

# **Standardization training program E-60 discipline: Control**

***Control Performance Standard***

***E-ST-60-10C***

***& ESA Pointing Error Engineering Handbook***

***ESSB-HB-E-003 Issue 1***

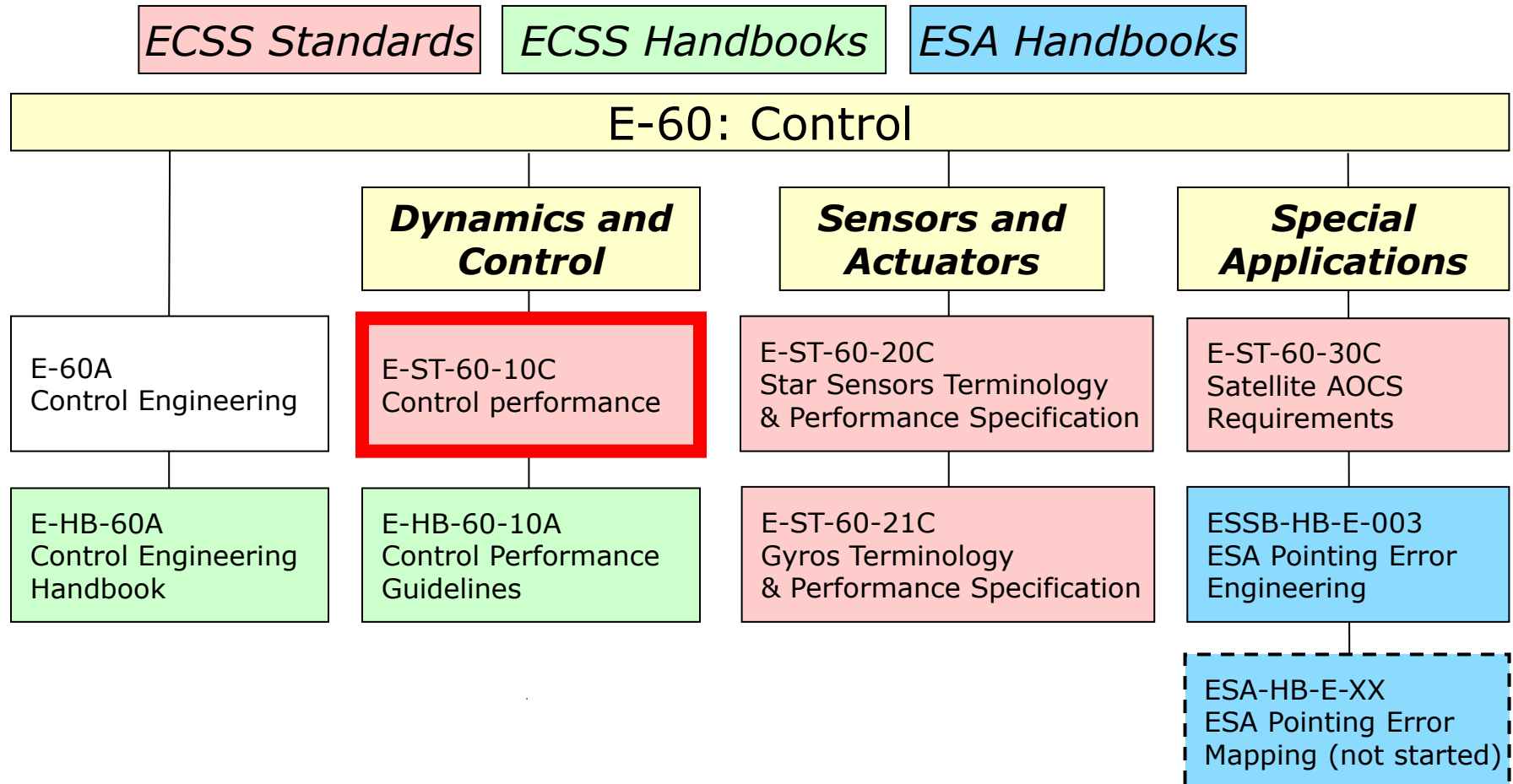
***& ESA Pointing Error Engineering Tool (PEET)***

# ***Control Performance Standard***

## ***E-ST-60-10C***

# The ECSS E60 branch

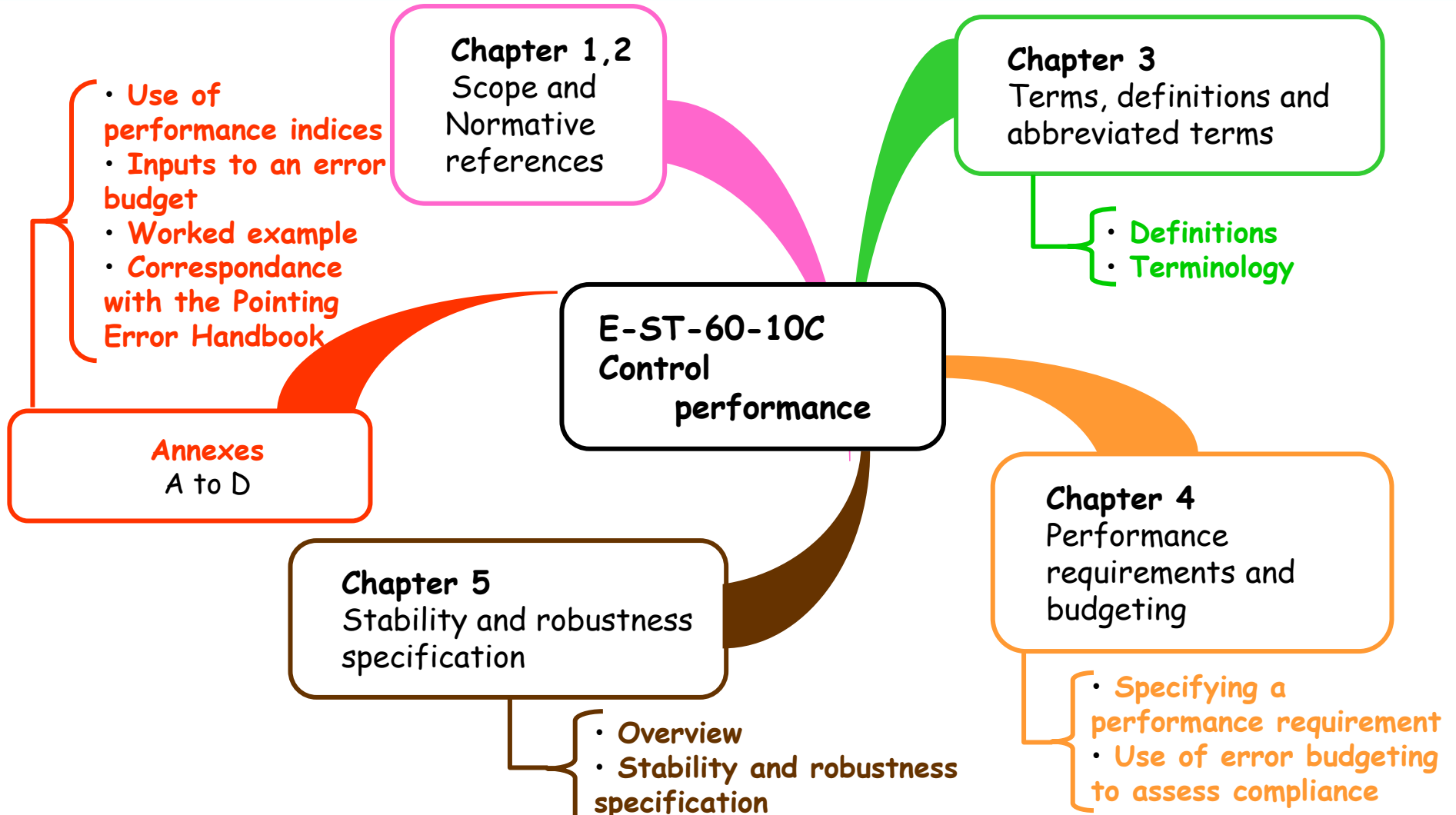
Standardization  
training program  
E60 discipline:  
Control



- The standard addresses the issue of control performance, in terms of definition, specification, verification and validation methods and processes.
  - It defines a general framework for handling **performance indicators**, which applies to all disciplines involving control engineering, at different levels ranging **from equipment to system level**.
  - It also focuses on the specific performance indicators applicable to the case of closed-loop control systems – mainly **stability and robustness**.
- Rules are provided for combining different error sources to build a **performance error budget** and use it to assess the **compliance** with a requirement.
  - Complementary material and guidelines for a step-by-step procedure for pointing error budgeting can be found in ECSS-E-HB-60-10A (Control Performance Guidelines) and in ESSB-E-HB-003 (Pointing Error Engineering Handbook)

# Table of contents

Standardization  
training program  
E60 discipline:  
Control



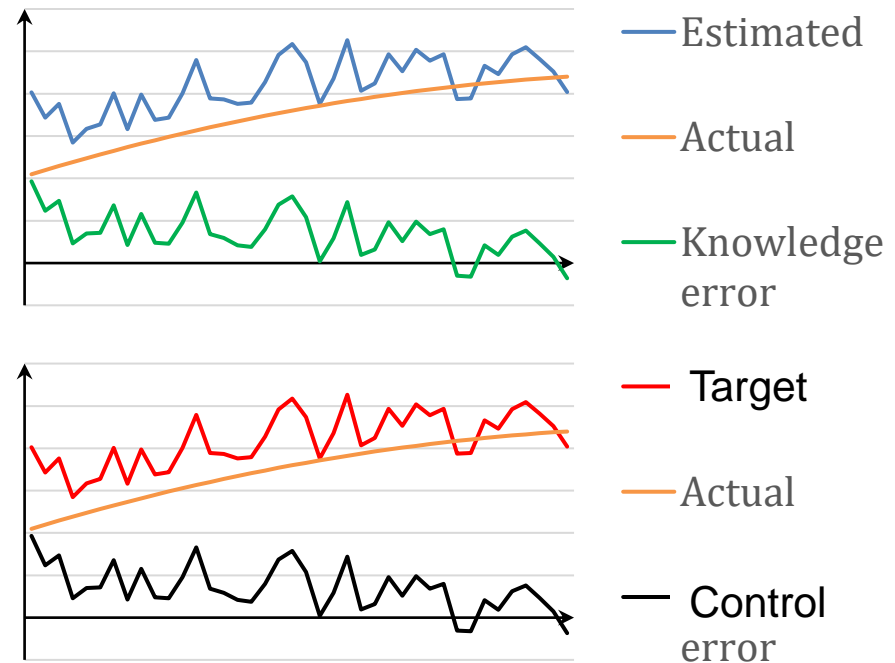
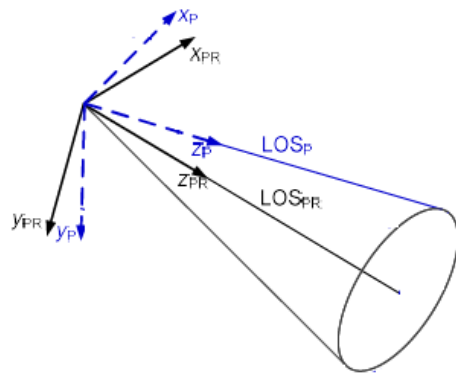
- **Chapter 3** gives the **definition of control performance**-related terms
  - Absolute knowledge error (AKE), absolute performance error (APE), error, error index, individual error source, knowledge error, ...
- **Chapter 4** treats **performance specifications** and rules for **error budgeting**
  - The main features of performance requirements specification are defined together with the fundamental rules for identification, characterisation, and combination of contributors of an error budget.
- **Chapter 5** treats **stability and robustness** specification and verification
  - Stability and robustness concepts for linear systems are introduced and requirements for their specification and verification are defined
- **Annex A** provides **guidelines** on using performance error **indices**
  - Statistical elements are provided that are useful in formulating error requirements and using and building an error budget
- **Annex B** discusses common types of errors and **rules to budget** for them
  - Formulas (some exact, some approximate) are provided to show how typical errors are budgeted for depending on the index and statistical interpretation

