensitivity analysis uses difference quotients to compute a sensitivity value for each axis. In general, this sensitivity value expresses the change of the final pointing error in relation to a change of a scalar parameter value.

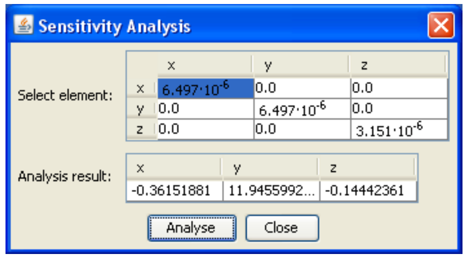


Figure 8. Sensitivity analysis manager

Figure 8 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.

C. PEET Core

PEET supports a wide range of pointing error sources which can be either 1- or 3-dimensional. Each PES can be described by its statistical properties or it can be described by a PSD (see section II.C) depending on the available data. Using the PSD representation, the time-correlation of the noise, i.e. its 'coloring' can be fully taken into account. 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). For the computations, each PES signal type is internally mapped either to PSD or to covariance matrices. This is necessary as the underlying treatment in the system transfer is different for each signal type.

The system transfer definition is a user task, but PEET provides tools and models in order to support the user in the definition. The system transfer from PES to contributors on pointing level is realized by the definition of various models like generic MIMO static systems (matrix) or dynamic systems (transfer function, state-space, zero-pole-gain, etc.). Inherited from generic systems, PEET also provides special blocks for most of the common models with special parameters. The models currently supported by PEET (and further described in Ref. 11) are:

- Coordinate Transformations
- Dynamics Systems
- Feedback Systems
- Flexible and Rigid Plant Model
- Gyro-Rate Noise (parametric model based on Ref. 12)
- Gyro-Stellar Estimator
- Mapping (used to map from 1D signals to 3D signals)
- Pointing Error Sources
- PID Controllers
- Static Systems
- Star Tracker Noise (parametric model for sensor field-of-view and pixel noise)

More models (e.g. for reaction wheel microvibrations) are under development.

Each of the models is internally converted to a state-space model (or linear combination of state-space models) in order to be computationally efficient and stable for the transfer of each signal type and to prepare an overall system model in MATLAB. 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, i.e. the coloring of the noise, or in terms of covariance via covariance propagation. During the system transfer, the correlation of signals is fully accounted for.

PEET provides the user the possibility to dselect various pointing error indices (among those defined in Table 1) and statistical interpretation options (temporal, ensemble and mixed) according to Ref. 1 and Ref. 2.

The related statistical interpretation and the pointing error index contribution are automatically performed by PEET.

The compilation of the total pointing error is performed by using the globally defined level of confidence. While, according to the PEEH, all pointing error contributors (PEC) are summed under the assumption of full or no correlation after system transfer, PEET allows the definition of these correlation states on initial PES level between axes of a single PES and between different PES. The correct correlation state is automatically accounted for in MIMO system transfer via a special signal history and no final assumption on PEC level (i.e. no upper bound approach as described in section II.F) is necessary. Indeed, after the system transfer and summation, the resulting signal can be retrieved from input signals to transfer systems and to summation blocks and from the initial setup of error source correlations, without further assumptions on the input signal correlation. And the key point to achieve this is to properly maintain and retrieve the signal route (history of transfer and summation, etc.).

PEET implements signal history as a built-in signal property. Each signal at different points of the pointing system keeps record of all its previous signal history, so that, eventually, the signal resulting from the combination of all the branches of the pointing system keeps track of the signal history for the whole pointing system. Each signal at a certain point of the pointing system behaves as a snapshot of the partial system transfer up to that point, and this enables the user to examine (by applying error evaluation) the contribution of the different systems and the effects of different system configurations.

PEET can be used throughout all design phases by a successive adaptation of the modeling degree. This means, starting from a system description using variance and mean values and by using 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.

PEET supports the pointing error break-down by proper grouping of subsystems (see Figure 9) and also in terms of a sensitivity analysis which can be used for the identification of error drivers in the pointing system.

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. This also inherently improves code reuse and results in a concise structure. 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 proper computations of signal correlations, and enables extension for user-specified correlation.

