GNC 2014: 9th International ESA Conference on
Guidance, Navigation & Control Systems
Porto, Portugal | 2 - 6 June 2014
European Space Agency

Pointing budgeting using the ESA Pointing Error Engineering Handbook and Tool: benefits and limitations

Ott T.[1], Wiedermann G.[1], Crombez V.[2], Damilano P.[2]
[1]Airbus Defence and Space, Germany. [2]Airbus Defence and Space, France.

ABSTRACT

In 2011 ESA published the ESA Pointing Error Engineering (EPEE) Handbook as applicable document. Based on it ESA recently developed a software prototype of the Pointing Error Engineering Tool PEET to support the user in applying the process and techniques in the EPEE Handbook and in the respective ECSS standards. This article provides an industry perspective of using the EPEE Handbook and PEET software for pointing error engineering and thus the compilation of pointing budgets. An earth observation LEO satellite similar to a satellite in the MetOp-SG mission serves as general benchmark to assess the applicability and analyse the resulting budgets of the following three approaches: (1) ECSS standard with classical summation rules, (2) EPEE Handbook with ECSS standards, (3) PEET software and thus implicitly the EPEE Handbook with ECSS standards. An operational scenario of MetOp-SG is analysed with the simplified statistical method, which is based on the specific summation of random variables. In addition, the attitude control system of a precision pointing satellite in a fine pointing science observation scenario serves as another benchmark to make use of the advanced frequency-domain methods in the EPEE Handbook and PEET. These methods are based on the summation of random processes and their system transfer. The benchmark scenarios in this article eventually serve as input to assess the different approaches and discuss the benefits and limitations in applying the EPEE Handbook and PEET with respect to the needs in pointing error engineering. To this end a comprehensive summary of needs is derived based on the challenges and experiences of Airbus Defence and Space in recent and current ESA missions. The paper concludes by suggesting future research and development directions.

1. INTRODUCTION

As pointing error engineering is relevant to any space mission, ESA together with industry and research institutes identified the need to support technology developments and research in this field. As results the ESA pointing error engineering (EPEE) Handbook [1] and a prototype of the Pointing Error Engineering Tool PEET were released. The EPEE Handbook was introduced in [2] and PEET in [3]. Both developments address the needs to cope with pointing error engineering challenges in current and future ESA missions. A representative set of such challenges is given in Table 1.

Table 1: pointing error engineering challenges in future ESA missions

challenge	mission	driving requirement					
		math. type	specification				
			index	Δt [s]	Δt_S [s]	P_c [%]	arcsec
high precision	**GEO-HR**: Geostationary High Resolution earth observation mission	scalar	RPE PDE	~ 0.1 ~ 0.1	- < 5	68.3 99.7	~ 10e-3
	Euclid: science mission observing the dark universe	scalar	RPE	< 700	-	68.3	~ 10e-3
cost-efficient design process	**MetOp-SG**: Meteorological Operational Satellite - Second Generation	scalar	AKE	-	-	99.7	~ 100
accurately deal with non-scalar requirements	**PLATO**: science mission detecting terrestrial exoplanets in the habitable zone of solar-type stars	PSD	pointing error PSD has to be below specified magnitude bound in a certain frequency band				
	Lisa Pathfinder: science mission to demonstrate the technology for detecting low-frequency gravitational waves						
	EDRS: data relay GEO satellite with broadband inter-satellite laser link to user LEO satellites	PDF	accurate determination of pointing error PDF to determine link availability				

Reliably achieving high precision pointing requirements is one of the challenges. The Euclid mission [4] and the mission study GEO-HR [5] are examples in that case. However, as international competition becomes stronger another very important challenge is to achieve a cost-efficient design process across a project team (ESA, prime-contractor, sub-contractor, consultants). This is where the standardized satellite platform approach serves as reference to perform pointing error engineering such that it combines heritage with flexibility for customization: standard architecture, modularity, ECSS compatibility. The MetOp-SG mission [6] is an example in that case. The third challenge is to accurately deal with distinct requirements that cannot be formulated in line with the ECSS Control Performance Standard [7]. An example in this respect is the science mission PLATO [8] that shall detect terrestrial exoplanets in the habitable zone of solar-type stars. In this case it is necessary limit the noise level in the frequency band of interest to detect oscillatory signals. The same type of pointing requirement is specified for Lisa Pathfinder mission [9] and exemplary shown in Figure 1on the left.

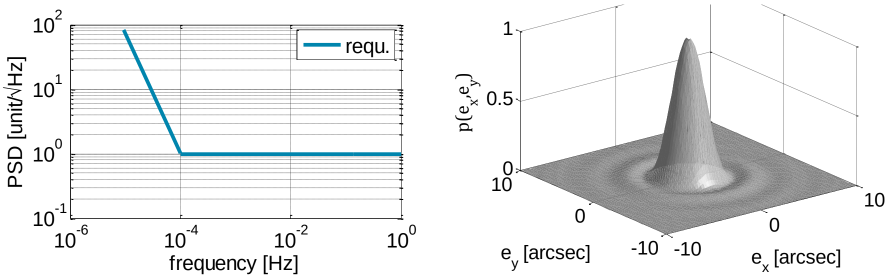


Figure 1: power spectral density requirement (left) and probability density function requirement (right)

Another type of requirement has to be achieved by the Laser Communication Terminal (LCT) on the EDRS satellites [10]. Here it is necessary to accurately determine the probability density function (PDF) of the LOS uncertainty cone of the LCT as shown in Figure 1 on the right. The uncertainty cone serves as input to determine the time until a link is established between two LCT. This time drives the link availability time and thus the profitability of a business case.

The above mentioned challenges have to be addressed by pointing error engineering. However, before this request is analysed a concise definition of pointing error engineering is given and the terminology used in this paper is defined. The activities in pointing error engineering are divided in apportionment and budgeting as shown in Figure 2. The budgeting activities characterize the pointing error sources (PES), analyse their pointing system transfer (dynamic and static transformations), determine their error index contribution in view of the statistical interpretation and finally quantify the pointing error contributors (PEC) to the system pointing error budget, cf. [1]. The apportionment is concerned with the break-down of the overall pointing error requirement (PER) to allocate fractions of it to different levels: unit, subsystem, system. Apportionment and budgeting span the V-diagram in Figure 2. The V-diagram draws the pointing error engineering process in view of verifying and validating pointing performance or knowledge. The budgeting activities in the process are supported by ECSS documents, the EPEE Handbook and PEET. An overview in this respect is given in section 3. The apportionment is currently not supported by any standardized documents or tools.

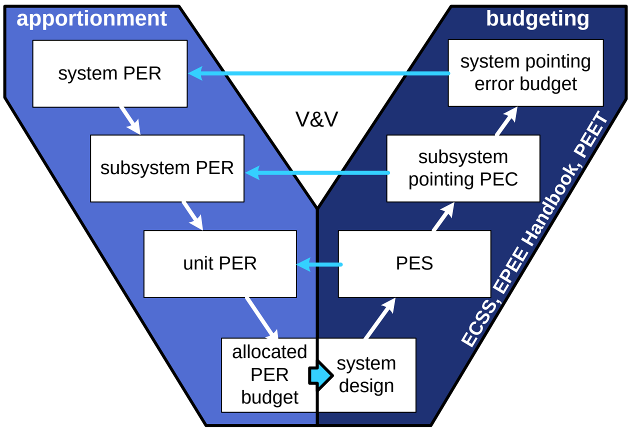


Figure 2: pointing error engineering process in a verification and validation illustration

The focus of this paper is on the industrial, practical and theoretical needs in pointing error engineering to successfully deal with the above mentioned challenges in the design of satellites. In chapter 2 five categories with needs are derived based on the gathered experience of Airbus Defence and Space in the projects of Table 1 and others. A summary of the current pointing error engineering framework is given in chapter 3. The focus is hereby on developments like standards, handbooks and software that are available within the ESA community. Out of these developments the application of the ECSS documents, the EPEE Handbook and PEET are analysed in a benchmark from an industrial perspective with respect to the derived needs. One benchmark case study is the analysis of a earth observation LEO satellite based on the MetOp-SG mission. In this benchmark a typical operational scenario is analysed with the simplified statistical method, which is based on the specific summation of random variables. The attitude control system (ACS) of a Euclid-like high precision mission in fine pointing science observation mode serves for another benchmark to make use of the advanced frequency-domain methods in the EPEE Handbook and PEET. These methods are based on the summation of random processes and their system transfer. The benchmark objectives and case studies are introduced in chapter 4. The benchmark results are summarized in the same chapter and evaluated in chapter 5. The evaluation shows if and to which extend the current pointing error engineering framework addresses the needs in chapter 2. The focus of the evaluation is on the benefits and limitations in the application of the ECSS documents, the EPEE Handbook and PEET. The evaluation results are complemented by lessons learned in Airbus Defence and Space.

2. NEEDS IN POINTING ERROR ENGINEERING

The following five categories of needs in pointing error engineering are derived based on the challenges and thereby gathered project experience of Airbus Defence and Space:

- *Optimal Engineering Process*
- *Approximate Budgeting and Accurate Analysis*
- *Robustness Guarantee*
- *Sensitivity Analysis*
- *Requirements Engineering Support*

The categories with their needs are introduced and justified hereafter.