# Ch. 3 - Terminology – Pointing Error Indices

Different indexes may be used to define the requirements considering:

- If Knowledge error or Performance/Control error
- The relevant directions & axes (like across / along Line Of Sight / half-cone around it)
- The type of error index: Instantaneous values / Mean over a time window / Standard deviation over window (or similar) / Drift between windows (see next slide example)

$$e_{X,control} = X_{target} - X_{actual}$$
$$e_{X,knowledge} = X_{estimated} - X_{actual}$$



# Ch. 3 - Terminology – Pointing Error Indices

- Examples for performance requirements (same may be defined for knowledge):
  - APE**: absolute performance error (as opposed to relative, not as absolute value)
  - MPE**: mean performance error over window of defined duration
  - RPE**: relative performance (stability) as difference between APE and MPE (or similar)
  - PDE**: drift of APE/MPE over an observation/stability period

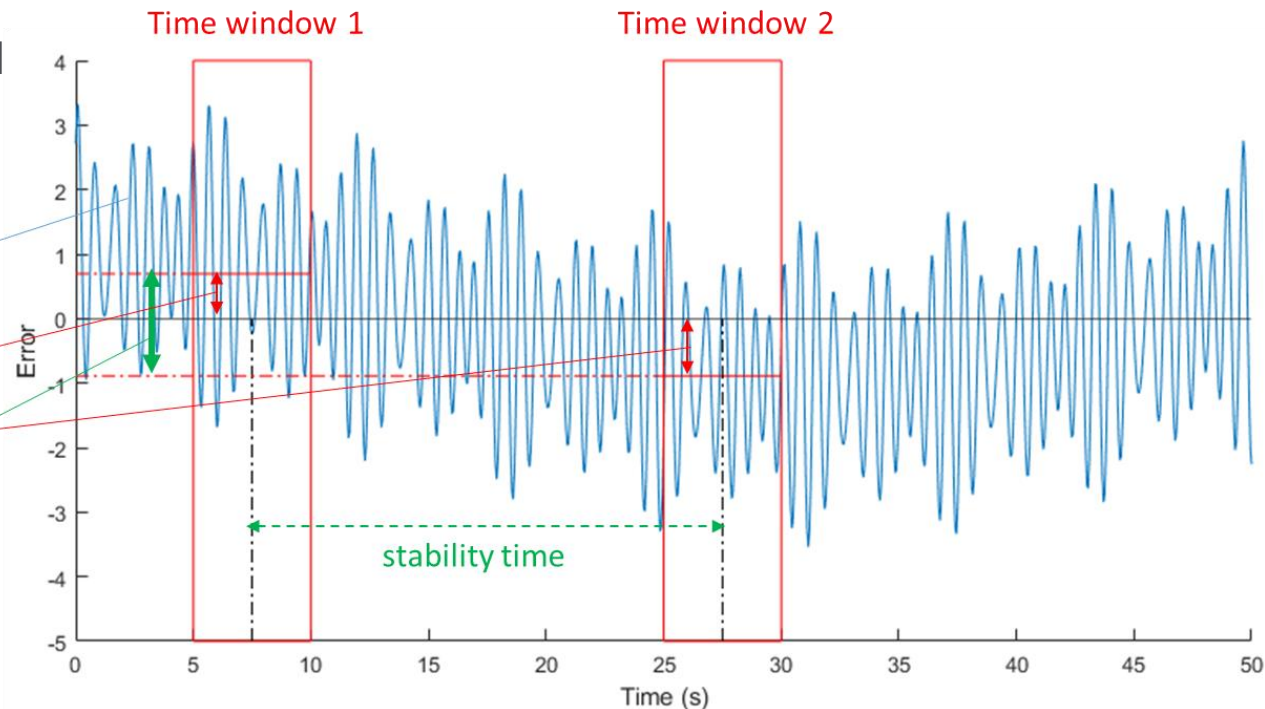
Absolute Error (AE)

Mean Error (ME)

- for time window 1
- for time window 2

Drift Error (DE)

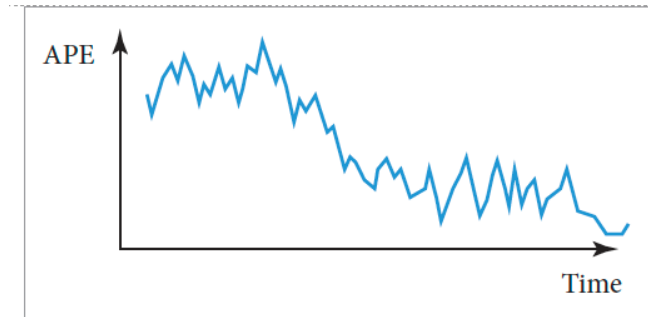
for time windows 1 & 2



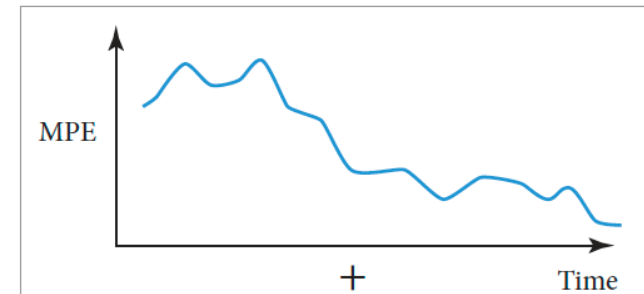


# Ch. 3 - Terminology – Pointing Error Indices

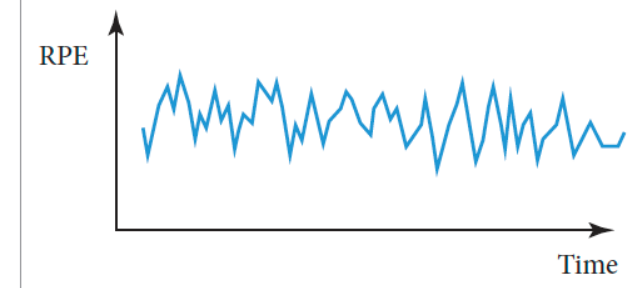
- The **MPE** represents the **low-frequency** portion of the APE
- The **RPE** represents the **high-frequency** portion of the APE
- The **APE** consists of the **combination** of the two:  $APE = MPE + RPE$



=



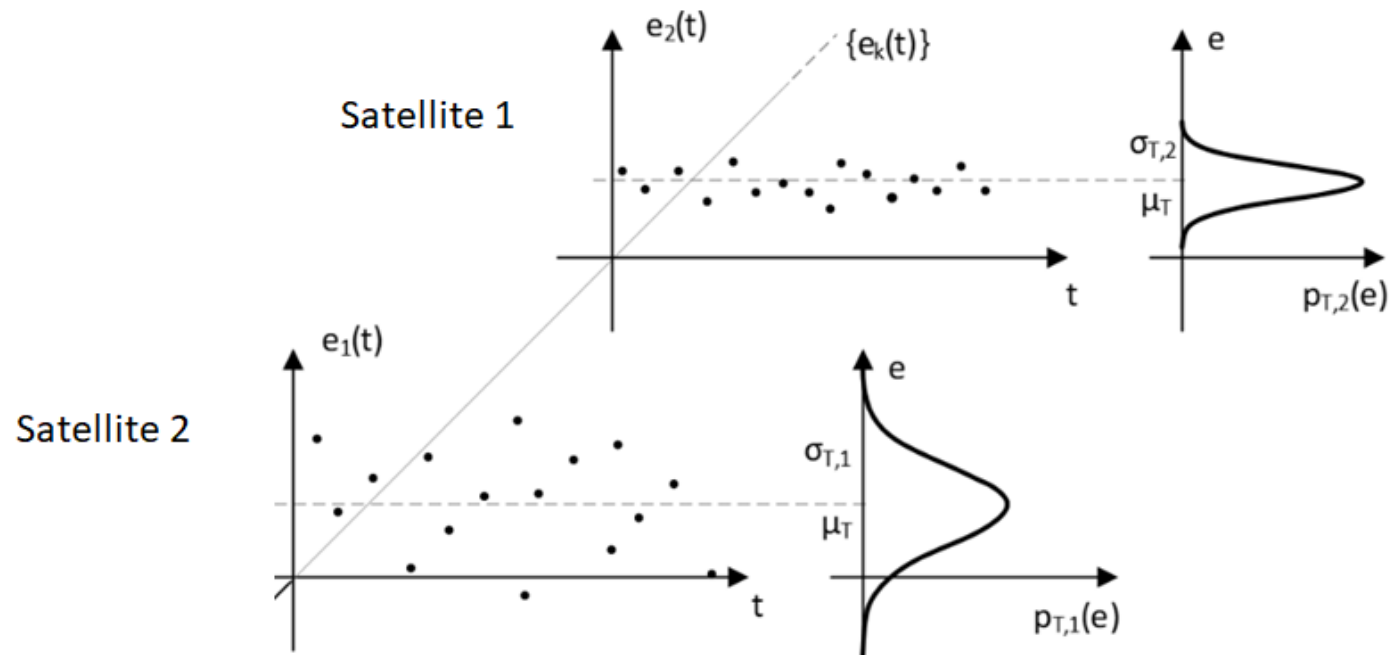
+



# Ch. 4 - Statistical interpretation

There are 2 types of statistical domains:

- Time
- Ensemble  
(over satellites with the same design but different manufacturing errors or in different configurations)



# Ch. 4 - Statistical interpretation

The confidence level is applied depending on the specified statistical interpretation:

Statistical interpretation	Temporal domain treatment	Ensemble domains treatment
Temporal interpretation	<b>Specified confidence level over temporal statistics</b>	Worst case configuration (for the defined Confidence level over time)
Ensemble interpretation	Worst case (100%) over time	<b>Specified confidence level over ensemble statistics</b>
Mixed interpretation	<b>Specified confidence level over temporal &amp; ensemble statistics together</b>	
Combined interpretation (planned for future next PEEH release)	Specified time confidence level over temporal statistics	Specified ensemble confidence level over ensemble statistics (for the defined Confidence level over time)

# Ch. 4 – Confidence level with Statistical interpretation

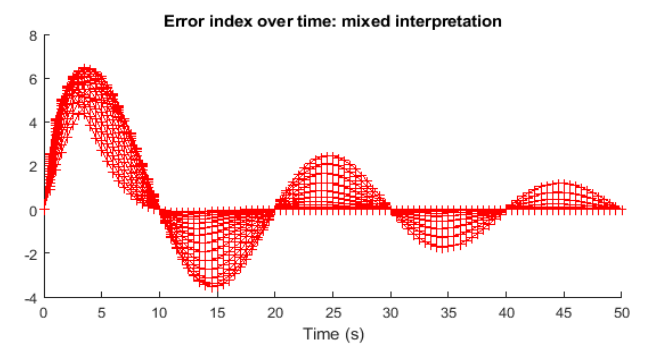
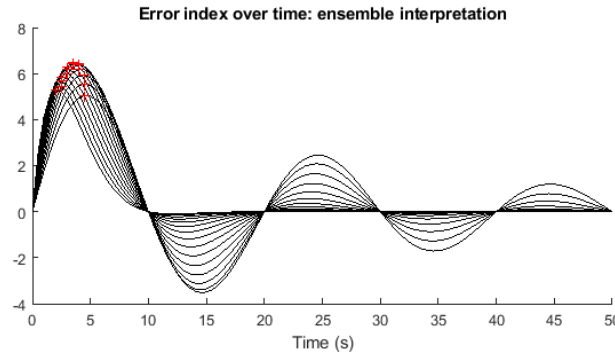
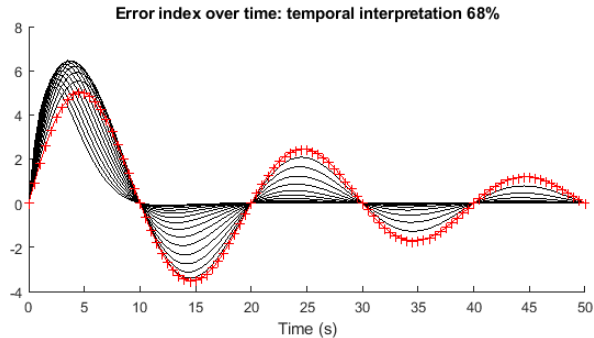
Steps to apply the Statistical interpretation:

- Get the index error over time for all configurations
- Apply confidence level over time (100% for ensemble interpretation)
- Apply confidence level over ensemble (100% for worst case if temporal)

Temporal interpretation  
(for fixed confidence here 68%)

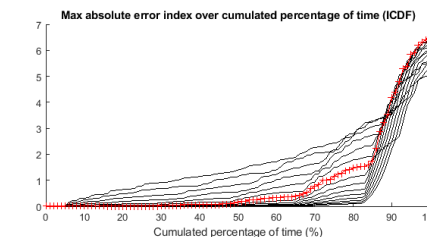
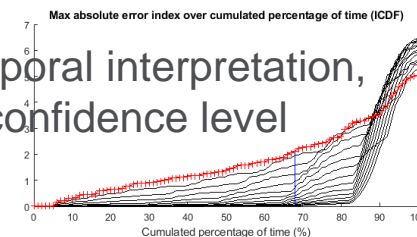
Ensemble interpretation

Mixed interpretation



In red, the errors points considered for the distribution on which applies the specified confidence level

When sorting the error over time for temporal interpretation, we see the worst case depends on the confidence level



# Ch. 4 - Performance requirements and budgeting

## → SPECIFYING A PERFORMANCE REQUIREMENT (SECTION 4.1)

- The following needs to be defined for a complete pointing requirement:
  - Required **error value** (e.g.  $0.1^\circ$ )
  - Reference **frame and axis** of angular deviation (or half-cone) (e.g. pitch axis in orbital frame)
  - **Error index**: APE, MPE, RPE, PDE, PRE (or knowledge error)
  - **Confidence level**  $P_c$   
Note: The n- $\sigma$  notation shall be restricted to cases where the Gaussian distribution holds. (due to central limit theorem, this hypothesis may be applicable)
  - **Statistical interpretation**: ensemble, temporal or mixed
  - Evaluation period (e.g. in fine pointing mode)
  - Optional: PSD profile in bandwidth of interest
- This is mathematically expressed as:  $prob(|X| < X_{\max}) \geq P_c$

# Ch. 4 - Performance requirements and budgeting

## → SPECIFYING A PERFORMANCE REQUIREMENT (SECTION 4.1)

### ● Example of requirements:

- *"The APE of the half cone angle error of the payload boresight direction (instantaneous error between the actual and desired directions) shall be less than 1,0 arcmin with 95 % Confidence Level over time with temporal statistical interpretation of the time"*
- *"Over a 10 second integration time, the Euler angles for the transformation between the target and actual payload frames shall have an RPE less than 20 arcsec at 99 % confidence, using the mixed statistical interpretation."*

## Using error budgeting to assess compliance (section 4.2)

- A common way to assess compliance is to compile an **error budget**
  - This involves **taking the known information** about the contributing sources, then **combining them** to estimate the behaviour of the overall performance error, which can then be compared to the original requirement.
  - It is important to emphasise that the common methods of budgeting are **approximate** only, and therefore must be **used with care**. They are based on the assumption from the central limit theorem that the distribution of the total error is Gaussian, and therefore completely specified by its mean and variance per axis.
  - It is not possible to give quantitative limits on its domain of validity: **a degree of engineering judgement is involved**.
- Assessment of correlation between error terms: e.g. correlated by temperature

## Using error budgeting to assess compliance (section 4.2) (continued)

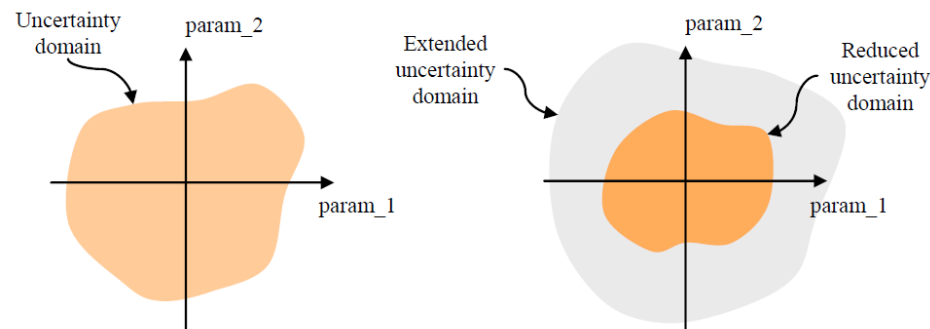
- Then contributing errors are classified into groups and characterised
  - A common classification is to distinguish between **biases, random errors, harmonic errors** with various periods, etc.
    - All errors which can potentially be correlated with each other shall be classified in the same group.
    - But a group shall not contain a mixture of **correlated** and **uncorrelated** errors
  - **The period of variation (short term, long term, systematic) is not a sufficient classification criterion**, as by itself it provides no insight into whether or not the errors can be correlated.
  - For each error source, a **mean** and **standard deviation** shall be allocated along each axis. Guidelines for obtaining these parameters are given in **Annex B** and are complemented by the methodology presented in the **Pointing Error Engineering Handbook**.



# Ch. 5 - Stability and robustness specification

- Terminology (section 3)
  - Stability**: ability of a system submitted to bounded external disturbances to remain indefinitely in a bounded domain around an equilibrium position or around an equilibrium trajectory
  - Performance**: the system output meets the accuracy requirements
  - Robustness**: ability of a controlled system to maintain some performance and stability characteristics in the presence of plant, sensors, actuators and/or environmental uncertainties
  - Stability margin**: maximum excursion of some parameters describing a given control system for which the system remains stable
- The state-of-the-art for stability specification is not fully satisfactory.
  - A traditional rule exists, going back to the times of analogue controllers, asking for a gain margin  $> 6$  dB, and a phase margin  $> 30^\circ$ .
- A more consistent method is proposed for specifying stability & robustness (section 5.2)
  - For SISO loops **the gain margin, the phase margin and the modulus margin** shall be used as default indicators.
  - For MIMO loops use **sensitivity (disturb.) and complementary sensitivity (noise) functions** (with structured singular value)
  - If other indicators are selected by the supplier, this deviation shall be justified

- In order to handle uncertainties (see the previous definition of robustness), **uncertainty domains** must be specified that consist of:
  - A list of **uncertain parameters**
  - For each of these parameters, an interval of uncertainty (or a dispersion) around the nominal value (in percentage or based on a statistical distribution)
  - When relevant, the root cause for the uncertainty (normally lack of characterisation, errors in parameters measurements, **changes over life**)
- A single uncertainty domain or **reduced/extended domains** may be defined
  - Nominal stability margins shall be demonstrated over the entire uncertainty domain (if a single domain is defined – section 5.2.6) or over the reduced domain (if reduced and extended domains are defined - section 5.2.7).
  - Degraded stability margins shall be demonstrated over the extended domain (section 5.2.7).