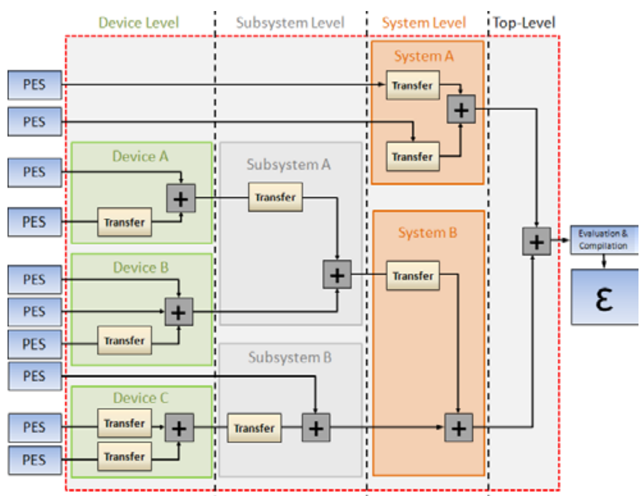


Figure 9. Proper grouping of subsystems

One of the possible limitations of the software at current status is that the pointing error evaluation is based on the central limit theorem (see section III.B), i.e. non-Gaussian distributions are converted into equivalent Gaussian distributions. The user has to be aware of this in case of driving non-Gaussian pointing error contributors. Furthermore, all PES are assumed to be stationary and systems transfers are treated as LTI models. Concerning cross-correlation, the current version of PEET does not yet include its accurate computation but only full or no correlation of PES.

IV. PointingSat Application Example

PointingSat describes an artificial high accuracy pointing satellite mission with typical pointing error sources and system transfers that convert these sources into final pointing error contributors. It serves as an example for a possible application of the methodology described in Ref. 2 concerning the setup and interpretation of pointing error sources and how this methodology can be implemented using the PEET software. A detailed description of the PointingSat scenario is provided with the software documentation[13].

This application example is modeled upon a geostationary mission supporting the disaster assessment and monitoring for the European continent. The primary payload of PointingSat is a high-resolution telescope for multispectral imaging that allows detection and tracking of different ecological, economical and humanitarian incident follow-ups such as fires, algal bloom spread, oil slick or infrastructural damages after earthquakes, floods or windstorms.

Since - depending on the incident to be observed - the areas to be monitored are much larger than the telescope field of view, highly accurate pointing and pointing stability of the satellite is required to allow single raster scanning of the relevant area and repeated scanning of the same area in different spectral ranges.

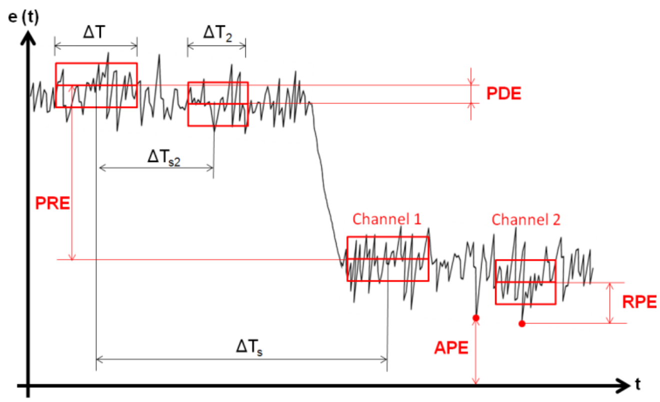


Figure 10. Relevant error indices for PointingSat application example

Above mentioned image acquisition strategy and multi-channel usage leads to requirements on different kinds of pointing errors which are represented by respective pointing error indices as illustrated in Figure 10.

First, the main mission objective is to point at the correct scene during one entire observation period (out of N_{tot} observations) with a probability P_c to guarantee that images are acquired from the actual pointing scene of interest in order to reliably detect disasters. This implies a requirement on the Absolute Pointing Error (APE) $e_{APE,req}$ and application of an ensemble interpretation, as it has to be ensured that the pointing error meets the requirement during 100% of time for one observation k (k = 1… N_{tot}).