A. OPTIMAL ENGINEERING PROCESS

The need of an optimal engineering design process with guidelines and tools is especially important because pointing error engineering is a topic throughout the entire mission design lifecycle and a key design driver for several missions as shown in chapter 1. The "optimal" stands for a design process that is:

- **integrated**: unified methodology with exact mathematical elements and practical guidelines that enables a systematic multidisciplinary and continuous design flow with standardized and coherent interfaces between design steps, phases, disciplines and project partners
  - ⇨ moreover, standardization enables continuous further development in pointing error engineering
- **responsive**: quick assessment with the right level of detail on system level
  - ⇨ minimum involvement of analysis by various engineering disciplines
- **tailorable**: to the right level of detail and accuracy for the corresponding design phase and mission type
  - ⇨ design level of detail determines the accuracy level of design techniques, which raises the need of having approximate budgeting and accurate analysis techniques

The introduced optimality properties are general and often only linked to process methodology. However, it is also necessary to embed suitable techniques in the process methodology to achieve a high quality but cost-efficient design solution. In this context "high quality" stands for a design that reliably complies with the mission requirements. A cost-efficient design is achieved by finding the balance between engineering costs and costs caused by design risks. This is illustrated in Figure 3. The greater the level of detail is the higher the costs and the lower the design risk, cf. [11]. These curves, and thus the cost optimum, change throughout the mission design lifecycle. In pointing error engineering it had been difficult to achieve this optimum due to the lack of an integrated, responsive and tailorable step-by-step design process and tool. On system level (pointing error engineering = system level task) such a design process and tool is especially important in order to reduce design iterations that involve comprehensive analysis of various engineering disciplines.

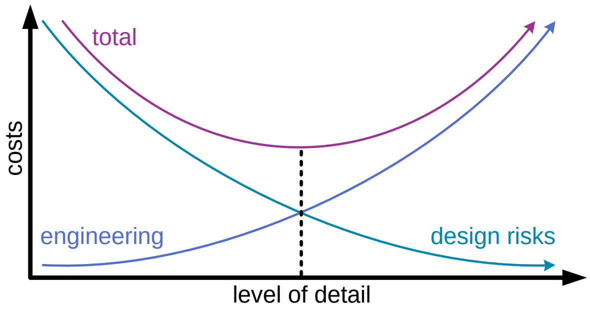


Figure 3: optimal pointing error engineering

SUMMARY

- *tailorable process to right level of detail and accuracy for respective design phase and mission type*
- *techniques and methods responsively useable on system level*
- *integrated process based on unified methodology with:*
    - *exact mathematical elements*
    - *practical guidelines*
    - *systematic and multidisciplinary design flow*
    - *standardized and coherent interfaces*
    - *continuous (thus hybrid) flow from approximate to accurate budgeting techniques*

B. APPROXIMATE BUDGETING AND ACCURATE ANALYSIS

Approximate budgeting and accurate analysis techniques and methods are needed to handle pointing error knowledge and performance throughout the mission design lifecycle as requested in section A. In this sense the approximate budgeting needs to be supported by:

- case-by-case rules for PES modelling and statistical interpretation
    - ⇨ most frequent occurring PES cases are covered in a look-up table
- error mapping guidelines: axis to LOS
    - ⇨ worst case, but low level of conservatism and mathematically rigorous, guidelines for mapping axis to LOS errors

The accurate analysis needs to be supported by:

- general rules for PES modelling and statistical interpretation
- accurate error modelling: PSD, PDF, cross-correlation among PES and axes
- exact determination of the PDF to evaluate the level of confidence if:
    - ⇨ central limit theorem in [1] applies: the convolution of PES with different PDF in the same order of magnitude converges to a Gaussian PDF
    - ⇨ central limit theorem does not apply
- MIMO Linear Time Invariant (LTI) system transfer and pointing error index contribution of PES that can (and also cannot) be described as stationary random processes
    - ⇨ MIMO because several PES act simultaneously on three axes
    - ⇨ LTI because in that case there is the following simple relation for transforming a PSD, $\mathbf{G}_{ss}$, of the PES, $\mathbf{e}_s$, through a pointing system, $\mathbf{H}$, to obtain the PSD, $\mathbf{G}_{ee}$, at the output, $\mathbf{e}$, cf. [1]:

$$\mathbf{G}_{ee} = \mathbf{H}\mathbf{G}_{ss}\mathbf{H}^{*} \tag{1}$$

Approximate budgeting and accurate analysis techniques and methods are needed to address each design phase with the right level of detail in a continuous flow from approximate to accurate. The lack of such approximate engineering techniques and methods leads to:

- the infeasibility of error budgeting in early design phases when detailed system information is not available

The lack of accurate error engineering techniques and methods leads to:

- either conservative designs (high margins, upper bound estimates) and thus overly strict hardware specifications, which lead to the selection of over-performing hardware and thus too high hardware costs,
- or high number of design iterations that involve comprehensive and thus costly analysis by different engineering disciplines,
- or infeasibility in case of high precision pointing missions.

In terms of accuracy it has been sufficient in past missions to perform analysis step AST-1, 3 and 4 in Figure 4. However, current missions have precision pointing requirements that are < 1 arcsec as stated in Table 1. In that case accurate modelling techniques and analysis methods are needed to identify design drivers and deal with uncertainties in the design of pointing systems like satellites. It has to be possible to apply these techniques throughout all mission design phases. In Figure 4 that would mean to more accurately model PES, $\mathbf{e}_s$, in AST-1 and to model the satellite system in AST-2. In this way the impact of the pointing system and individual sources can be analysed systematically and in more detail.

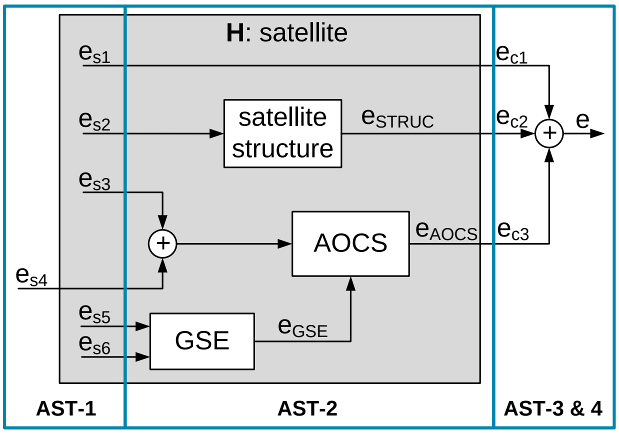


Figure 4: pointing error budgeting and analysis steps as in EPPE Handbook

SUMMARY

- *Approximate budgeting*
  - *case-by-case rules for PES modelling and statistical interpretation*
  - *error mapping guidelines: axis to LOS*
- *Accurate analysis*
  - *general rules for PES modelling and statistical interpretation*
  - *accurate error modelling: PSD, PDF, cross-correlation among sources and axes*
  - *accurate determination of PDF*
  - *MIMO LTI system transfer of stationary random processes*
  - *MIMO LTI system transfer of arbitrary PES, i.e. transients, drift, arbitrary distributed PES*
  - *Accurate pointing error index contribution*

C. ROBUSTNESS GUARANTEE

The need of giving robustness guarantees for the pointing budget is caused by the fact that design changes in an advanced state of the project produce high costs. These costs can be avoided if uncertainties in the pointing system are considered throughout the entire mission design lifecycle, including in early design phases. This can be done e.g. with robust control techniques in [13] for LTI pointing systems, **H**, with uncertain parameters, δ_i :

$$\mathbf{H}(\delta_1, \delta_2, \ldots, \delta_i) \tag{2}$$

The robustness of the pointing knowledge or performance can thus be analysed in terms of:

- **worst case pointing**: determine the maximum possible value of a pointing error index per axis, $e_{\text{index,axis}}$, after the transformation of the PES, $\mathbf{e}_S$, by the LTI pointing system, **H**, with uncertain but bounded parameters:

  $$\delta_i \in \left[\delta_{\text{i,min}} \quad \delta_{\text{i,max}}\right] \tag{3}$$

  ⇨ uncertain parameters are bounded, but their distribution is discarded
  ⇨ example for an uncertain parameter: orientation angles of a moving payload as in MetOp-SG, cf. [6]
- **pointing with certain robustness level of confidence**: determine worst case cumulative distribution function describing the likelihood of achieving a certain pointing error value, cf. [12]:

  $$p(\delta_i) \tag{4}$$

  ⇨ uncertain parameters are considered by their truncated (= value bounded) distribution and not only by their bounds
  ⇨ NOTE: In [12] PES still have to be Gaussian distributed.

As shown in [13], the worst case parameter combination is not necessarily the combination of all maxima in the interval of the uncertain parameter values because parameters can occur in arbitrary combinations. Currently margins are set based on engineering judgment to account for uncertainties in the pointing system. However, the engineering judgement in the allocation of margins can be misleading.

More effective and accurate methods with well-established techniques exist in [12] and [13]. These methods enable the analysis of robust stability and performance against uncertain parameters in LTI control systems.

They could also be used to give guarantees about the robustness of the pointing error, but for that purpose they need to be transferred from the AOCS domain to system level. With such robustness guarantees compliance with the mission's application requirement can be ensured although parameter values in the pointing system might arbitrarily change in the defined interval or PDF throughout the design phases. As mentioned before, ignoring such uncertainties might lead to mission infeasibility or costly design changes in later design phases.