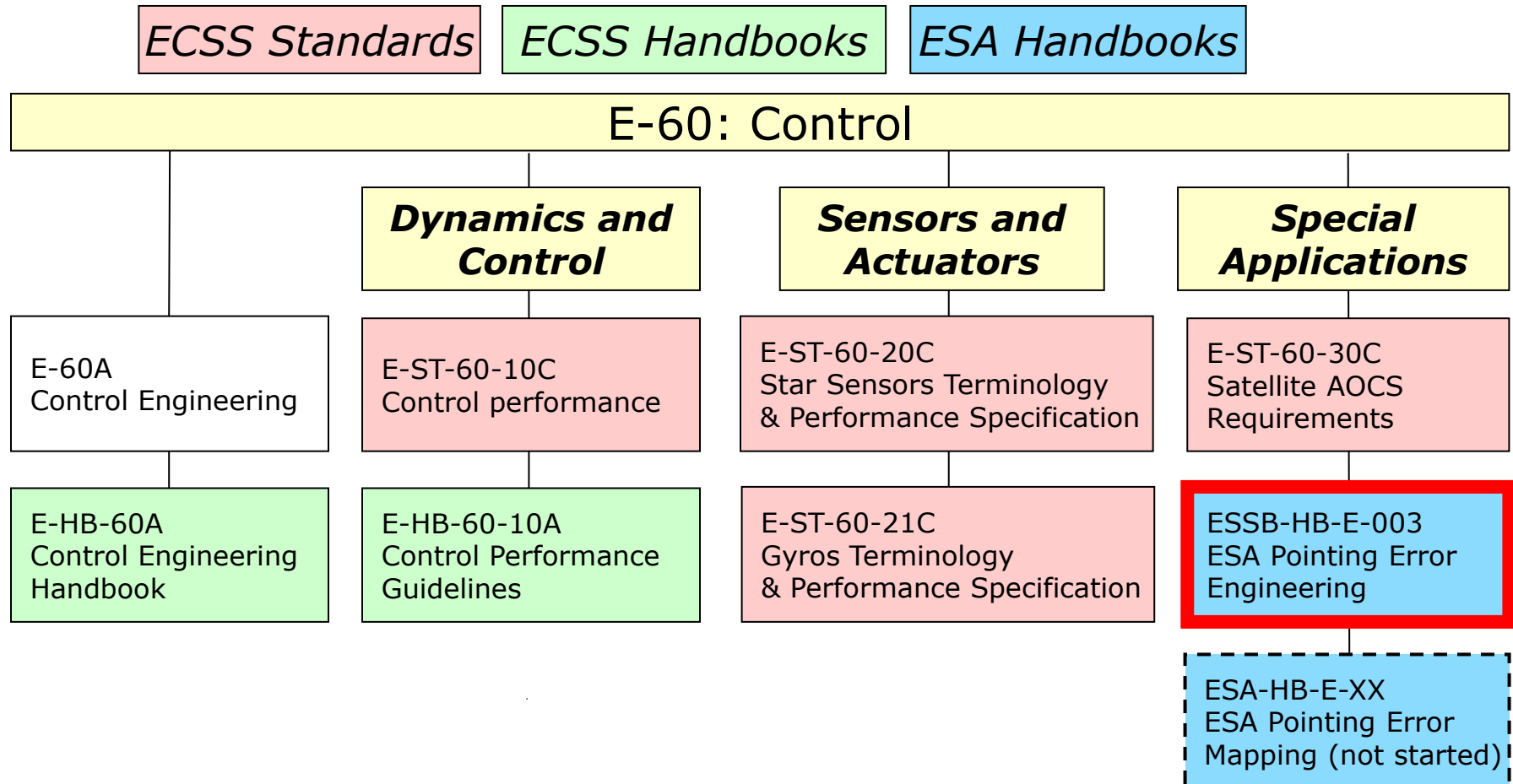
- **Extension** transversally (other subsystems) and vertically (up to system level)
- Control performance engineering **tasks** during the project phases:
  - Requirements definition
  - Design and configuration
  - Analysis
  - Verification and validation
- Guidelines for **extrinsic** control performance (pointing performance)
  - Formulation of requirements, selection of indices and interpretations
  - Verification: Monte Carlo vs budgeting
- Guidelines for **intrinsic** control performance (stability and robustness)
  - Basic control theory, stability margins, sensitivity and complementary sensitivity functions
  - Performance indicators: settling time, overshoot, tracking performance,...
- SPOT-5 **example**

- The Control Performance Standard E-ST-60-10C provides a set of normative definitions, budget rules, and specification templates needed to manage performance aspects of control systems in the frame of space projects.
- The standard is split into two main clauses:
  - Performance error indices and analysis methods
  - Stability and robustness specification and verification for linear systems
- This normative standard is accompanied by the guidelines provided in ECSS-E-HB-60-10A - Control Performance Guidelines and especially in ESSB-E-HB-003 – Pointing Error Engineering Handbook
- This standard together with the Pointing Error Engineering Handbook provides all the mathematical elements to carry out pointing error budgeting activities

***ESA Pointing Error Engineering  
Handbook  
ESSB-HB-E-003 Issue 1***

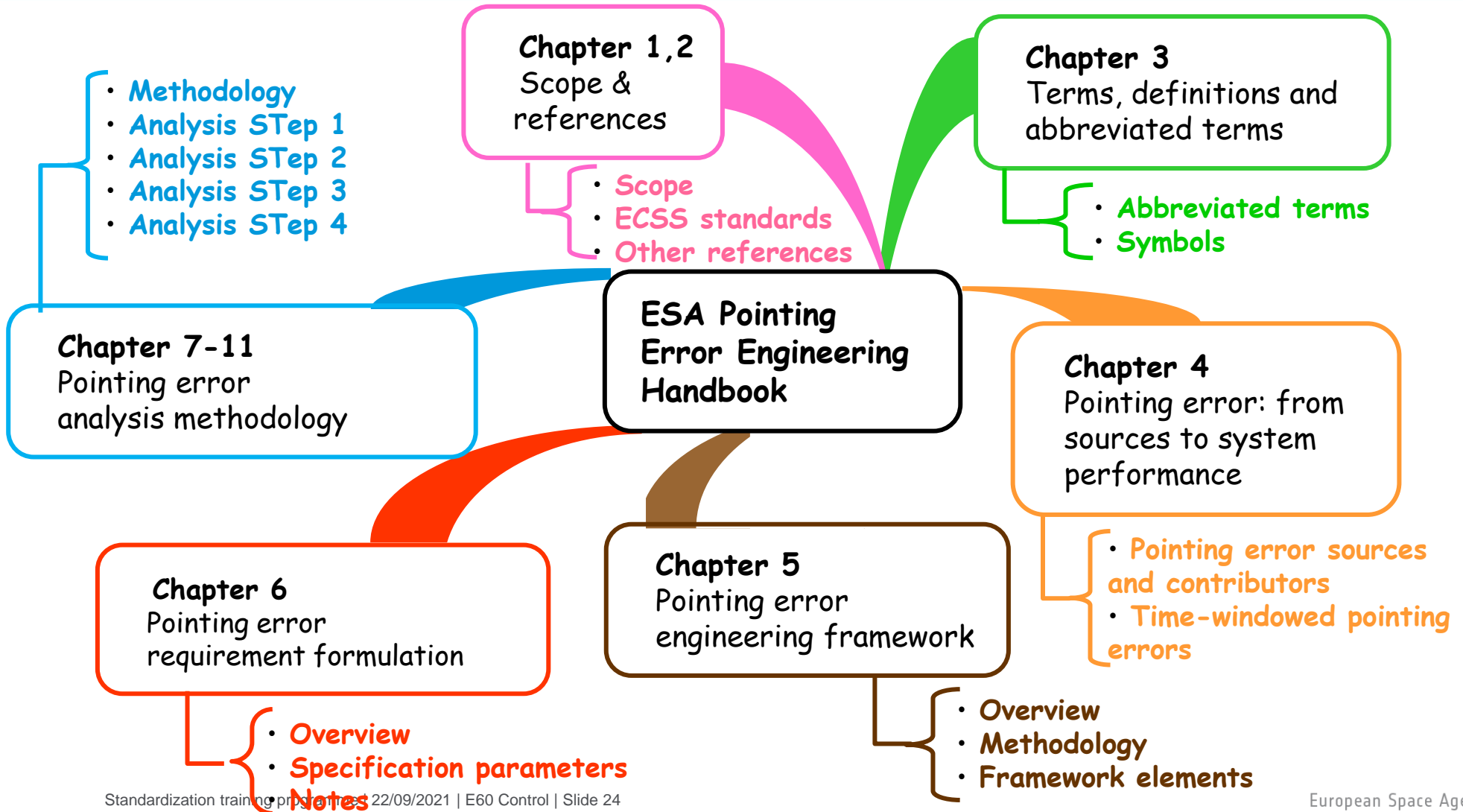
# The ECSS E60 branch

Standardization  
training program  
E60 discipline:  
Control



- This handbook provides a **consistent and mathematically justified set of guidelines and summation rules** to define an engineering process to build pointing error budgets.
- It embeds the elements of the E-ST-60-10C standard in a **step-by-step engineering process**, providing guidelines for:
  - Characterising pointing error **sources**
  - Analysing the pointing error source **contribution** to the ECSS pointing error indexes
  - Compiling system pointing **error budgets**
- It replaces the old ESA Pointing Error Handbook Ref. ESA-NCR-502 (19 Feb 1993)
- It was adopted in July 2011 in the **LEAS** (List of ESA Applicable Standards)
- A prototype of the accompanying tool PEET (Pointing Error Engineering Tool) has been developed in 2012, now finalized in 2017, with update in 2022.