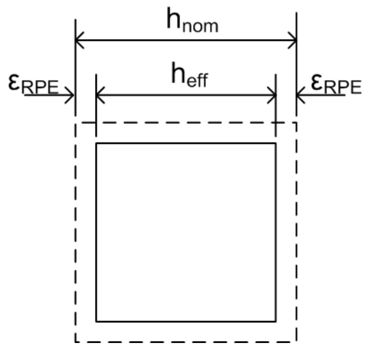
Second, the image quality is determined by the aberration of the point spread function during the integration time ΔT_i of a single observation of spectral channel i. Pointing variations during exposure lead to a broadening of the point spread function and thus to aberration[4]. These variations not only degrade the image quality, but the raster/mosaic scan is affected as well. Pointing variations ε_{RPE} during exposure lead to a narrowing of the effective field of view as areas at the edges of the image are not covered for the entire integration time. Consequently the nominal image size h_{nom} is reduced to an effective image size h_{eff} and gaps between adjacent images may occur (see Figure 11).

The need of a stable relative orientation throughout the integration time of the respective spectral channel leads to the definition of a requirement on the Relative Pointing Error (RPE) together with a window time ΔT corresponding to the maximum integration time out of the individual channels. As it has to be ensured that this requirement is met for at least a fraction P_c of the overall integration time with 100% probability for sufficient image quality of all N_{tot} observations, temporal statistical interpretation has to be applied in this case.

Finally, another requirement on a Performance Reproducibility Error (PRE) exists which is driven by the need of a proper orientation between adjacent images of the scanning raster in case the target area cannot be covered by the telescope field of view. Let ΔT_s represent the total time required for one observation cycle, i.e. to achieve the full mosaic image. If a raster of N x M pointing scenes is assumed (see Figure 12), ΔT_s can be derived from Eq. (8).

$$\Delta T_s = N \cdot M \cdot \left(T_{slew} + T_{channels}\right) - T_{slew} \tag{8}$$

where T_{slew} is the time required for the spacecraft reorientation to the next pointing scene and $T_{channels}$ represents the total time required for the image acquisition on all channels for one pointing scene. Note that generally another PDE requirement exists related to a stability time Δt_s = ΔT_{s2} (not treated here). This stability time corresponds to the adjustment/processing time between an acquisition of the same pointing scene on different channels. The respective window time $\Delta t = \Delta T_2$ is then identical to the minimum integration time ΔT of all channels.



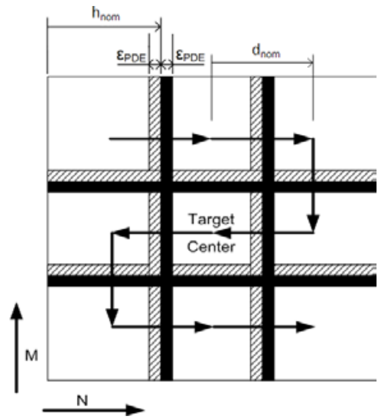


Figure 11. Impact of RPE on effective image size **Figure 12. Impact of PRE on raster pointing**

The effect of the PRE can be illustrated as follows: assume that the nominal size of the field of view for one raster image is given by h_{nom} and a slew $d_{nom} = h_{nom}$ is commanded between each of the different pointing scenes. Then - depending on its direction - a line of sight error motion of the first image (point spread function centroid) with respect to all other images (point spread function centroids) can result in gaps (black areas in Figure 12) or overlaps (dashed areas in Figure 12) between images.

For this PRE error, mixed statistical interpretation is required as both the temporal (stability during one observation) and the ensemble behavior (stability between subsequent observations) are of interest. The required window time ΔT is equivalent to the maximum integration time out of the individual channels as in the RPE case.

Dedicated values for the different requirements together with their levels of confidence, corresponding confidence coefficients and pointing error index parameters are summarized in Table 2.

Table 2. Summary of PointingSat requirements

Index	Requirement (Half-cone) [µrad]	Level of Confidence P_c [%]	Confidence Coeff. n_p (n_p x σ)	Statistical Interpretation	Pointing Error Index Parameters [s]
APE	450	99.7	3	ensemble	-
RPE	10	68.2	1	temporal	$\Delta t = \Delta T = 0.5$
PRE	50	99.7	3	mixed	$\Delta t = \Delta T = 0.5$, $\Delta t_s = \Delta T_s = 600$

Before giving an overview of the total pointing system setup, some more information on the PointingSat configuration is provided. The primary payload telescope is assumed mounted on an ultra-stable optical bench and the IR focal planes are housed in cryostats and cooled by mechanical cryocoolers. The PointingSat AOCS uses star-trackers (2 camera heads in cold redundancy) and fiber-optic gyros (3+3 cold-redundant) for attitude and rate determination via a gyro-stellar estimator. A set of 10 cold-gas thrusters (thrust range from 1 µN to 0.5 mN) is used for attitude control.