SUMMARY

- *worst case pointing budget: system with uncertain but bounded parameters of any PDF*
- *pointing with certain robustness level of confidence: system with uncertain parameters of distinct and bounded PDF*

D. SENSITIVITY ANALYSIS

Especially in early design phases it is necessary to identify design drivers to perform trade-offs. Sensitivity analysis helps to identify the design drivers with respect to pointing knowledge and performance. Considering a pointing system the sensitivity can be computed with respect to:

- varying parameters in the pointing system (= varying transfer gain):

$$e_{index,axis} = \mathbf{H}(\delta_1, \delta_2, ..., \delta_i)\mathbf{e}_S \qquad \textbf{(5)}$$

- varying PES:

$$e_{index,axis} = \mathbf{H}\left(\mathbf{e}_S \pm \begin{bmatrix}\delta_1 & \delta_2 & \dots & \delta_i\end{bmatrix}^T\right) \qquad \textbf{(6)}$$

where $e_{index,axis}$ is the pointing error index per axis of interest. It is a result of the transfer of several PES, $\mathbf{e}_S$, acting on the pointing system, **H**. With (5) the relative impact on the final pointing error value is analysed if a system parameter (= varying parameter) changes. With (6) the impact on the final pointing error value is analysed if a PES changes its value (= varying parameter).

SUMMARY

- *relative impact of varying parameters in the pointing system*
- *relative impact of varying PES*

E. REQUIREMENTS ENGINEERING SUPPORT

Unlike the stated needs in section A to D, the requirements engineering support concerns the left part of Figure 2, the apportionment of required error values. This task defines the design constraints for the different subsystems, which makes it very important. It is a highly iterative and thus cumbersome process, which would benefit if there were guidelines for requirements engineering:

- to unambiguously specify pointing requirements
- to map, for typical mission scenarios, PES per axes in the body-frame to payload LOS PES
- to map application requirements (e.g. of a science instrument) to system requirements

Lessons learned show that discussions between customer and contractor about the pointing error budgeting approach (e.g. summation rules) could have been avoided in most cases if the requirement had been specified completely and unambiguously. Another cause for discussions is the application and reference frame mapping. However, in this case it is difficult to derive general guidelines because this task is mission specific.

Another need is to support the error apportionment itself by developing a systematic that guides the break-down of error requirements based on a set of constraints and objectives. Currently this is done based on experience and engineering practice.

SUMMARY

- *unambiguous specification of pointing requirements*
- *mapping of pointing requirements: application ↔ system, LOS ↔ axes*
- *systematic error requirement allocation support*

3. CURRENT EUROPEAN POINTING ERROR ENGINEERING FRAMEWORK

The current pointing error engineering framework in Europe is defined in the ESA Pointing Error Engineering (EPEE) Handbook and the ECSS standards and handbooks in the E-60 discipline of control engineering, which are available at [14]. An overview of current ECSS and ESA documents in the E-60 discipline is given in Figure 5. The EPEE Handbook is based on the ECSS standards and handbooks and complements those by providing practical guidelines and a step-by-step process. The ECSS-E-ST-60-10C [7] and ECSS-E-HB-60-10A [15] are the most relevant ones for pointing error engineering. The E-ST-60-20C [16] and the E-ST-60-21C [17] are relevant for describing PES inherent in a star sensor or gyro.

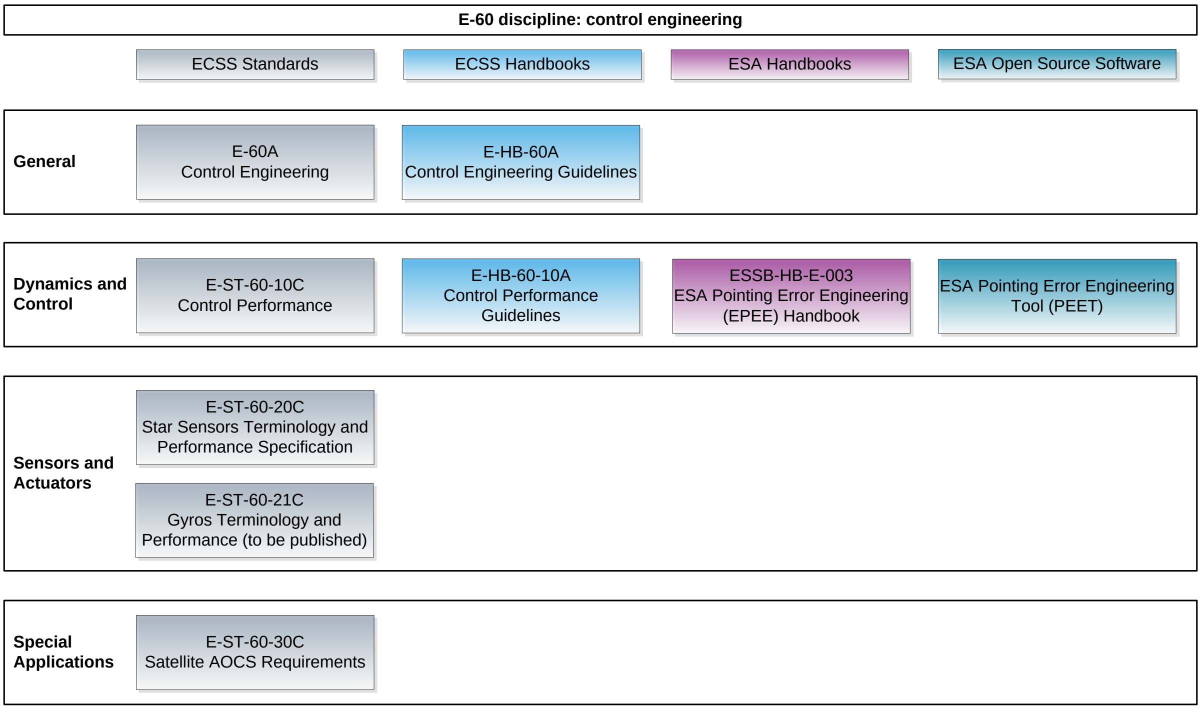


Figure 5: ECSS and ESA documents relevant for pointing error engineering

The ECSS documents provide an approximate pointing error engineering approach covering AST-1, 3 and 4 in Figure 4. But they do not provide an approach with proper level of accuracy for high accuracy pointing missions. In this case a more accurate approach is needed that also covers AST-2 and AST-1 in more detail. The EPEE Handbook addresses this need and provides accurate modelling techniques for describing PES with their frequency domain properties in AST-1. By modelling and analysing the frequency domain properties an exact error index contribution can be determined. As introduced in [1] these techniques are based on various publications that trace back to the initial paper of [18]. The ESA pointing error engineering tool (PEET) in [3] has been developed to support the application of these techniques. However, the main purpose of PEET is to guide and support the pointing budgeting and analysis in general by being conform to all ECSS and ESA documents. Eventually it has to be noted that the ECSS and ESA documents in Figure 5 support and guide the pointing budgeting and analysis, but not the apportionment in Figure 2 on the left.

An overview on the evolution of the pointing error engineering framework is given in Figure 6. The "ECSS" column on the left lists the main features of the pointing error engineering framework defined by ECSS documents only. The column "ECSS & EPEE" in the middle lists all features of the framework defined by the ECSS documents and the EPEE handbook. The column on the right "PEET (ECSS & EPEE)" lists all features of the framework defined and supported by the ECSS documents, the EPEE handbook and the PEET software. The evolution of framework features is indicated in different colours. As indicated by the arrow the level of detail and accuracy for analysis increases from the left to the right, which at the same time represents the chronological development.

Before the release of the EPEE Handbook in the year 2011, the ECSS-E-ST-60-10C and ECSS-E-HB-60-10 were generally required in projects at Airbus Defence and Space. After the year 2011 the EPEE Handbook has been applied in new projects and studies, like MetOp-SG, Euclid, EDRS, MTG and LOFT.