# Table of contents





- **Chapter 4** introduces the **fundamental definitions**
  - Pointing error source (PES) and contributors (PEC), system transfers, time-windowed pointing errors, mathematical formulation of error indexes
- **Chapter 5** describes the pointing error engineering **framework**
  - The methodology of the engineering process for establishing pointing error requirements, their systematic analysis and eventually compliance verification and the related mathematical elements are described
- **Chapter 6** provides guidelines for pointing error **requirements formulation**
  - Includes notes on reference frames, statistical interpretation, evaluation period, level of confidence and requirement break-down and allocation
- **Chapters 7-11** define the **step-by-step methodology** for budget compilation
  - Pointing error source characterisation, pointing error source transfer analysis, error index contribution analysis, system pointing error evaluation
- **Annexes** support the main chapter and provide complementary information
  - Details on definition of pointing scene, examples on using statistical interpretations, notes on system norms, error metrics and summation rules

- The old ESA Pointing Error Handbook (ESA-NCR-502, 1993) introduced the basic concepts for pointing error budgeting and proposed:
  - an **exact method** for budget compilation (based on PDF characterisation via convolution of different errors, **hardly applicable in practice**)
  - **simplified summation rules** that (can lead to under- or over-estimates of the pointing performance and lack mathematical justification). Examples:
    - Herschel:  $\sigma_{total} = \sigma_{Bias} + \sigma_{LongTerm} + \sigma_{ShortTerm}$  (RSS in each class)
    - Sentinel 2:  $\sigma_{total} = \sigma_{ST} + \sigma_B + \sqrt{(\sigma_H^2 + \sigma_N^2)}$
    - MTG:  $\sigma_{total} = \sigma_{ST} + \sqrt{(\sigma_B^2 + \sigma_H^2 + \sigma_N^2)}$
- **Simplified summation rules are not endorsed by E-ST-60-10C**, which foresaw the necessity of a complementary document:
  - “For their own specific purpose, each entity (ESA, national agencies, primes) can further elaborate internal documents, deriving appropriate guidelines and summation rules based on the top level clauses gathered in this E-ST-60-10C standard.”
- Action from ESA Engineering Standardisation Board meeting ESB#40:  
«To draft and finalize a normative text to replace the ESA Pointing Error HB»

# Purpose and approach

Standardization  
training program  
E60 discipline:  
Control

- Purpose of ESA PEEH
  - Pointing Error Budgeting
  - Consistent with and elaborating the E-ST-60-10C ECSS (no additional clause, **no "shall"** )
  - To be used by ESA projects as an **applicable document complementing E-ST-60-10C**
- Approach and added value
  - **Generic process for any mission type and any design phase** through 4 Analysis Steps: AST-1 to -4
  - Added value of frequency domain analysis via PSD on stationary random processes: **windowing in time domain is equivalent to filtering in frequency domain**

**AST-1:** Pointing Error Sources  
Characterization



**AST-2:** Pointing Error Sources  
Transfer Analysis



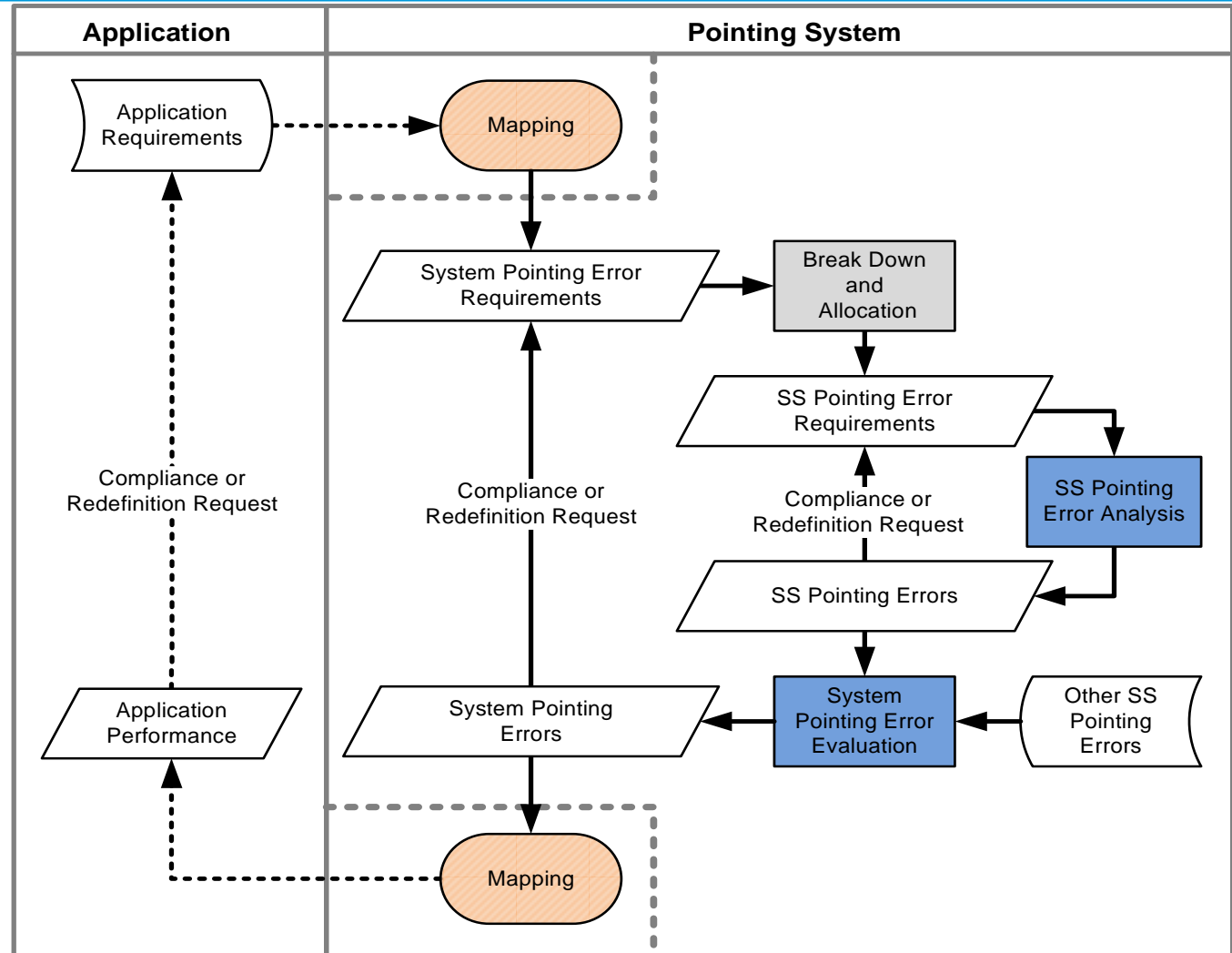
**AST-3:** Error Index  
Contribution Analysis



**AST-4:** System Pointing Error  
Evaluation

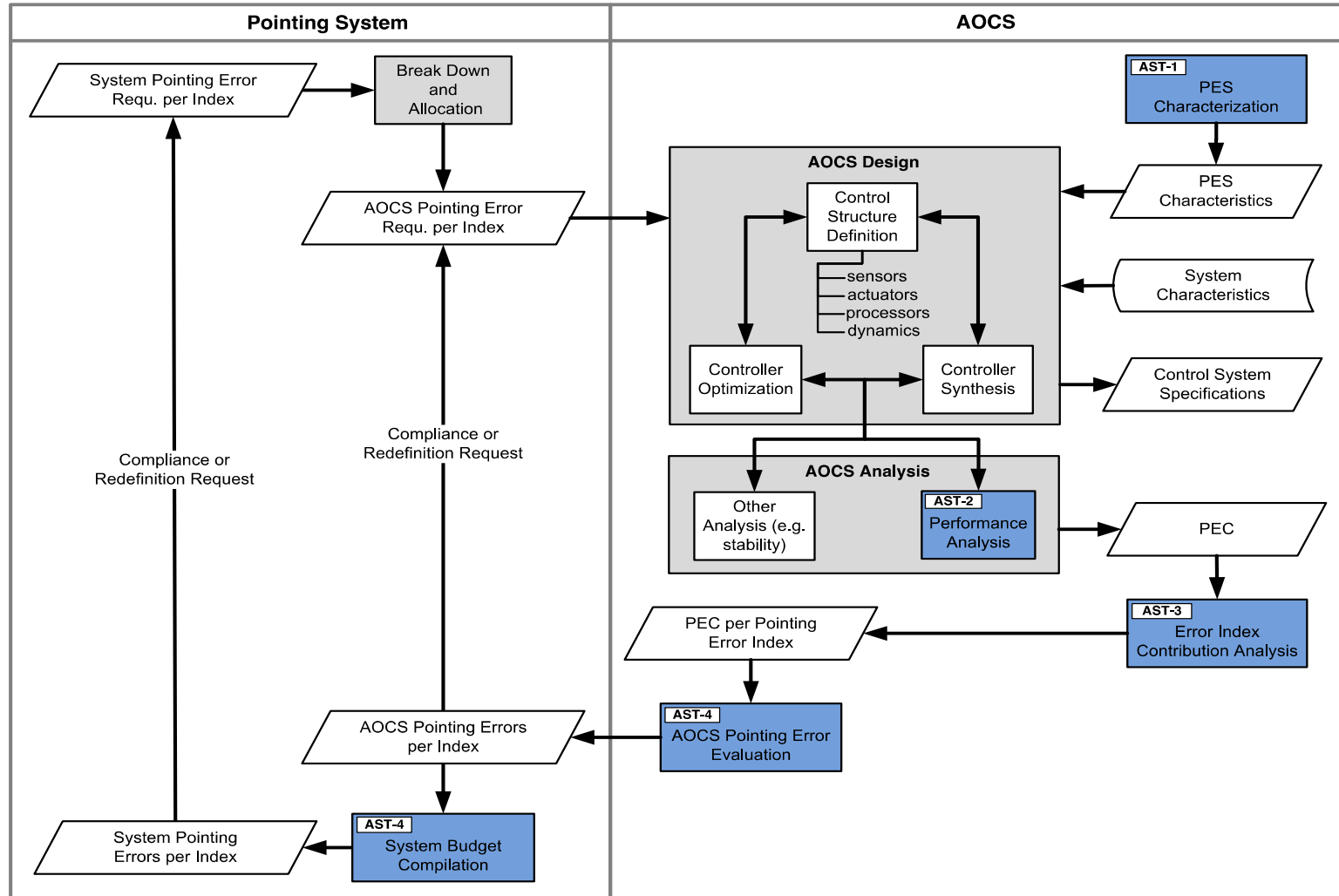
# Pointing Error Engineering Cycle

- Mapping from science requirements to spacecraft pointing requirements and identification of pointing error sources (PES) are strongly application specific steps and, as such, are currently excluded from PEEH