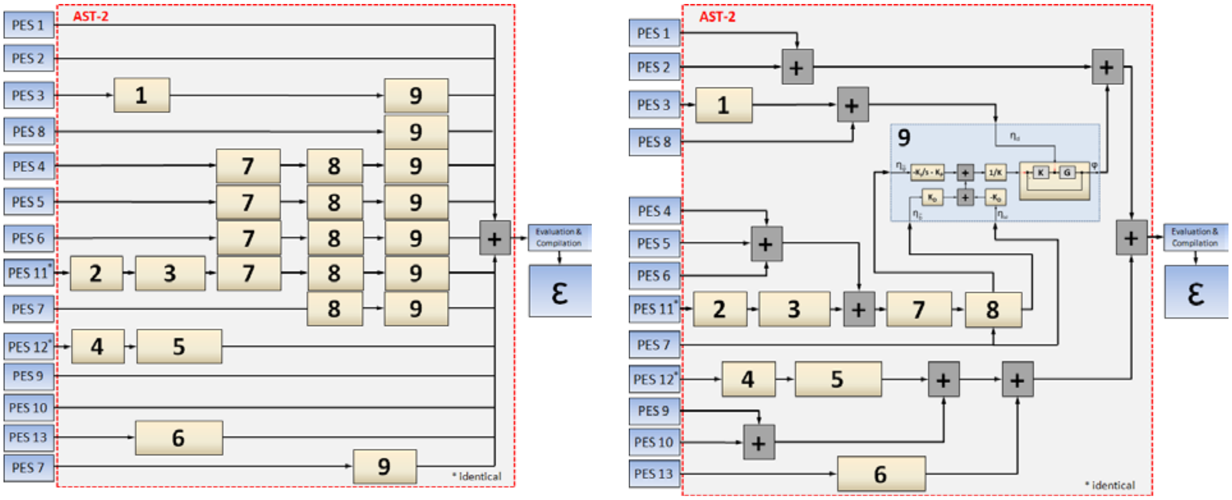
The set of PES identified for this configuration (according to AST-1, see section II.C) is shown in Table 3 together with the ECSS and PEEH error classification and information about the probability distribution of the sources. A more detailed description of each PES can be found in Ref. 13. The listed PES can be roughly categorized as integration & assembly errors (PES 1&2), actuator errors (PES 3), sensor errors (PES 4-7) and environmental & thermo-mechanical driven errors (PES 8-13). Table 4 lists all the systems models and PEET blocks required to map the initial error sources to error contributors on pointing level (AST-2, see section II.D). Figure 13 shows possible realizations of the overall pointing system setup, i.e. the interconnection of PES and systems.

Table 3. Summary of PointingSat pointing error sources (WN: white noise, CRV: time-constant random variable, RV: time-random variable, RP: random process, D: drift)

PES	ECSS Description	PEEH Description	PES Random Properties temporal	ensemble	Physical Source of Error
1	Bias	CRV	-	Uniform	Star tracker - telescope misalignment after assembly
2	Bias	CRV	-	Gaussian	Launch effect on star tracker - telescope misalignment
3	WN	RP	PSD		Force noise of individual thruster
4	Bias	CRV	-	Uniform	Star tracker measurement bias
5	WN	RP	PSD		Star tracker temporal measurement noise (temperature-/orientation- independent part)
6	WN	RP	PSD		Star tracker FOV and pixel noise
7	WN	RP	PSD		Rate noise of gyro assembly
8	Bias D WN	CRV D RP	- - PSD	Discrete Discrete	Time series of environmental disturbances on satellite body from simulation
9	Harmonic	RP	Bimodal	Uniform	Periodic thermo-elastic distortion affecting the alignment between payload and star-tracker
10	WN	RV	Gaussian	Uniform	Random thermo-elastic distortion affecting the alignment between payload and star-tracker
11	WN	RP	PSD		Thermal stability effect on star tracker measurement
12	TR	RP	PSD		Thermal stability effect on focal point
13	Harmonic	RP	Bimodal	Discrete	Cryocooler induced micro-vibrations