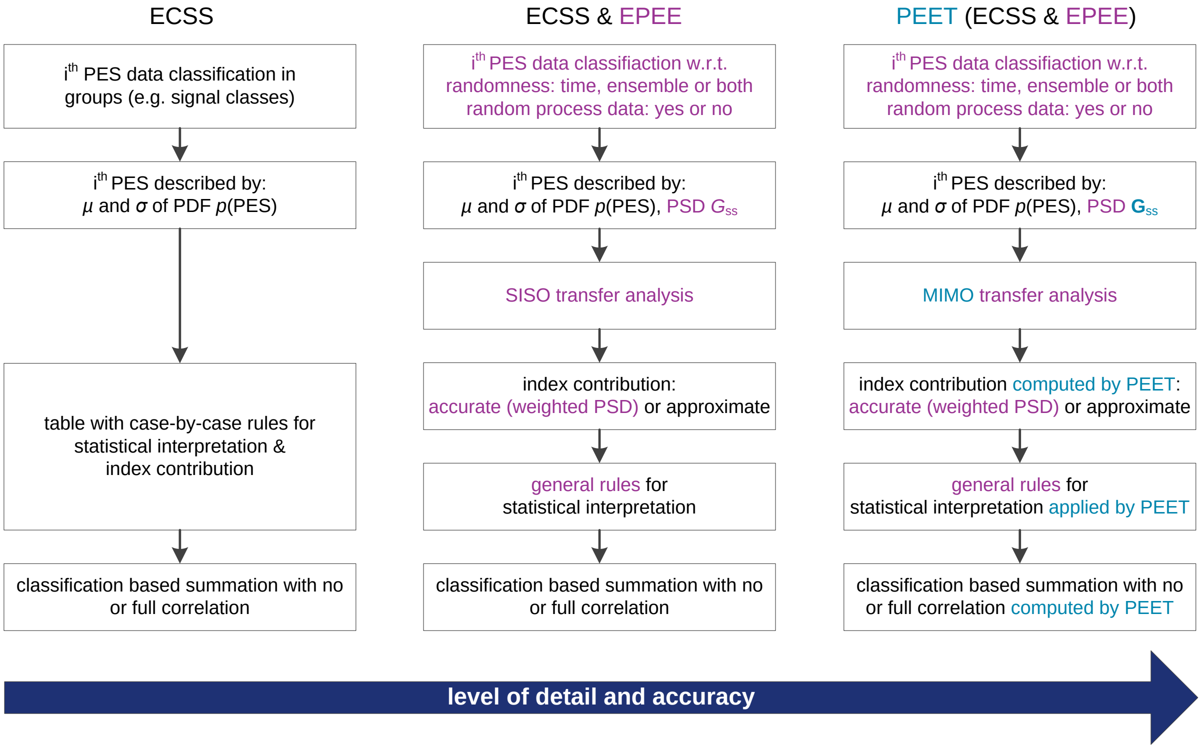


Figure 6: evolution of the pointing error engineering framework

4. EPEE HANDBOOK AND PEET: A STEP-BY-STEP BENCHMARK

The aim of the benchmark is to analyse if the challenges and needs, stated in chapter 1 and 2, are addressed by the current framework that is defined by the ECSS and ESA documents in Figure 5. In particular, the benefits and limitations in applying the new developments, namely the EPEE Handbook and PEET software, shall be assessed.

A. BENCHMARK SCENARIOS

Two different benchmark scenarios for pointing budgeting are set up to assess the applicability of the following three pointing error engineering approaches:

1) ECSS standard with classical summation rules
2) EPEE Handbook with ECSS
3) PEET software and thus implicitly EPEE Handbook with ECSS

The approaches emerge from the chronological development steps in Figure 6. In this paper each approach represents a development status of the pointing error engineering framework at a certain point in time. Until 2011 only ECSS documents were available. Approach 1 represents this development status. It corresponds to the left column in Figure 6. After 2011 the EPEE Handbook has been available in addition. Approach 2 represents this development status. It corresponds to the middle column in Figure 6. After 2013 the PEET software has been available as a framework complement. Approach 3 represents this development status. It corresponds to the right column in Figure 6.

The first benchmark scenario is based on the MetOp-SG mission. The scenario is called "Earth Observation LEO Satellite" and its pointing error requirements (PER) are defined in Table 2. It is analysed with the simplified statistical method, which is based on the specific summation of random variables. In addition, the ACS of a Euclid like high precision mission in fine pointing science observation serves as another benchmark scenario to make use of the advanced frequency-domain methods in the EPEE Handbook and PEET. These methods are based on the summation of random processes and their system transfer, cf. [1]. This scenario is called "ACS of Precision Pointing Satellite".

The benefits for each approach are given separately to illustrate the evolution of developments in pointing error engineering. The limitations are given with respect to the current status, i.e. approach 3, that includes the ECSS standards, the EPEE handbook and the PEET software.

Table 2: pointing error requirements of benchmark scenarios

pointing error requirement (PER)	Earth Observation LEO Satellite			ACS of Precision Pointing Satellite		
evaluation period	observation period of adjacent field-steps until re-orientation			science observation of one image frame		
error index	APE			RPE		
window-time	-			100s		
required error value	x	y	z	x	y	z
	2	2	2	40	40	250
unit	mrad			mas		
level of confidence P_c	99.7%			68.3%		
statistical interpretation	temporal			temporal		
error reference	per axis of payload reference frame			per axis of payload reference frame		

The values of the requirements and PES had to be adapted for this benchmark to not disclose intellectual property. However, that does not have any impact on the quality of results and conclusions that can be drawn.

B. EARTH OBSERVATION LEO SATELLITE

In this scenario the PES are equal to the PEC because the ECSS standard does not provide any guidelines for the analysis step AST-2 in Figure 4. The PEC with their values are given in Table 3. The last column in the table called "type" has the following categories:

- **Bias - B**: PEC that are constant with respect to time
- **Drift - D**: PEC that drift with time
- **Periodic - P**: PEC that change periodically with time
- **Correlated Periodic - P_c**: PEC that change periodically with time and that can be in phase such that their peaks occur at the same time, e.g. periodic PEC with the same frequency
- **Random - R**: PEC that change randomly over time

The in Table 3 highlighted PEC 3, 7-10 and 21 will be considered in detail in the following.

Table 3: LEO satellite - pointing error contributors

	e_{PEC} (=e_{PES})			$\mu_B(e_{PEC})$	$\sigma_{[Group]}(e_{PEC})$	Type
nr.				in µrad		
1	AOCS	Star trackers	Internal alignment	30.0		B
2			Internal alignment drift		9.0	D
3			Thermal distortion		11.3	P_c
4			Random		4.0	R
5		GPS Receiver	Position error		1.0	R
6			Velocity error		1.3	R
7		Attitude control	Bias	90.0		B
8			Orbital		14.1	P_c
9			Random		30.0	R
10	Dynamics	Instruments scan			12.0	P
11		Platform mechanisms			4.9	P
12		Microvibrations			3.0	R
13	Structure	Alignments	Instrument alignment	300.0		B
14			Desorption	12.0		B
15			Gravity release	45.0		B
16			Setting star tracker	90.0		B
17			Setting instrument	99.0		B
18		Thermo-elastic	Constant	66.0		B
19			Orbital		4.2	P_c
20			Season		8.0	D
21			Ageing		18.0	D

APPROACH 1: ECSS STANDARD WITH CLASSICAL SUMMATION RULES

The pointing error budget is compiled in line with [7] using classical summation rules as applied for the MetOp-SG in early design phases in Airbus Defence and Space. The summation rules are listed in Table 4. The final pointing error is 900 µrad.

Table 4: LEO satellite - classical summation rules as used for MetOp-SG

PEC groups			
	not cross-correlated	correlated	error in µrad
Bias	$B = \sum \mu_B$	N.A.	732
Periodic	$P = \sum \sigma_P^2 = \sum A^2/2$	$P_c \le \left(\sum \sigma_{Pc}\right)^2 = \left(\sum A/\sqrt{2}\right)^2$	35
Gaussian Noise	$G = \sum \sigma_G^2$	N.A.	30
Drift	$D = \left(\sum \sigma_D\right)^2$	N.A.	13
Total	$e_{APE} \le B + n_p\sqrt{P + P_c + G + D}$		**900**

APPROACH 2: EPEE HANDBOOK WITH ECSS

The pointing error budget is compiled in line with the EPEE Handbook [1]. The computations are implemented in Microsoft® Excel. According to the EPEE Handbook the PEC (=PES) are categorized in the first analysis step AST-1. The categorization "types" correspond to the ones in section B of this chapter and are in line with the EPEE Handbook as well as the nomenclature and definitions used in this section. The categorization of ensemble-random PEC is shown in Table 5.

Table 5: LEO satellite - categorization of ensemble-random PEC

ensemble-random (= time-constant) PES					
nr.	signal class	description	random parameter	distribution	type
7	bias	variable	bias magnitude	uniform	B

The categorization of time-random PEC is shown in Table 6. The frequency of the Gaussian error signals defines the corner frequency of band-limited white noise.

Table 6: LEO satellite - categorization of time-random PEC

time-random PES								
nr.	signal class	description	random parameter	distribution	frequency [rad/s]			type
					x	y	z	
3	periodic	variable	distortion magnitude	bimodal	1.0E-03	1.0E-03	1.0E-03	P_c
8	periodic	variable	mag. of dominant mode	bimodal	1.0E-03	1.0E-03	1.0E-03	P_c
9	random	variable	noise magnitude	Gaussian	8.0E-02	8.0E-02	8.0E-02	G
10	periodic	variable	mag. of dominant mode	bimodal	4.7E+00	4.7E+00	4.7E+00	P
21	drift	variable	distortion magnitude	uniform	-	-	-	D

The description, index contribution and summation of the ensemble-random PEC indicated in Table 3 are given in Table 7. The equivalent statistical properties are the statistical properties of a PES after statistical interpretation with respect to the PER in Table 2.

Table 7: LEO satellite - description, index contribution and summation of ensemble-random PEC

	ensemble-random			equivalent statistical properties				
PES	**PDF (*1)**	**PDF parameters**		**μ**		**σ**		**unit**
7	$U(e_{min},e_{max})$	e_{min}	-90.0	$\max(\lvert e_{min}\rvert,\lvert e_{max}\rvert)$	90.0	-	0.0	
		e_{max}	90.0					
						$\sqrt{\sum\sigma_{Bnc}^2}$	**0.0**	
						$\sum\sigma_{Bc}$	**0.0**	
RMS SUM				μ_B	**732.0**	σ_B	**0.0**	

The description, index contribution and summation of the time-random PEC highlighted in Table 3 are given in Table 8. The table includes a column for time-random and one for ensemble-random properties because the time-random PEC statistical properties can also be ensemble-random.