# Pointing Error Analysis and Evaluation Methodology (1/2)

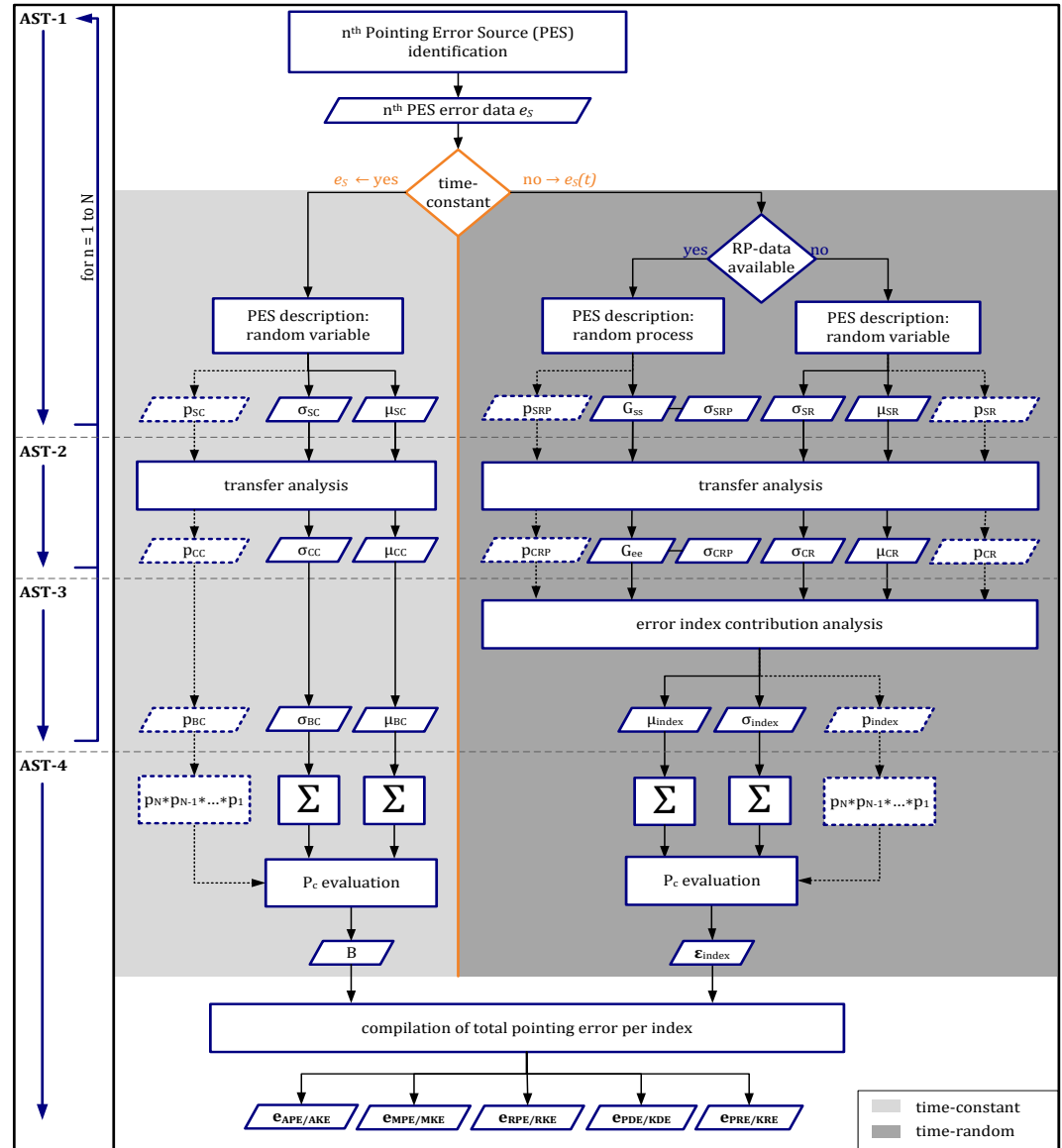
Standardization training program  
E60 discipline:  
Control



# Pointing Error Analysis and Evaluation Methodology (2/2)

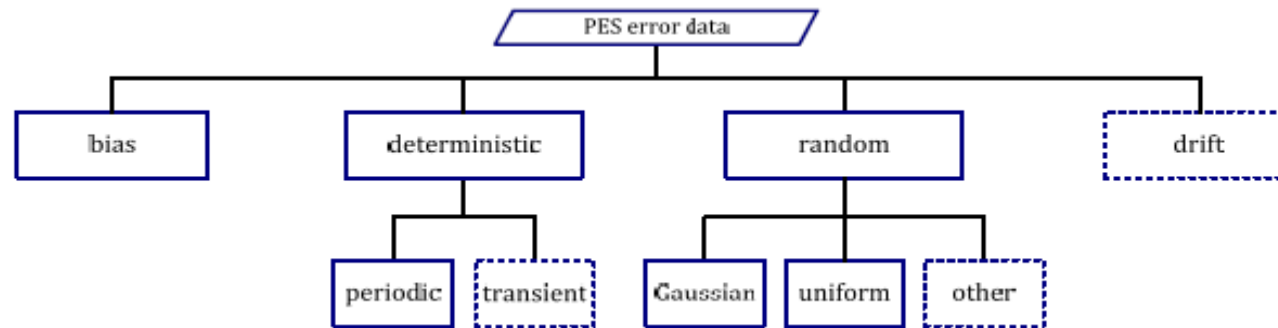
Standardization training program  
E60 discipline:  
Control

- Simplified statistical method:** analysis via variance ( $\sigma$ ) and mean ( $\mu$ ) and their summation per ECSS error index under the assumption of the central limit theorem, solid lines in figure
- Advanced statistical method:** analysis by joint PDF characterisation via convolution of different error PDF  $p_{...}(e)$ , dashed lines in figure



# AST-1: PES Characterisation (1/3)

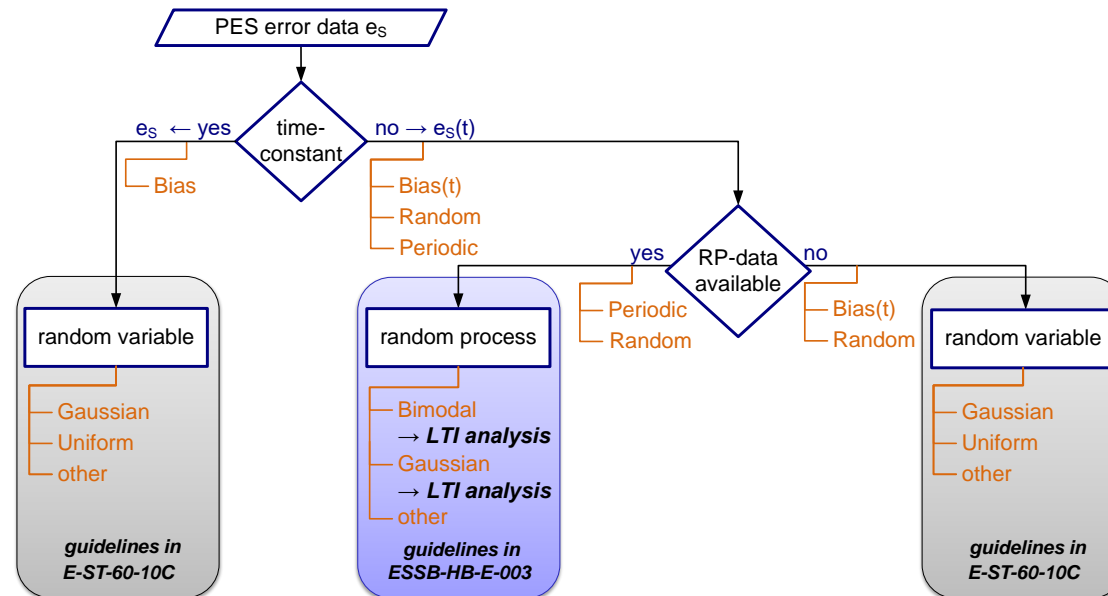
- First step for pointing error sources characterisation is their **categorisation into signal classes** (bias, periodic, random, etc.) that are analogous to those defined and considered in the ECSS standard



- PES data categorisation into signal classes allows simplifying the pointing error analysis **without loss of generality**

# AST-1: PES Characterisation (2/3)

- A decision tree is defined, which provides guidelines for selecting eligible mathematical elements to describe the signal-classified PES data



- Differently from the current approach in E-ST-60-10C, description of PES as stationary random process (together with characterising its PSD) allows **exact capture** of window and stability time properties.



# AST-1: PES Characterisation (3/3)

- Example: 2 PES with temporal interpretation specified

characterization steps	STR-Payload misalignment error	Gyro-Stellar Estimator (GSE) noise
temporal behavior	time-constant	time-random
signal class	bias	random
description	random variable	random process
<b>reference document</b>	<b>ECSS</b>	<b>ESA PEE Handbook</b>
characteristic data	$U(0, e_{max})$	$G(0, \sigma_G), G_{ee}$
temporal interpretation	$p(e) = \delta(e_{max})$	$G(0, \sigma_G)$
<b>AST-1 output data</b>	$\mu(e) = e_{max}, \sigma(e) = 0$	$\mu(e) = 0, \sigma(e) = \sigma_G, G_{ee}$

# AST-1: PES vs Statistical Interpretation

- ESA requirement must define the considered statistical population by specifying one of the three **statistical interpretations** introduced in ECSS-E-ST-60-10C:

- Ensemble**  $\text{Prob}[|e_{\max}(k)| < e_r] \geq P_c$  with  $e_{\max}(k) = \max_t\{e_k(t)\}$   
i.e. WC time but error statistically averaged over a population of S/C
- Temporal**  $\text{Prob}[|e_{\max}(t)| < e_r] \geq P_c$  with  $e_{\max}(t) = \max_k\{e_k(t)\}$   
i.e. WC S/C but error statistically averaged over time
- Mixed**  $\text{Prob}[|e(k,t)| < e_r] \geq P_c$   
i.e. time and ensemble considered concurrently (conditional probability)

- The PEEH highlights this **important feature**: the mixed interpretation automatically yields:

- to the ensemble one if a PES does not randomly vary in time (time constant, e.g. alignment bias)
- to the temporal interpretation if it does not vary over the population (ensemble-constant, e.g. sensor typical noise ).