Table 4. Summary of PointingSat system transfer blocks

System	Block	Description
1	Mapping	N_{thr}x3 thruster actuation matrix converting noise of N_{thr} identical thrusters to torque noise on physical axis
2	Dynamic System	Thermal filter describing structural temperature stability "damping" between optical bench reference and star tracker camera head
3	Static System	Simplified static model converting temperature stability into equivalent pointing error due to thermal expansion of star tracker detector
4	Dynamic System	Thermal filter describing structural temperature stability "damping" between optical bench reference and telescope focal point
5	Static System	Simplified static model converting temperature stability into equivalent pointing error due to thermal expansion effect on focal point
6	Dynamic System	Simplified structural damping model describing effect of cryocooler micro-vibrations on pointing
7	CTF	Coordinate transformation from star tracker to telescope frame
8	GSE	Gyro-Stellar-Estimator providing attitude and rate bias estimation errors
9	Feedback	Simplified AOCS closed-loop model with disturbance and measurement noise inputs
+	Summation	Summation of any block input signals



a) **b)**

Figure 13. Different realizations of the PointingSat system in PEET: “Parallel” setup (a), “Compact” setup (b) with PES and system numbering according to Table 3 and Table 4

The following paragraphs highlight dedicated practical issues during the pointing error engineering process using this application example rather than providing numerical results for the PointingSat scenario itself (which are available in Ref. 13).

A. Pointing system setup

A first issue is directly related to the system setup approach using the PEET software. As indicated in Figure 13, one and the same system can be fully equivalently represented in multiple ways. The strategy described in the PEEH is closely related to the approach in Figure 13a): each PES is individually transferred to a PEC in parallel before finally all PEC are summed in a final step. This configuration allows a quick determination of the driving PES using the software’s analysis and display features. Possible disadvantages arise from the multiple usages of system blocks which increase the number of signals and system models to be computed. In addition, it is also possible to create a compact system setup in terms of block usage as in Figure 13b), which allows quick parameter changes and investigation of their impact on the total pointing error, or to use a subsystem approach as in Figure 9 for a compromise. Thus, it is always up to the pointing error engineer to set up a system that fits best into the current application purpose.

B. Random process description in terms of PSD

As an extension to the error classification described in Ref. 1, the PEEH (and PEET) covers a frequency-domain approach which allows the definition of PES as PSD. This significantly increases the possible modeling degree and propagation accuracy of the error source through transfer systems compared to the white noise random variable description of Ref. 1. The latter defines the error source by a variance only. From a frequency domain point of view, the variance corresponds to the area enclosed by the PSD and a description by variance only is ambiguous. For instance, all power spectral densities G_i in Figure 14 have the same initial variance but different error contributions after error index evaluation due to the frequency dependence of the applied metric filters. Thus - as it is most accurate for error propagation through system transfers - a description in terms of PSD is always favorable provided that random process data is available.

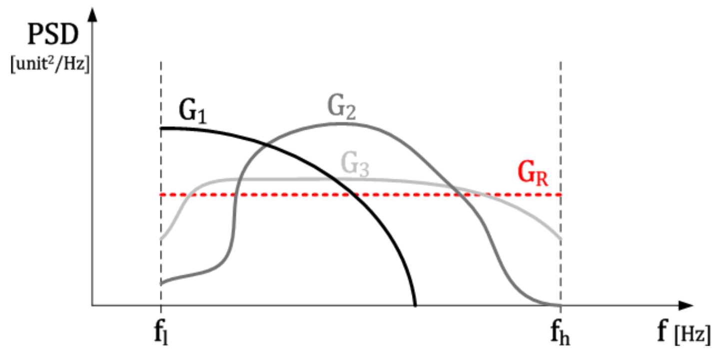


Figure 14. Exemplary spectra with identical variance

If only the standard deviation σ_n together with a sampling time t_s of the noise is known, the assumption of a band-limited white noise with equivalent standard deviation and cut-off at the Nyquist frequency of the sampling time can be made, e.g. by using an *m*-th order low-pass filter to obtain a spectrum of the form shown in Eq. (9).

$$P_{nn} = \sqrt{G_{nn}} = \sigma_n^{-1}(s + \omega_c)^m = \sigma_n^{-1}(s + \pi t_s^{-1})^m \tag{9}$$

This model is still a better approximation and more exact in terms of system transfer than defining a white noise random variable PES only. This latter option is implemented in PEET and used for instance for the definition of PES 5.