Table 8: LEO satellite - description, index contribution and summation of time-random PEC

	Time-Random				Ensemble-Random				Equivalent Statistical Properties	
Nr	**PDF**	**PDF parameters**		**PDF**	**PDF parameters**		**μ in μrad**		**σ in μrad**	
3	BM(A)	e_{min}	-16.0	δ(A)	$\mu_\delta = A$	16.0	$(e_{max}+e_{min})/2$	0.0	$\sqrt{\int\sigma^2(e\vert A)\delta(A)dA} = A/\sqrt{2}$	11.3
		e_{max}	16.0		$\sigma_\delta = 0$	0.0				
		$A=(e_{max}-e_{min})/2$	16.0							
8	BM(A)	e_{min}	-20.0	δ(A)	$\mu_\delta = A$	20.0	$(e_{max}+e_{min})/2$	0.0	$\sqrt{\int\sigma^2(e\vert A)\delta(A)dA} = A/\sqrt{2}$	14.1
		e_{max}	20.0		$\sigma_\delta = 0$	0.0				
		$A=(e_{max}-e_{min})/2$	20.0							
9	$G(\mu_G,\sigma_G)$	μ_G	0.0	$\delta(\sigma_G)$	$\mu_\delta=\sigma_G$	90.0	μ_G	0.0	$\sqrt{\int\sigma^2(e\vert\sigma_G)\delta(\sigma_G)d\sigma_G} = \sigma_G$	30.0
		σ_G	30.0		$\sigma_\delta=0$	0.0				
10	BM(A)	e_{min}	-17.0	δ(A)	$\mu_\delta = A$	17.0	$(e_{max}+e_{min})/2$	0.0	$\sqrt{\int\sigma^2(e\vert A)\delta(A)dA} = A/\sqrt{2}$	12.0
		e_{max}	17.0		$\sigma_\delta = 0$	0.0				
		$A=(e_{max}-e_{min})/2$	17.0							
21	$U(e_{min},e_{max})$	e_{min}	0.0	δ(C)	$\mu_\delta = C$	18.0	$\max(\lvert e_{min}\rvert,\lvert e_{max}\rvert)$	18.0	-	0.0
		$e_{max}=C$	18.0		$\sigma_\delta = 0$	0.0				
		C	18.0							
							$\sum\mu_G$	**0.0**	$\sqrt{\sum\sigma_G^2}$	**30.5**
							$\sum\mu_D$	**17.0**	$\sqrt{\sum\sigma_D^2}$	**0.0**
							$\sum\mu_P$	**0.0**	$\sqrt{\sum\sigma_P^2}$	**13.0**
									$\sum\sigma_{Pc}$	**29.7**
SUM							μ_{APE}	**35.0**	σ_{APE}	**44.5**

The final summation and evaluation of the intermediate results in Table 7 and Table 8 is performed in Table 9 in direct comparison to the results of approach 3.

APPROACH 3: PEET SOFTWARE AND THUS IMPLICITLY EPEE HANDBOOK WITH ECSS

The pointing budget is set up in the PEET software with the same inputs, i.e. categorization and description, as in approach 2. The budget model as displayed in PEET is shown in Figure 7. The user interface looks and works similar to the one of MATLAB® Simulink. The results computed by PEET are listed in Table 9 in direct comparison to the error values computed with the other approaches.

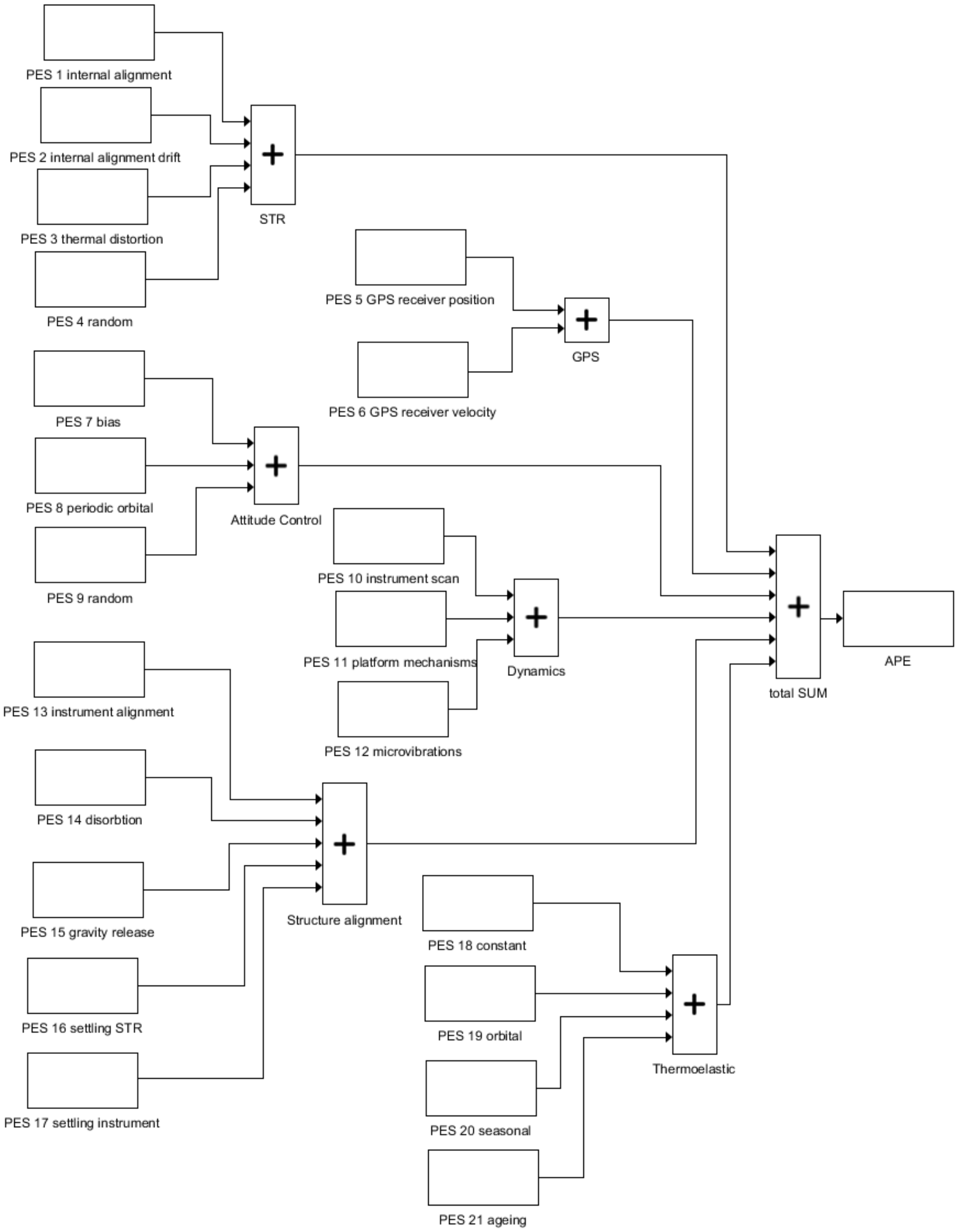


Figure 7: LEO satellite - PEET model

EVALUATION OF OVERALL POINTING ERROR FOR APPROACH 1-3

The final pointing error as computed with the different approaches is given in Table 9. The approach based on the ECSS standard with classical summation rules (approach 1) produces the same results as the approach based on the EPEE Handbook (approach 2). The value computed by PEET (approach 3) is different for the time-random contribution, σ_{APE}. This deviation was analysed together with the software producer Astos Solutions. It occurs due to a bug in the software that results in a wrong computation of correlated periodic signals, the P_c type. Meanwhile the bug has been corrected for future software releases.

Table 9: LEO satellite - evaluation of overall pointing error with approach 1-3

Error Sums	ECSS	EPEE	PEET	unit
μ_B	-	732.00	732.00	µrad
σ_B	-	0.00	0.00	
B	-	**732.00**	**732.00**	
μ_{APE}	-	35.00	35.00	
σ_{APE}	-	44.48	30.33	
$\boldsymbol{\varepsilon_{APE}}$	**-**	**168.45**	**126.00**	
$e_{APE} = B + \varepsilon_{APE}$	**900.45**	**900.45**	**858.00**	
n_p	3.00	3.00	3.00	
e_{APE}	**0.90**	**0.90**	**0.86**	mrad
$e_{APE,r}$	2.00			

In the following it is shown that the results of approach 1 and 2 have to correspond because the pointing budget computation is equal although not obvious from the non-rearranged equations. The correspondence becomes obvious by rearranging the equations in the EPEE Handbook for the evaluation of the APE such that:

$$\begin{aligned}
e_{APE} &\le n_p\sigma_B + |\mu_B| + n_p\sigma_{APE} + |\mu_{APE}| \\
&\le \underbrace{n_p\sqrt{\underbrace{\sum\sigma_{Bnc}^2}_{=0} + \left(\underbrace{\sum\sigma_{Bc}}_{=0}\right)^2}}_{\sigma_B} + \left|\sum\mu_B\right| + \underbrace{n_p\sqrt{\sum_{i=1}^{N}\left(\sum\sigma_{APEnc}^2 + \left(\sum\sigma_{APEc}\right)^2\right)_i}}_{\sigma_{APE}} + \left|\underbrace{\sum\mu_{APE}}_{=0}\right| \\
&\le \left|\sum\mu_B\right| + n_p\sqrt{\sum_{i=1}^{N}\left(\sum\sigma_{APEnc}^2 + \left(\sum\sigma_{APEc}\right)^2\right)_i} \\
&\le \left|\sum\mu_B\right| + n_p\sqrt{\sum\sigma_G^2 + \sum\sigma_P^2 + \left(\sum\sigma_{Pc}\right)^2 + \left(\sum\sigma_D\right)^2} \\
&\le B + n_p\sqrt{P + P_c + G + D}
\end{aligned} \tag{7}$$

Now it can be seen that the APE formula in the EPEE Handbook is equal to the formulas used for MetOp-SG in Table 4.