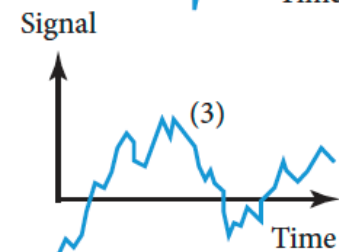
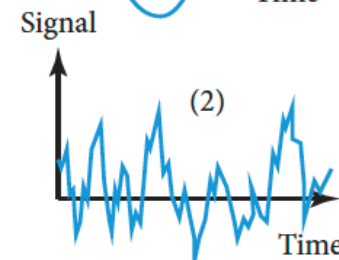
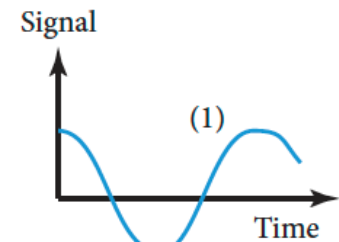
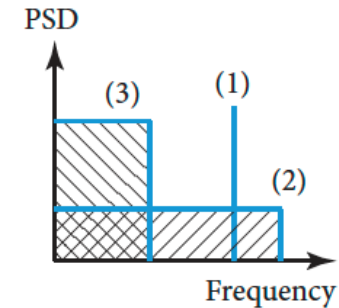
# Transfer analysis: Power Spectral Density (PSD) introduction

- Power spectral density (PSD) describes the frequency content of a random process
- The mean square value of the signal is equal to the integral under the PSD curve:

$$E[x^2] = \int_0^{\infty} S_x(f)df$$

- White noise has a flat PSD curve
- Propagation through a transfer function H:

$$S_y(\omega) = |H(i\omega)|^2 S_x(\omega)$$

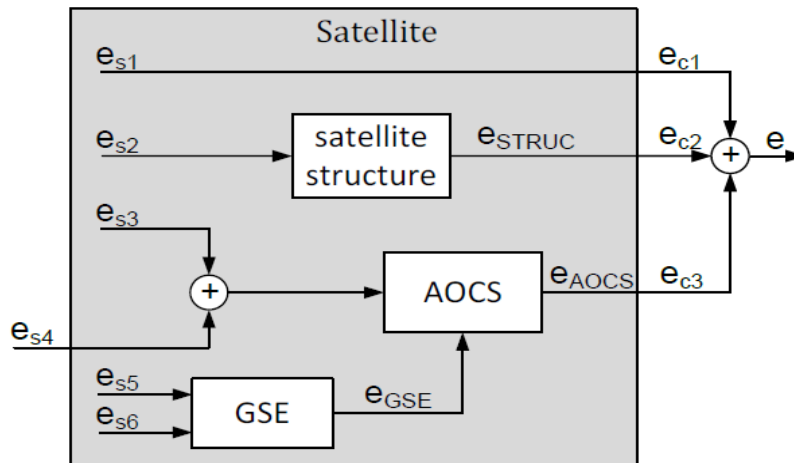


# AST-2: Transfer Analysis (1/2)



Transfer from a PES point of origin to the point of interest for error calculation:

- Coordinate frame transformation
- Closed-loop control system
- Thermo-structural transformation



	<b>PES</b>
$e_{s1}$	STR-Pay. misalignment
$e_{s2}$	$\mu$ Vibrations
$e_{s3}$	thruster noise
$e_{s4}$	solar pressure noise
$e_{s5}$	STR noise
$e_{s6}$	gyro noise

- Transfer analysis of time-random PES described as random process can be performed in either time- or frequency-domain:
  - The time-domain approach relies on simulations that, based on the PES PDF or PSD, provide a statistical sample that can be used to characterise the PEC at the point of interest
  - The frequency-domain approach relies on linear transformation of statistical properties

- For LTI systems, the PSD transfer relation can be used:

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

- One **advantage of the frequency-domain approach** is that it can be **used to tune the system transfer function**  $H$  based on signal and system norms

# AST-3: Error Index Contribution Analysis (1/3)

- Time-domain approach is based on simulations and described in the ECSS standard
- Frequency-domain approach provides exact contribution analysis via PSD weighting functions

$$\sigma_e^2 = \frac{1}{2\pi} \int_0^\infty G_{ee}(\omega) d\omega$$

↓

$$\sigma_{index}^2 = \frac{1}{2\pi} \int_0^\infty G_{ee}(\omega) F_{metric}(\omega) d\omega$$

Time-Windowed Error Metrics (Lucke et al. & Pittelkau)			
index:= metric		time domain	frequency domain
APE, AKE:= Absolute (ABS) Metric	$\sigma_{index}^2 :=$	$\sigma_{ABS}^2 = E[(e(t) - \mu_{ABS})^2]$	$= \frac{1}{2\pi} \int_0^\infty G_{ee}(\omega) d\omega$
MPE, MKE:= Windowed Mean (WM) Metric	$\sigma_{index}^2 :=$	$\sigma_{WM}^2(\Delta t) = E[\langle (e(t))_{\Delta t} - \mu_{ABS} \rangle^2]$	$= \frac{1}{2\pi} \int_0^\infty G_{ee}(\omega) F_{WM}(\omega, \Delta t) d\omega$
RPE, RKE:= Windowed Variance (WV) Metric	$\sigma_{index}^2 :=$	$\sigma_{WV}^2(\Delta t) = E[\langle (e(t) - \langle e(t) \rangle_{\Delta t})^2 \rangle_{\Delta t}]$	$= \frac{1}{2\pi} \int_0^\infty G_{ee}(\omega) F_{WV}(\omega, \Delta t) d\omega$
PDE, PRD, KDE, KRE:= Windowed Mean Stability (WMS) Metric	$\sigma_{index}^2 :=$	$\sigma_{WMS}^2(\Delta t, \Delta t_s) = E[\langle (e(t))_{\Delta t} - \langle e(t - \Delta t_s) \rangle_{\Delta t} \rangle^2]$ $\Delta t = \Delta t_1 = \Delta t_2$	$= \frac{1}{2\pi} \int_0^\infty G_{ee}(\omega) F_{WMS}(\omega, \Delta t, \Delta t_s) d\omega$
PDE, PRD, KDE, KRE:= Stability (STA) Metric	$\sigma_{index}^2 :=$	$\sigma_{STA}^2(\Delta t_s) = E[(e(t) - e(t - \Delta t_s))^2]$	$= \frac{1}{2\pi} \int_0^\infty G_{ee}(\omega) F_{STA}(\omega, \Delta t_s) d\omega$

# AST-3: Error Index Contribution Analysis (2/3)

Standardization training program  
E60 discipline:  
Control

Pointing Error Metric Weighting Functions $F_{\text{metric}}$	
<b>Windowed Mean (WM)</b>	
$F_{WM}(\omega, \Delta t) = \frac{2(1 - \cos(\omega\Delta t))}{(\omega\Delta t)^2}$	
<p>rational approximation:</p> $\tilde{F}_{WM}(s, \Delta t) = \frac{2(s\Delta t + 6)}{(s\Delta t)^2 + 6(s\Delta t) + 12}$	
<b>Windowed Variance (WV)</b>	
$F_{WV}(\omega, \Delta t) = 1 - \frac{2(1 - \cos(\omega\Delta t))}{(\omega\Delta t)^2}$	
<p>rational approximation:</p> $\tilde{F}_{WV}(s, \Delta t) = \frac{s\Delta t(s\Delta t + \sqrt{12})}{(s\Delta t)^2 + 6(s\Delta t) + 12}$	



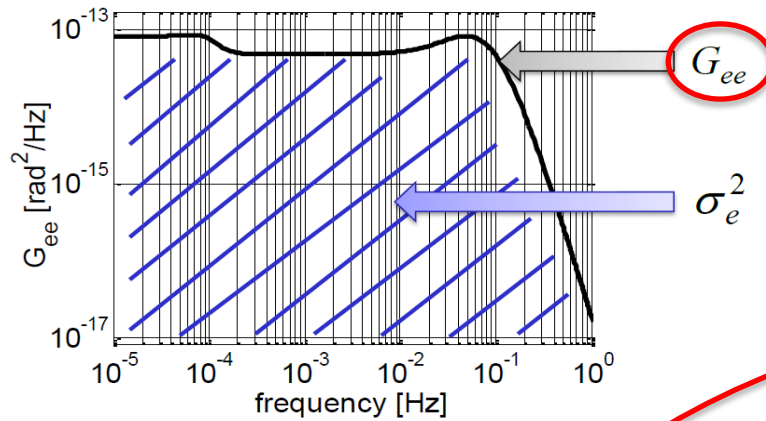
Mean error:  
MPE, MKE



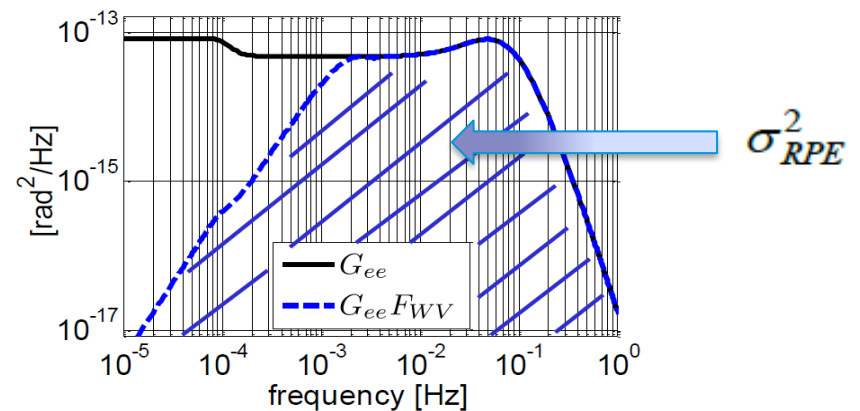
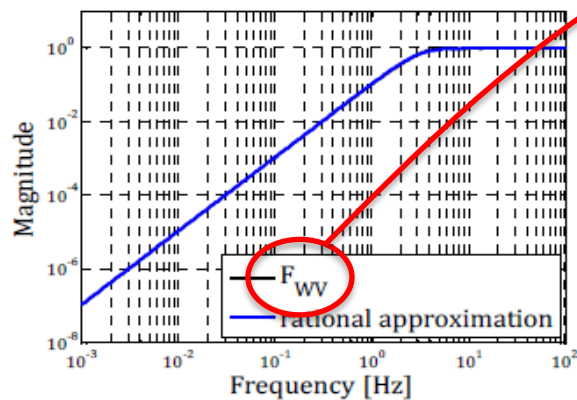
Relative error:  
RPE, RKE

# AST-3: Error Index Contribution Analysis (3/3)

● Example: GSE noise contribution to RKE



$$\sigma_{RPE}^2(\Delta t) = \frac{1}{2\pi} \int_0^{\infty} G_{ee} F_{WV}(\Delta t) d\omega$$





# AST-4: Pointing Error Evaluation (1/3)

- The **simplified statistical method** is treated in detail in the PEEH
- In the simplified statistical method, time-constant and time-random error contributors (per index) are first summed separately

Time-Constant PEC: Summation and Level of Confidence Evaluation				
$P_c$ evaluation		summation		
$B \leq n_p \times \sigma_B +  \mu_B $		$\mu_B = \sum_{i=1}^N \mu_i$		
		$\sigma_B^2 \leq \sigma_c^2 + \sigma_{uc}^2$	<i>correlated</i>	$\sigma_c^2 \leq N \sum_{i=1}^N \sigma_i^2$ (*1) or $\sigma_c^2 \leq \left[ \sum_{i=1}^N \sigma_i \right]^2$ (*2)
			<i>uncorrelated</i>	$\sigma_{uc}^2 = \sum_{i=1}^N \sigma_i^2$
$B$	<i>bias</i>	$P_c$	<i>level of confidence:</i>	
$\sigma^2$	<i>variance</i>	$n_p$	<i>defines max. error complying with <math>P_c</math> requirement of a Gaussian distribution by <math>n_p \times \sigma</math></i>	
$\mu$	<i>mean</i>			
$N$	<i>number of summands</i>			
*1: general worst case upper bound summation (see Annex for derivation). *2: upper bound summation as in ECSS standard.				