C. Modeling of periodic PES

The classification of periodic PES in Ref. 1 distinguishes only between low period and high period harmonic errors (relative to the observation period) and provides rules to compute an equivalent Gaussian mean and variance dependent on the chosen error index and statistical interpretation. Consequently, the treatment of periodic sources is always an approximation and a decision in favor of one of the categories is ambiguous in case the period of the harmonic error is close to the observation period.

The PEEH describes a more sophisticated approach by modeling the harmonic error source with an underlying bimodal distribution in terms of a "discrete", (i.e. single peak) spectrum. This power spectral density G_p is given by Eq. (10).

$$G_p = \frac{1}{2}A^2\delta(f - f_0) \tag{10}$$

where $\delta(f\text{-}f_0)$ is the Dirac-Delta function at the frequency f_0 of the periodic error source with amplitude *A*. The advantage of this description is that the noise power can be propagated exactly when feeding the source through any transfer system or when applying the pointing error metric filters for the chosen error index (by multiplication of the system or metric filter magnitude at the periodic frequency). Furthermore, this approach allows the definition and proper treatment of periodic errors at multiple harmonic frequencies in one single PES (as is the case e.g. for PES 13).

D. Time-constant/-random PES description and temporal/ensemble distribution of random variables

A more general issue related to the PES definition is a clear distinction between time-constant and time-random behavior and a correct interpretation and mapping of the probability distribution of the error source.

Time-constant PES correspond to what is commonly understood as "bias" (CRV class in PEEH), i.e. a source that does not change in the timescale of the mission phase under consideration. For instance, the misalignment between star-tracker and telescope after assembly and after launch (PES 1&2) does not vary between and during different observation periods. Any statistical properties of this type of error sources are consequently related to the ensemble distribution of the error (e.g. PES 1 has an underlying uncertainty in the integration process or different – but constant – errors in case of multiple satellites). Mean values of time-random variables (RV class in PEET) also belong to this class as they are independent of time, i.e. $e_{CRV} = \mu(e_{RV}(t)) \neq \mu(t)$. They are however defined as part of the time-random part (in the PEEH and PEET) to avoid the splitting of properties of one single physical PES and PEET performs the correct mapping internally.

Time-random variables are primarily defined by the statistical properties of their temporal distribution, e.g. a Gaussian distribution is used to describe the random misalignment between star tracker and telescope due thermo-elastic distortion (PES 10). During a single observation period, this description of the error can be considered as sufficient. However, the variance of this source may depend on the operational conditions (e.g. temperature) which may vary between different observations. Then the PES model requires an additional description of its ensemble behavior which means that the statistical properties become random variables themselves. In case of PES 10 the (temporal) variance of the error is uniformly distributed between a minimum and a maximum value assuming a linear effect of the steady-state temperature on the variance.

This combined description of temporal and ensemble behavior is also possible for other time-random source classes such as periodic or drift class PES, e.g. when the amplitude of the periodic error (PES 9) or a drift rate is dependent on the operational conditions. However, time-random PES of PSD type need some special treatment. Assume that the force noise spectra of one single cold-gas thruster (PES 3) have been recorded by the manufacturer in test facilities dependent on e.g. the tank temperature and thrust level as shown in Figure 15 and each spectrum represents a stationary process. In this case, the given information is not sufficient for a proper description of the ensemble distribution (in terms of the mentioned parameters) and the worst case spectrum should be chosen.