C. ACS OF PRECISION POINTING SATELLITE

The benchmark scenario for the application of the advanced frequency-domain methods in the EPEE Handbook and PEET is a precision pointing telescope in fine pointing mode for science observation. The pointing performance of a typical ACS of such a satellite is analysed hereafter. The PES considered have properties as they are encountered in such missions. The objective of this benchmark is to analyse the three approaches with respect to each other by focusing on the accurate modelling techniques and analysis methods in the EPEE handbook and PEET. Consequently approach 1 is not considered because it does not provide any framework to perform high accuracy budgeting. In approach 2 the computations are performed directly in MATLAB® with well proven scripts developed by Airbus Defence and Space. In approach 3 the computations are performed by PEET, which also uses MATLAB® in the background, cf. [3].

The ACS considered for analysis is shown in Figure 8. The inputs, outputs and dynamic systems are given in Table 10.

Table 10: precision pointing satellite - system matrices and signals

System matrices		Input and output signal vectors	
K:	controller dynamics	**r**:	reference
$\mathbf{G}_{ad}$:	actuator dynamics	**e**:	pointing error
$\mathbf{G}_{p}$:	pointing system plant dynamics	**n**:	attitude noise
$\mathbf{G}_{sd}$:	sensor/estimator dynamics	**d**:	actuation noise
$\mathbf{C}_{A/B}$:	actuation matrix (actuator to body frame)		
$\mathbf{C}_{S/B}$:	configuration matrix (sensor to body frame)		

With the dynamic systems being 3 x 3 matrices except for $\mathbf{C}_{A/B}$ and $\mathbf{C}_{S/B}$, which are 3 x n with n being the number of actuators and sensors respectively. The inputs and outputs are 3 x 1. For analysis the signals **n** and **d** are represented by their PSD $\mathbf{G}_{nn}$ and $\mathbf{G}_{dd}$.

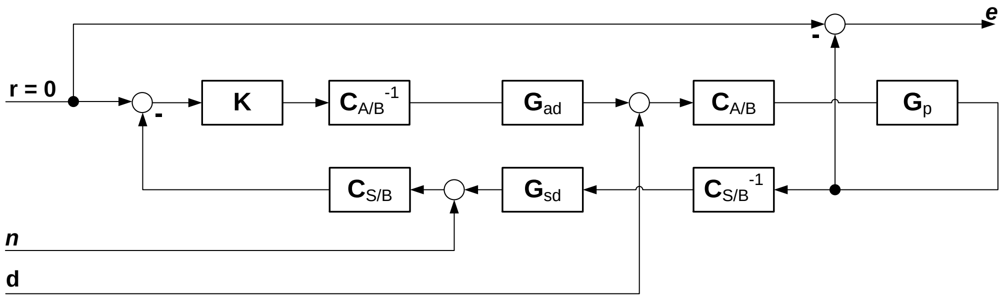
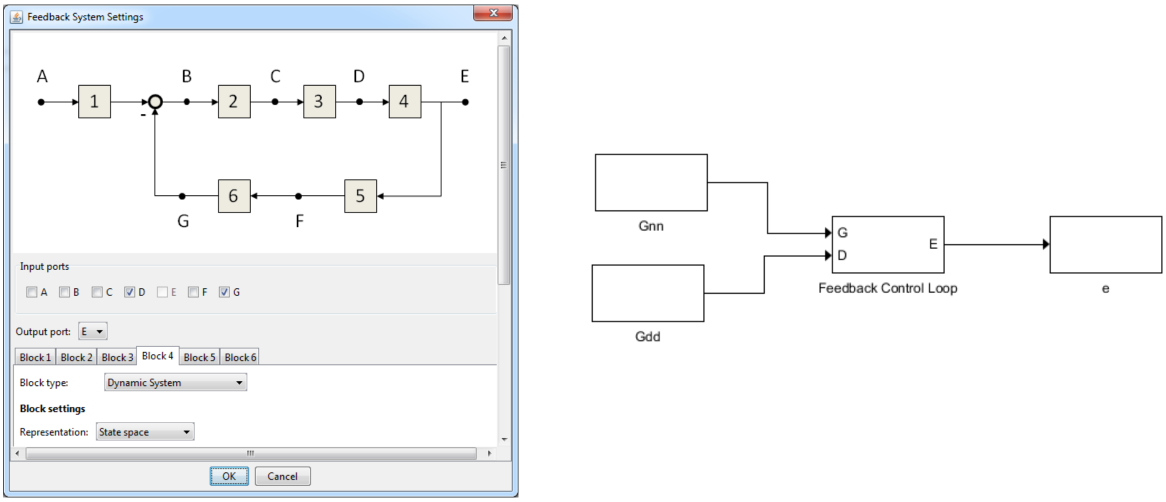


Figure 8: precision pointing satellite - ACS system

In PEET such a system can be represented by the "feedback system block" in Figure 9 on the left. As can be seen in the figure there are only 6 system blocks available in PEET. However, the system blocks can be combined because only two inputs and one output are needed to represent the system in Figure 8. The final PEET model representing the ACS in Figure 8 is shown in Figure 9 on the right. The parameterization of the blocks is simple and flexible because each system block can be linked to the workspace of MATLAB®.



Input ports ☐ A ☐ B ☐ C ☑ D ☐ E ☐ F ☑ G

Output port: E

Block type: Dynamic System

Block settings

Representation: State space

Figure 9: precision pointing satellite - PEET model

ATTITUDE KNOWLEDGE AND DISTURBANCE NOISE

The measurement noise is the attitude estimation error of a Gyro-Stellar-Estimator (GSE), which fuses FGS and IMU measurements. It is most accuratly described by its PSD because this type of noise is Gaussian distributed and time-correlated. The single-sided attitude knowledge error √PSD considered for the benchmark scenario is plotted in Figure 10. The figure only contains the diagonal entries of $\mathbf{G}_{nn}$ because the measurement noise is considered to be not cross-correlated.

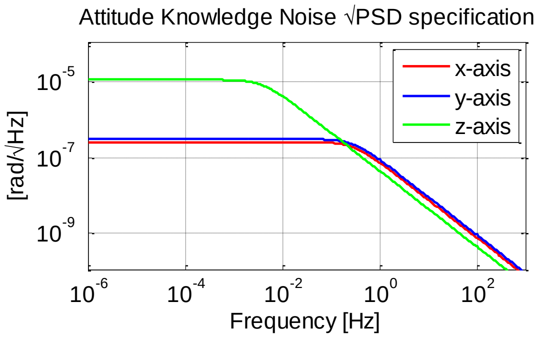


Figure 10: precision pointing satellite - measurement noise

The disturbance noise is a combination of the reaction wheel torque noise and solar pressure noise. As for the measurement noise the disturbance noise is also modelled by its PSD, $\mathbf{G}_{dd}$. The single-sided disturbance noise √PSD is plotted in Figure 11.

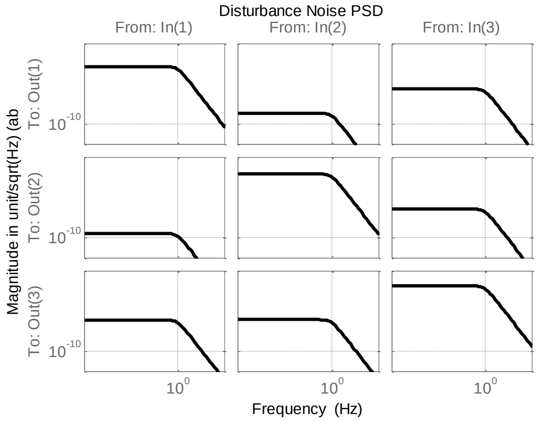


Figure 11: precision pointing satellite - disturbance noise

CLOSED-LOOP TRANSFER FUNCTIONS

The diagonal entries of the closed-loop transfer functions for all axes are plotted in Figure 12. The off-diagonal entries of the closed transfer function matrix are omitted.

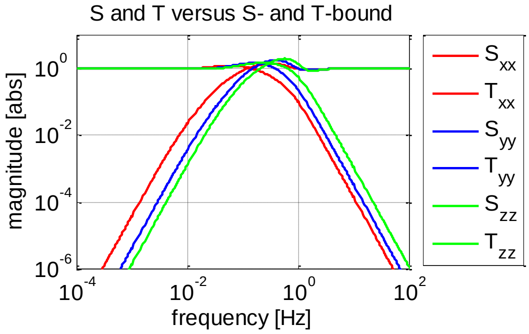


Figure 12: precision pointing satellite - closed-loop tranfer functions

G_{ee} of e_{RPE}: PEET VERSUS MATLAB®

The resulting pointing error PSD, $\mathbf{G}_{ee}$, after transformation of $\mathbf{G}_{nn}$ and $\mathbf{G}_{dd}$ by the closed-loop dynamics is plotted as √PSD in Figure 13. The pointing error √PSD obtained by computing the budget according to the approach in the EPEE Handbook is plotted on the left. The plot on the right represents the pointing error √PSD computed by PEET. The results are identical.