# AST-4: Pointing Error Evaluation (2/3)

- Means are summed linearly
- Uncorrelated variances are RSSed
- Correlated variances summation is estimated by upper bound
- Formulas are valid under the assumption that the central limit theorem applies

TIME-RANDOM PEC: SUMMATION			
statistics	summation		
$\mu_{index\_sum}$	$= \sum_{i=1}^N \mu_i$		
$\sigma_{index\_sum}^2$	<i>group summation</i>	<i>signal class groups</i>	
	$\leq \sum_{i=1}^N (\sigma_{uc}^2)_i + \sum_{i=1}^N (\sigma_c^2)_i$	correlated	$\sigma_c^2 \leq N \sum_{i=1}^N \sigma_i^2$ (*1) or $\sigma_c^2 \leq \left[ \sum_{i=1}^N \sigma_i \right]^2$ (*2)
		uncorrelated	$\sigma_{uc}^2 = \sum_{i=1}^N \sigma_i^2$
$\sigma^2$	<i>variance</i>	$N$	<i>number of summands</i>
$\mu$	<i>mean</i>		
*1: general worst case upper bound summation (see Annex for derivation).			
*2: upper bound summation as in ECSS standard [ ] suggested for periodic error summation.			

# AST-4: Pointing Error Evaluation (3/3)

- The applicable confidence level specified in the requirements is then incorporated in the error index evaluation
- Total pointing error per index is finally obtained as sum of time-constant (only for APE, AKE, MPE, MKE) and time-random partial sums

Time-Random PEC: Level of Confidence Evaluation		
<i>index</i>	<i>P<sub>c</sub> evaluation assuming applicability of central limit theorem</i>	<i>P<sub>c</sub> requirement</i>
<i>APE(Δt<sub>D</sub>)/AKE(Δt<sub>D</sub>)</i>	$\epsilon_{index} = n_p \cdot \sigma_{index,sum}(\Delta t_D) +  \mu_{index} $	$\text{Prob}( e  < \epsilon_{index}) \geq P_c$
<i>MPE(Δt, Δt<sub>D</sub>)/MKE(Δt, Δt<sub>D</sub>)</i>	$\epsilon_{index} = n_p \cdot \sigma_{index,sum}(\Delta t, \Delta t_D) +  \mu_{index} $	
<i>RPE(Δt, Δt<sub>D</sub>)/RKE(Δt, Δt<sub>D</sub>)</i>	$\epsilon_{index} = n_p \cdot \sigma_{index,sum}(\Delta t, \Delta t_D)$	
<i>PDE(Δt, Δt<sub>s</sub>, Δt<sub>D</sub>)/KDE(Δt, Δt<sub>s</sub>, Δt<sub>D</sub>)</i>	$\epsilon_{index} = n_p \cdot \sigma_{index,sum}(\Delta t, \Delta t_s, \Delta t_D)$	
<i>PRE(Δt, Δt<sub>s</sub>)/KRE(Δt, Δt<sub>s</sub>)</i>	$\epsilon_{index} = n_p \cdot \sigma_{index,sum}(\Delta t, \Delta t_s)$	
<i>e</i>	<i>pointing error</i>	<i>P<sub>c</sub> level of confidence</i>
<i>ε<sub>index</sub></i>	<i>max. time-random error compliant with P<sub>c</sub></i>	<i>n<sub>p</sub> defines error that complies with P<sub>c</sub> requirement of a Gaussian distribution by n<sub>p</sub> × σ<sub>index</sub></i>
<i>μ<sub>index</sub></i>	<i>mean of time-random error</i>	
<i>Δt<sub>D</sub></i>	<i>drift re-set time</i>	

# Pointing Error Engineering Tool (PEET)

- With the aim to complement and to ease the application in daily work of the methodology outlined in the ESA PEEH and the ECSS standard E-ST-60-10C, an accompanying SW tool called PEET has been developed
- Development timeline:
  - 2012 - Prototype of the accompanying SW tool PEET completed (see [peet.estec.esa.int](http://peet.estec.esa.int))
  - 2013 – Prototype update for positioning and relative pointing
  - 2017 – Fully functional operational version of the accompanying SW tool PEET completed
- Scope of PEET:
  - Translation of the engineering methodology of [ESSB-HB-E-003] into a software tool that supports system engineers
  - General error propagation and analysis tool for the assessment of any type of quantity
  - Systematic compilation of budgets for any mission including high precision pointing missions
  - Common interface for model exchange between departments and users

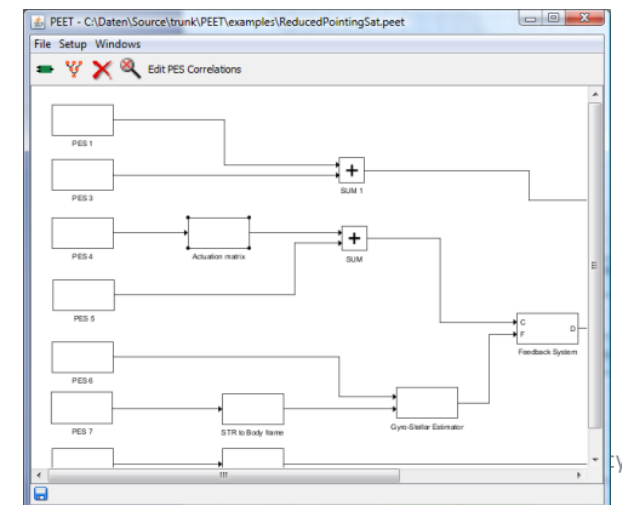
- The ESA PEEH together with the ECSS standard E-ST-60-10C provides all the **mathematical elements to carry out pointing error budgeting** activities
- The exact capture of window time and stability properties that is allowed by the **frequency domain methodology** described by PEEH limits the risk of over- or under-estimates that is inherent in the simplified summation rules
- The frequency domain approach also makes the PEEH methodology a useful tool to complement **the tuning activities** on system transfer functions (typically attitude estimator, attitude controller, etc.)
- The PEEH methodology can be used to assist **requirement breakdown** and allocation activities by providing sensitivity evaluation of the pointing performance with respect to the PES
- PEEH has become **an [AD] in ESA ITTs** in association with the ECSS Control Performance Standard E-ST-60-10C

# BACK-UP SLIDES

## on PEET

# PEET Software Facts

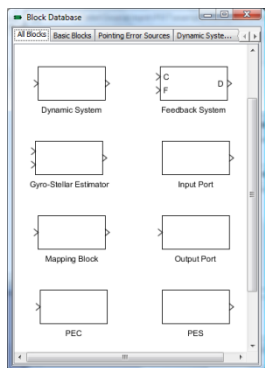
- **Open-source** software available free of charge to industry and institutions based in the ESA member states
- Software realized as a **Matlab toolbox**
  - but no dedicated Matlab knowledge required.
  - analysis algorithms coded in Matlab using well-proven standard functions.
- Dedicated **graphical user interface**
  - tailored to engineering methodology
  - intuitive Simulink-like system editor with drag & drop functionality for a fast and flexible setup
  - extendable block database with generic and specific models
- **Interfaces**
  - Data import from Matlab workspace and MS Excel
  - Data export and reporting functionality to MS Excel



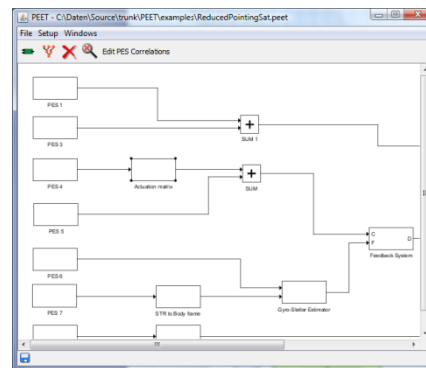
# Model based pointing error engineering

PEET guides the user in applying the rules defined in [ESSB-HB-E-003] and [ECSS-E-ST-60-10C]. The workflow is fully compatible with the methodology defined in [ESSB-HB-E-003]:

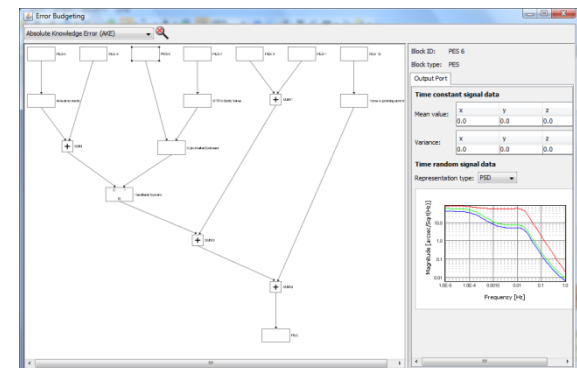
1. Characterize and define pointing error sources using the Block Database (AST-1)
2. Setup the pointing error system within the System Editor of PEET using blocks from the model database (AST-2)
3. Analyse the pointing error budget using the Tree View of PEET (AST-3 & AST-4)



Block Database



System Editor



Tree View



# Limitations

- PEET is not a substitute for E2E simulators. It cannot cover:
  - **Non-linearities** as the transfer analysis is based on LTI systems
  - **Transient** signals may be covered by the new tool update (with Fourier series modelling) but if not, they have to be analysed by other means, e.g. time-domain simulations (
  - Control design issues as no validity checks are performed concerning the system **stability**
- The initial classification of error sources cannot be generalized and remains an engineering task.

# Sensitivity analyses and system optimization

- PEET provides a built-in analysis functionality to assess the sensitivity of error source and system transfer parameters on the final error
- Parameter **trade-off studies** and system optimization are supported:
  - implicitly, as **single parameters** (error sources, transfer system,...) and even global settings (statistical interpretation, pointing error index) can easily be **modified using the GUI**
  - explicitly, as the software provides **batch mode capabilities**, i.e. the possibility of script-based execution without using the GUI.

# Pointing budget break-down

- Multi-level budgets are inherently supported by proper grouping of subsystems and **error evaluations** at different “nodes” of the tree.
- An explicit feature allows automatic consistency checks of broken down requirement values and display of requirement violations after the evaluation.