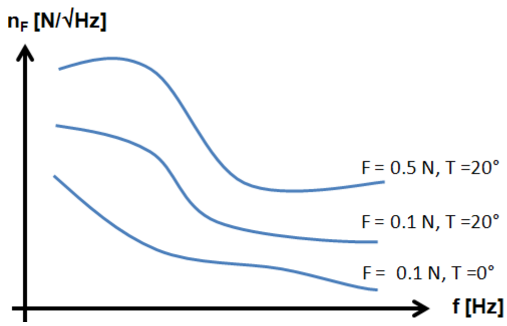


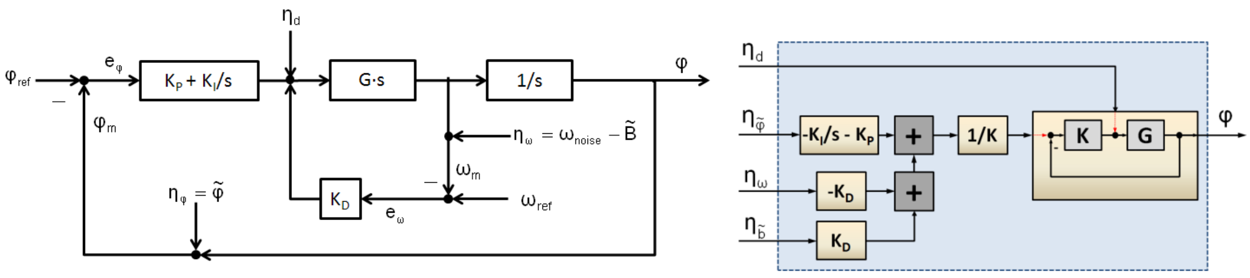
Figure 15. Exemplary thrust noise spectra for different conditions

E. Feedback system realization

In most pointing scenarios the implementation of feedback loops in the error model is beneficial. The PEET software supports such implementation by the Feedback Block (see section III). This block allows multiple input signals but does not support nested loops which is however not a real restriction. In the PointingSat scenario for instance, attitude control is based on rate and attitude feedback which basically corresponds to a nested-loop structure as shown in Figure 16a) and is not compatible with the PEET Feedback Block. However, loop manipulation (inner loop reduction, moving summing junctions, cascading blocks, etc.) allows the setup of a totally equivalent representation of the nested loop (see in Figure 16b), which allows implementation in PEET.

Alternatively, each input signal could be fed individually through its respective closed-loop transfer function (using standard PEET dynamic system blocks) without usage of the Feedback Block. This however complicates the quick investigation/modification of e.g. controller or plant parameters.

At this point it is important to note that the PEET software is no control design tool, i.e. it provides no statements about the system stability: parameter modification in the control system loop may lead to a better pointing performance, but also destabilize the system. This is an issue which has to be carefully assessed by other means or tools.



a) **b)**

Figure 16. Nested closed-loop system (a) and equivalent realization in PEET after loop manipulation (b)

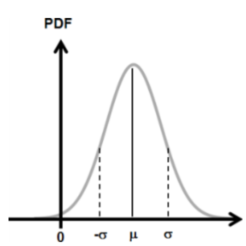
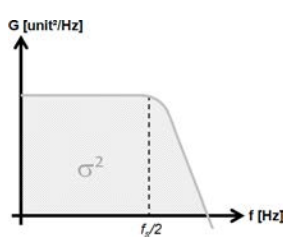
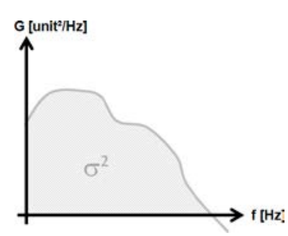
F. Time-series handling

Within a typical project development, various simulation tools will be used that could provide time-series data of potential error source. In the PointingSat example, PES 8 is represented by time-series of the mainly solar pressure noise driven external disturbances from simulations for a comparable geostationary satellite. While the processing of this data could also be handled by external tools, PEET provides the option of a direct import of time-series for PES. Before computing the power spectrum the data is de-trended (removal of mean and linear trend) which leads to a CRV, drift and random process contribution of the respective PES. The computation of the PSD (auto- and cross-spectra for 3D signals) is based on algorithms from the LTPDA toolbox[14].

G. PEET usage throughout different development phases

Although a special focus was set on pre-phase A and phase A activities for the software prototype, PEET is generally capable to support the pointing error engineering process throughout all project phases. In early phases, the pointing system might be established by using highly simplified descriptions of the error sources (e.g. by random variables only). The same is true for the system transfer models where certain "error branches" might be fully represented by PES as placeholder only, i.e. without a dedicated description of the transfer. If more sophisticated information/data becomes available in the project development (see scheme in Table 5), the existing PEET model can be adapted step-by-step by introducing new error sources or transfer models, or replacing by more substantial models. For instance, RV type PES can be replaced by random process descriptions in terms of PSD which are then transferred through dynamic models of growing complexity, or time-series from external simulation tools can be directly imported.