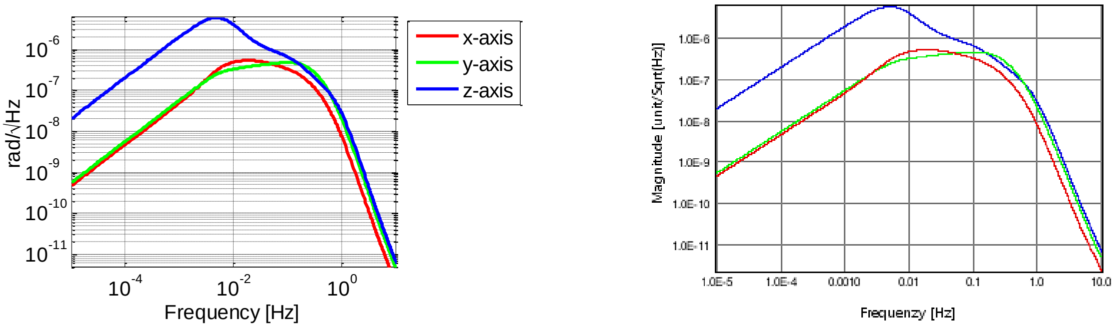


Figure 13: precision pointing satellite - pointing error √PSD of PEET versus MATLAB®

EVALUATION OF OVERALL POINTING ERROR

The evaluation of the pointing error per axis is computed according to Eq.(1) with $\mathbf{G}_{ss} = \mathrm{diag}(\mathbf{G}_{nn}, \mathbf{G}_{dd})$ and finally in line with [1] such that:

$$e_{index,axis} = \sqrt{\int_0^\infty G_{ee,axis}(f)F_{metric}(f)df} \qquad \textbf{(8)}$$

where $F_{metric}(f)$ is the metric weighting filter that determines the error index contribution of the pointing error PSD, $G_{ee,axis}(f)$, of the selected axis. The results computed by MATLAB® and the results computed by PEET are listed in Table 11. As expected the computed overall pointing error, $\mathbf{e}_{RPE}$, is equal in approach 2 and 3 because their PSD are also equal. The difference between the results is < 0.1 mas. Approach 1 has not been analysed as stated before.

Table 11: precision pointing satellite - evaluation overall pointing error with approach 1-3

Error Sums	ECSS	MATLAB®(PEEH)			PEET			unit
		x-axis	y-axis	z-axis	x-axis	y-axis	z-axis	
μ_B	-	0.00	0.00	0.00	0.00	0.00	0.00	mas
σ_B	-	0.00	0.00	0.00	0.00	0.00	0.00	
B	**-**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	
μ_{RPE}	-	0.00	0.00	0.00	0.00	0.00	0.00	
σ_{RPE}	-	35.5	50.8	134.1	35.5	50.7	134.1	
ε_{RPE}	**-**	**35.5**	**50.8**	**134.1**	**35.5**	**50.7**	**134.1**	
$e_{RPE} = B + \varepsilon_{RPE}$	**-**	35.5	50.8	134.1	35.5	50.7	134.1	
n_p	-	1.00	1.00	1.00	1.00	1.00	1.00	
e_{RPE}	**-**	**35.5**	**50.8**	**134.1**	**35.5**	**50.7**	**134.1**	mas
$e_{RPE,r}$	40.0	40.0	40.0	250.0	40.0	40.0	250.0	

5. BENCHMARK EVALUATION

The objective of the benchmark is to analyse the three pointing error engineering approaches defined in chapter 4 with respect to each other. In order to cover all analysis criteria two benchmark scenarios have been analysed. One scenario had the focus on the simplified statistical method, which is based on the specific summation of random variables. The other scenario had the focus on the accurate modelling techniques and analysis methods provided by the EPEE handbook and PEET. In this chapter the benchmark results are analysed and conclusions are drawn with respect to the needs stated in chapter 2. The evaluation answers the following questions:

- Which needs does the current pointing error engineering framework address?
- Which approach in the framework addresses a need and to which extend?
- Where is potential for improvement in the current framework?

This chapter is organized in such that the categories of needs A to E in chapter 2 are evaluated in a matrix with the three different pointing error engineering approaches. A need is marked in green with "Y" if it is fully addressed by one approach. It is marked in orange with "P" if it is partially addressed and it is marked in red with "N" if it is not addressed at all. In case a need is partially addressed it will be made clear to which extend.

A. OPTIMAL ENGINEERING PROCESS

The **benefit** in **approach 1** "ECSS standard with classical summation rules" is that it defines hands-on rules for the statistical interpretation of PEC and their error index contribution in look-up tables in [7]. These tables enable fast budgeting with the "simplified statistical method", which is defined in [1] and [7].

The main benefit in **approach 2** " EPEE Handbook with ECSS" is that it defines an integrated process as requested in the needs of chapter 2. The process embeds the mathematical elements in the ECSS documents listed in chapter 3. Changes in the pointing budget can thus be implemented responsively. The user is guided through the process step-by-step, i.e.: how to describe PES, when to do statistical interpretation, when to sum which contributors and others. A tailoring of the process complexity is possible due to a step-wise structure that allows skipping steps. This and the fact that the EPEE handbook provides computational methods with different levels of detail, accuracy and complexity provide the conditions to achieve a balanced design solution in terms of engineering level of detail versus associated costs.

The benefits in **approach 3** "PEET software and thus implicitly EPEE Handbook with ECSS" are the same as for approach 2. However, the process is even more responsive because the PEET software implements the formulas and rules to automatically perform analysis steps AST-2 to AST-4 of the pointing error budgeting and analysis process in the EPEE handbook.

The **limitations** of the current pointing error engineering framework, i.e. approach 3, are the current development status of the PEET software. PEET has not yet undergone a solid software verification and validation campaign and thus currently requires cross-checking of the results. Several minor software implementation bugs have been encountered in the prototype PEET V0.3. However, once the bugs were corrected the software produced the desired results. A bug that concerns cross-correlated periodic error signals is illustrated in section B of chapter 4. It was corrected afterwards and will be included in a new software prototype release. Another limitation is that the application of the analysis methodology and techniques in the EPEE Handbook, especially the tailoring of the process for optimally, requires practice and experience. However, the time to learn and apply the techniques and methods of the current framework is a one-time effort that will save many discussions compared to if it were done otherwise.

The **summary** on which needs of chapter 2 are addressed by the different approaches is given in Figure 14.

Figure 14: optimal engineering process - addressed needs

needs	ECSS classical	PEEH	PEET
tailorable process to right level of detail and accuracy for respective design phase and mission type	N	Y	Y
techniques and methods responsively useable on system level	Y	P	Y
integrated process based on unified methodology with:			
exact mathematical elements	Y	Y	Y
practical guidelines	P	Y	Y
systematic and multidisciplinary design flow	N	Y	Y
standardized and coherent interfaces	N	Y	Y
continuous (thus hybrid) flow from approximate to accurate budgeting techniques	N	Y	Y

B. APPROXIMATE BUDGETING AND ACCURATE ANALYSIS

The **benefit** in **approach 1** "ECSS standard with classical summation rules" is that it defines approximate modelling techniques and analysis methods for determining pointing error index contribution based on statistical properties (mean, variance). Ref. [7] provides:

- exact mathematical elements with standardized nomenclature
- explicit rules in look-up tables for the approximate contribution of a PEC to an error index for the most common budgeting cases
- summation rules approximating cross-correlation with an upper and lower bound
- evaluation of level of confidence under the assumption of the central limit theorem

A benefit in **approach 2** " EPEE Handbook with ECSS" is that the EPEE Handbook gives, in addition to the look-up tables in [7], general formulas and guidelines for the categorization of PES, system transfer of PES, statistical interpretation of PEC, and exact contribution of a PEC to an error index. The main benefit of this approach is that it thus provides accurate modelling techniques and analysis methods, which can be used in a hybrid manner with the approximate ones, for PES system transformation and PEC error index contribution based on PSD analysis. The EPEE handbook also gives guidelines for applying norms on system transfer matrices for design purposes, e.g. controller design.

The benefits in **approach 3** "PEET software and thus implicitly EPEE Handbook with ECSS" are the same as for approach 2. However, the process is even more responsive because the PEET software implements the formulas and rules to automatically perform analysis steps AST-2 to AST-4 of the pointing error budgeting and analysis process in the EPEE handbook with different:

- level of accuracies and process complexity
- PES model representations: PDF with statistical properties (variance, mean), covariance matrix, PSD as magnitude values on a frequency grid, PSD defined by different transfer function representations available in the MTLAB® control toolbox

- system model representations as they are available in the MATLAB® control toolbox

This enables fast assessment of different pointing error indices, system configurations and statistical interpretations.

The **limitation** of the current pointing error engineering framework, i.e. approach 3, is imposed by signal and system modelling constraints. Currently the EPEE Handbook guidelines and PEET software only support the modelling of LTI systems and stationary random process signals. Other systems and in particular arbitrary PES signals (i.e. transients, drift, arbitrary distributed PES) are very common and thus also require to be modelled accurately. The EPEE Handbook and PEET only give approximate modelling guidelines in that case. Also cross-correlation of PES and among axes needs to be computed accurately. Currently it can only be distinguished between full or not cross-correlated. Another limitation is that there are no guidelines for the evaluation of the level of confidence if the central limit theorem does not apply. If there are dominant non-Gaussian PES, an exact determination of the pointing error PDF is indispensable for precision pointing missions. At the end the analysis of course depends on the availability of data (PSD shape, cross-correlation, distribution, etc.) and the accuracy of the models. Otherwise the best modelling techniques and analysis methods are not effective.

The **summary** on which needs of chapter 2 are addressed by the different approaches is given in Figure 15.