Table 5. Increasing modeling degree and accuracy

	project phase, error source information		
Available PES Data	μ, σ	$\mu, \sigma, f_s=1/t_s$	$\mu, G(\omega)$
Description	Random Variable 	Random Process μ & 	Random Process μ & 

This is also reflected in the PointingSat application example, where partially detailed PES models are implemented but also very simplified models for the system transfer (e.g. the static conversion models of systems 3 and 5). Furthermore the collection of PES and system transfers does not cover all possible error sources and contributions as its main purpose is to show the potential of error source and system representations with PEET.

Although the PEET block database is not exhaustive, the software generally allows the user to define own system block types with arbitrary complexity of the models (by extending the existing PEET MATLAB classes) with the only restriction that the input/output structure of the signals is maintained.

V. Conclusion

The methodology and features of the PEET prototype have been presented and showcased with application to an artificial pointing problem. The paper has highlighted how the accurate calculations supported in PEET (and in general the methodology proposed in the ESA PEEH) can be beneficial in reducing the uncertainties in the pointing budgets, which could prove essential in driving the design decisions for high pointing accuracy missions, where comfortable margins with respect to requirements cannot be allocated. The current PEET prototype is being used at the European Space Agency to implement and analyze the pointing budgets of a number of missions, including Euclid, MetOp-SG and Proba-3. In addition, the successful results obtained with the PEET prototype are paving the way for the development of an enhanced SW 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 in ESA missions that need to process complex calculations for high pointing accuracy and to provide traceability in a common platform for exchange of information among the stakeholders (ESA, industrial primes, industrial subcontractors, scientific community, etc.).

References

[1]ECSS, "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., P. Van den Braembussche, W. Fichter, "ESA Pointing Error Engineering Handbook", *8th International ESA Conference on Guidance, Navigation & Control Systems*, Karlovy Vary CZ, June 2011.

[4]Lucke R.L., Sirlin S.W., San Martin A.M., "New Definition of Pointing Stability: AC and DC Effects", *The Journal of the Astronautical Sciences*, Vol. 40, No. 4, p. 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., Saavedra Criado G., "The Euclid Mission", *Proceedings of SPIE*. 7731 (2010)

[7]Kangas V., D'Addio S., Betto M., Barre H., Loiselet M., Mason G., "Metop Second Generation microwave sounding and microwave imaging missions," *Proceedings of the 2012 EUMETSAT Meteorological Satellite Conference*, Sopot, Poland, 3-7 September, 2012

[8]Tarabini Castellani L., Rodriguez G., Llorente S., Fernandez J.M., Ruiz M., Mastreau A., Cropp A., Santovincenzo A., "Proba-3 Formation Flyinf Mission", *7th International Workshop on Satellite Constellations and Formation Flying*, Lisbon Portugal, 13-15 March, 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, May-June 2004.

[10]Bayard D. S., Neat G., "Performance Characterization of a Stellar Interferometer", *IEEE Control Systems Magazine*, October 2007.

[11]Eggert J, Hirth M., Ott T., Su H., "Pointing Error Engineering Tool – PEET Software User Manual", Astos Solutions GmbH, Rept. ASTOS-PEET-SUM-001 Issue 1.5, 19 December 2012, URL: http://peet.estec.esa.int/files/ASTOS-PEET-SUM-001_Iss1.5_PEET_Software_User_Manual.pdf [cited 26 July 2013]

[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., Su H., "Pointing Error Engineering Tool - PointingSat Definition", Astos Solutions GmbH, Rept. ASTOS-PEET-TN-001 Issue 1.2, 19 December 2012, URL: http://peet.estec.esa.int/files/ASTOS-PEET-TN-001_Iss1.2_PointingSat_Definition.pdf [cited 26 July 2013]

[14]Hewitson M. et al., "Data analysis for the LISA Technology Package", *Classical and Quantum Gravity*, vol.26 no.9, 2009