Figure 15: approximate budgeting and accurate analysis - addressed needs

needs	ECSS classical	PEEH	PEET
Approximate budgeting			
case-by-case rules for PES modelling and statistical interpretation	Y	Y	Y
error mapping guidelines: axis to LOS	P	P	P
Accurate analysis			
general rules for PES modelling and statistical interpretation	N	Y	Y
MIMO LTI system transfer of stationary random processes	N	Y	Y
MIMO LTI system transfer of arbitrary PES, i.e. transients, drift, arbitrary distributed PES	N	P	P
accurate error modelling with:			
PSD	N	Y	Y
PDF	Y	Y	Y
cross-correlation among PES and axes	P	P	P
accurate determination of PDF	N	N	N

A. ROBUSTNESS GUARANTEE

All approaches do not address the need of having a robustness guarantee for the pointing budget. However, **approach 2** " EPEE Handbook with ECSS" has the **benefit** that the accurate modelling techniques and analysis methods in the EPEE Handbook are already compatible with the robust control analysis framework. How to explicitly address the pointing error indices with these methods is shown in [19].

Approach 3 "PEET software and thus implicitly EPEE Handbook with ECSS" has the same benefits as approach 2. In addition it is of advantage that the PEET computation engine is in MATLAB®. The pointing system is thus directly available in the MATLAB® workspace as well as the PES parameters. This allows the direct application of the tools in the MATLAB® robust control toolbox for robustness analysis.

The **limitations** of the current pointing error engineering framework, i.e. approach 3, are that there are currently no guidelines in the EPEE Handbook to compile robust pointing budgets. A research and development effort is necessary to fulfil the stated needs and see if existing MATLAB® toolboxes like the RC toolbox can indeed be applied without modifications. The same applies for the PEET software. It is currently not possible to compile a robust pointing budget with PEET. Hence research and development is necessary to extended PEET such that uncertain systems can be modelled and robustness analyses can be run with existing and well proven toolboxes.

The **summary** on which needs of chapter 2 are addressed by the different approaches is given in Figure 16.

Figure 16: robustness guarantee - addressed needs

needs	ECSS classical	PEEH	PEET
worst case pointing budget: system with uncertain but bounded parameters of any PDF	N	N	N
pointing with certain robustness level of confidence: system with uncertain parameters of distinct and bounded PDF	N	N	N

B. SENSITIVITY ANALYSIS

All approaches do not address the need of performing sensitivity analyses except for the PEET software in **approach 3**. As explained in [20], under the condition that a PES is modelled by its statistical properties PEET can determine the sensitivity of a PES with respect to the overall pointing error by adding small perturbations such that the perturbed pointing error is:

$$\tilde{e}_S = \begin{cases} PES + \Delta & \forall \left|PES\right| \le 1 \\ PES(1+\Delta) & \forall \left|PES\right| > 1 \end{cases} \tag{9}$$

The sensitivity is then computed by:

$$s = \frac{\tilde{e}_{index,axis} - e_{index,axis}}{\tilde{e}_S - e_S} \tag{10}$$

The **limitations** of the current pointing error engineering framework, i.e. approach 3, are that there are no guidelines in the EPEE Handbook to perform sensitivity analyses. PEET does currently not support to analyse the impact of varying parameters in the pointing system on the overall pointing error.

The **summary** on which needs of chapter 2 are addressed by the different approaches is given in Figure 17.

Figure 17: sensitivity analysis - addressed needs

needs	ECSS classical	PEEH	PEET
relative impact of varying parameters in the pointing system	N	N	N
relative impact of varying PES	N	N	Y

C. REQUIREMENTS ENGINEERING SUPPORT

All approaches do not address the need of giving requirements engineering support except for the EPEE Handbook in approach 2 and 3 that defines a list of parameters and information for unambiguously specifying pointing error requirements. This list of parameters and information is also requested in PEET for requirements specification.

The **limitation** of the current pointing error engineering framework, i.e. approach 3, is that there is the development need of providing guidelines in the EPEE Handbook to map application to system requirements (application mapping) and to map PES per axes in body-frame to payload LOS errors (frame mapping) for typical mission scenarios. However, the main need is to have a tool supporting the systematic error requirements apportionment.

The **summary** on which needs of chapter 2 are addressed by the different approaches is given in Figure 18.

Figure 18: requirements engineering support - addressed needs

needs	ECSS classical	PEEH	PEET
unambiguous specification of requirements	N	Y	Y
mapping of requirements: application ↔ system, LOS ↔ axes	N	N	N
systematic apportionment (error requirement allocation) support	N	N	N

6. CONCLUSIONS

Based on the experience in Airbus Defence and Space and as illustrated in the benchmark of this paper it can be concluded that the EPEE Handbook improves pointing error engineering. This is especially the case because standard nomenclature and a unified methodology for pointing error budgeting harmonize the design process at the interfaces between ESA, the prime-contractor and sub-contractors. Although the methodology and the advanced high precision methods in the Handbook seem to introduce more complexity at first sight, in the long-term the S/C system design process will be simplified and more accurate. In particular, the availability of advanced and high precision analysis methods will be an asset for several future missions.

The PEET software is considered to be an important complement of the EPEE Handbook to achieve such an integrated, responsive, tailorable and more accurate pointing budgeting and analysis process. It supports the user of the EPEE Handbook by making it intuitively accessible. This is stated based on the fact that the application of the PEET prototype V0.3 in the benchmark study and other projects at Airbus Defence and Space has produced reliable results besides a few encountered software implementation bugs.

Several research and developments needs have been identified and highlighted in this paper. These needs shall encourage the community in the field of pointing error engineering to pick-up the needs and find solutions.

ACKNOLEDGMENTS

In the course of writing this paper several discussions took place with engineers in Airbus Defence and Space, which have been involved in the compilation and analysis of pointing error budgets. The results obtained in this paper, especially the needs analysis, have been consolidated in these discussions. Topics involving the PEET software have been discussed with the software producers Astos Solutions and the Institute of Flight Mechanics and Control at the University of Stuttgart. Herewith we thank the colleagues in Airbus Defence and Space, Astos Solutions and the Institute of Flight Mechanics and Control for the valuable discussions.

REFERENCES

[1] ESA Engineering Standardisation Board, “Pointing Error Engineering Handbook ESSB-HB-E-003”, ESA-*ESTEC Requirements & Standards Division*, 2011.
[2] 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.
[3] Casasco M., Saavedra Criado G., Weikert S., Eggert J., Hirth M., Ott T., Su H., "Pointing Error Budgeting for High Pointing Accuracy Mission using the Pointing Error Engineering Tool", AIAA-2013-5251, *AIAA Guidance, Navigation, and Control Conference*, 19-22 August 2013, Boston, Massachusetts, USA.
[4] Winkler S., Cirillo F., Ergenzinger K., Ott T., Wilhelm R., Zaunick, E., "High-precision attitude determination and control of the EUCLID spacecraft: challenges and solutions", *8th International ESA Conference on Guidance, Navigation and Control Systems*, Karlovy Vary (2011)
[5] Del Bello U., Massotti L., "ESA Studies on HR imaging from geostationary Satellites - mission requirement", *Second GEO-HR user consultation workshop*, April 2013
[6] European Space Agency, "MetOp Second Generation (MetOp-SG) Space Segment Requirements Document (SSRD)", MOS-RS-ESA-SYS-0001, Issue 3, 2013
[7] ECSS - European Cooperation for Space Standardization, "Control Performance Standard", ECSS-E-ST-60-10C, 2008.
[8] European Space Agency, "Plato pointing requirements definition document", ESA/SRE-PA/2010.065, Issue 2, 2010
[9] Fichter W., Schleicher A., Bennani S., Wu S., "Closed Loop Performance and Limitations of the LISA Pathfinder Drag-Free Control System", AIAA 2007-6732, *AIAA Guidance, Navigation, and Control Conference*, 20-23 August 2007, Hilton Head, South Carolina, USA.
[10] Heine F., Kämpfner H., Lange R., Czichy R., Lutzer M., Meyer R., "Laser communication applied for EDRS, the European data relay system", *CEAS Space Journal 2.1-4 (2011): 85-90.*
[11] Haskins C., et al., "Systems Engineering Handbook", *INCOSE*, 2006.
[12] Balas G., Seiler P., Packard A., "Analysis of an UAV Flight Control System Using Probabilistic mu", AIAA 2012-4989, *AIAA Guidance, Navigation, and Control Conference*, 13 - 16 August 2012, Minneapolis, Minnesota, USA.
[13] MATLAB®, “Robust Control Toolbox”, *MathWorks*, 2011.
[14] ECSS - European Cooperation for Space Standardization, www.ecss.nl (30 June 2014)
[15] ECSS - European Cooperation for Space Standardization, "Control Performance Guidelines", ECSS-E-HB-60-10A, 2010
[16] ECSS - European Cooperation for Space Standardization, "Stars sensors terminology and performance specification Standard", ECSS-E-ST-60-20C, 2008
[17] ECSS, "Gyros terminology and performance Standard", ECSS-E-ST-60-21C, 2008
[18] 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.
[19] Ott T., Fichter W., Bennani S., Winkler S., "Precision Pointing H_∞ Control Design for Absolute, Window-, and Stability-Time Errors", *CEAS Space Journal*, pp. 13-30.
[20] Eggert J., et al., "PEET Software User Manual", ASTOS-PEET-SUM-001, Issue 1.7, July 2013.