The Pointing Error Engineering Tool (PEET): From Prototype to Release Version

Marc Hirth\textsuperscript{(1)}, Haifeng Su\textsuperscript{(1)}, Thomas Ott\textsuperscript{(2)}, Massimo Casasco\textsuperscript{(3)}, Sohrab Salehi\textsuperscript{(4)}

\textsuperscript{(1)} Astos Solutions GmbH
\textsuperscript{(2)} Airbus DS – Space Systems
\textsuperscript{(3)} ESA, ESTEC
\textsuperscript{(4)} Rhea System for ESA
Outline

• Introduction

• PEEH Methodology in the Tool

• Software Overview

• “Advanced Statistical Method”

• Conclusion
Introduction
History

• 2008: Control performance standard ECSS-ST-60-10C

• 2011: ESA Pointing Error Engineering Handbook (PEEH)

• 2012: PEET prototype released

• 2013: Prototype update for formation-flying missions

• 2014-2016: Development of extended framework
Motivation

More and more stringent performance requirements in ESA projects (e.g. scientific or laser-communication missions)

• Necessity for clear and accurate pointing error engineering methodology

Why PEEH?

• Systematic and user-friendly application of methodology
• Automated performance management process
• Replacement of “manual” computations
• Support dissemination of the methodology

Why PEET?
PEEH Methodology in the Tool
PEEH Methodology

PEEH provides explicit guideline for requirement definition and evaluation of error budgets using dedicated analysis steps:

- **“AST-0”**: Requirement specification
- **AST-1**: Error source characterization
- **AST-2**: Transfer analysis
- **AST-3**: Error index contribution analysis
- **AST-4**: Error evaluation

The PEET workflow is fully compatible with the methodology and provides all related parameters and setup options.
AST-0: Requirement specification

Unambiguous definition of requirements by parameters dependent on requirement type

• Statistical requirements
  • Max. error value (per axis or half-cone)
  • Related level of confidence
  • Metric for time-windowed errors
  • Statistical interpretation

• Spectral requirements
  • Spectral requirement function (PSD upper bound)
  • (Metric for time-windowed errors)
Identification of potential error sources and classification based on characteristic properties:

- Random variables
  - Time-constant & Time-random
- Periodic errors
- Drift errors
- Random processes
  - BLWN or PSD

Statistical distributions describe temporal behavior and/or ensemble distribution of parameters
Determines how initial error sources affect the figure of merit, i.e. the “route” to the final error contribution

- Static systems
  - Generic models
  - Coordinate Transformations, etc.
- Dynamic (LTI) systems
  - Generic models
  - Flexible plant, gyro-stellar estimator
- Feedback systems

Transfer rules dependent on error signal class
AST-3: Error Index Contribution Analysis

Impact of time-windowed errors for all metrics present in ECSS standard (e.g. APE, MPE):

- Frequency domain evaluation
  - Metric expressed as rational transfer functions
  - Applied to periodic and random process errors
- Evaluation according to ECSS rules
  - Random variables

### Table B-5: Budget contributions from zero mean Gaussian random errors

<table>
<thead>
<tr>
<th>Index</th>
<th>S.I.</th>
<th>Distribution</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>APE</td>
<td>E</td>
<td>$P(3\mu) = 3\mu$</td>
<td>$\mu(e)$, $\sigma(e)$</td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>$G(0, \text{swc})$</td>
<td>$\text{swc}$</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>$\int G(0,\sigma)^2 P(s) ds$</td>
<td>$\langle s \rangle$</td>
</tr>
</tbody>
</table>

### Windowed Mean Stability (WMS)

$$F_{\text{WMS}}(\omega, \Delta t, \Delta t_i) = F_{\text{STA}} \times F_{\text{RMS}}$$

$$F_{\text{RMS}}(\omega, \Delta t, \Delta t_i) = \frac{F_{\text{STA}} \times F_{\text{RMS}}}{4(1 - \cos(\omega \Delta t)^2)}$$

**rational approximation:**

$$\tilde{F}_{\text{RMS}}(\omega, \Delta t, \Delta t_i) = \tilde{F}_{\text{STA}} \times \tilde{F}_{\text{RMS}}$$

**$F_{\text{WMS}}$ with $\Delta t = 0.18s$ and $t_i = 60s$**

---

© 2016 Astos Solutions GmbH

ICATT 2016, Darmstadt, Germany
AST-4: Error evaluation

Summation of final error contributions according to the selected statistical method:

• Simplified statistical method (used in prototype)
  • Evaluation based on stat. moments (central limit theorem) and a confidence factor
  • Simplified correlation (full/no correlation)

\[ e(68\% / 95\% / 99.7\%) = \left( \sum |\mu_i| + n_p \cdot \sum \sigma_i \right), n_p = 1,2,3 \]

• Advanced statistical method (used in release version)
  • Evaluation based on probability density functions & explicit level of confidence values
  • Specific correlation between contributions

\[ p(e) = \left( \sum_{i=1}^{N_D} e_i \right) \quad e_{tot,LoC} = \int_0^{LoC/100} p(e) \, de \]
Software Overview
Platforms and Requirements

- Designed as extension to Matlab with following requirements:
  - Matlab 2011b and later
  - Control System Toolbox

- Multiple platform support
  - Windows
  - Linux
  - Macintosh

- No “critical” hardware requirements
  - Runs on standard desktop PCs and laptops
Architecture and External Interfaces

• Main components:
  • Java GUI
  • Matlab core algorithms
  • Communication via Java-Matlab Interface (JMI) and XML scenario definition files

• Interfaces:
  • Import from MS Excel spreadsheets
  • Links to Matlab workspace variables
  • Configurable report to MS Excel

• Operation modes:
  • GUI mode
  • Script-base execution controlled by (user-defined) Matlab scripts
Graphical User Interface

Block Database: Select identified error source and system models
Graphical User Interface (cont’d)

System Editor: Define requirements & system interconnections
Graphical User Interface (cont’d)

Block Dialogs: Specify source/system characteristics

- Custom or default units
- User-defined block description
- Quick-help (mouse-over)
Graphical User Interface (cont’d)

Budget Tree View: Analyze statistics of error signal components
Graphical User Interface (cont’d)

Breakdown Tree View: Analyze compliance with requirements
Graphical User Interface (cont’d)

Plot Viewer: Detailed result inspection with various plot types

- PDF plots
- CDF plots
- Cumulated variance (of random process)
- Correlation “scatter“ plots
- PSD plots (auto- and cross-spectra)
“Advanced Statistical Method”
Background

Simplified statistical method used in software prototype:

- Based on statistical moments ($\mu$, $\sigma$) and confidence factors only
- Relies on applicability of central limit theorem

For a valid level of confidence evaluation, total error needs to follow (at least nearly) a Gaussian distribution

If dominant non-Gaussian errors are present, this leads to significant systematic errors
Limitations of the Simplified Method (1)

Example:
„The error shall be smaller than X with 99.73% (3\(\sigma\)) probability“ (applied to a Gaussian and a uniform distribution)

\[ e = 3\sigma = 3\sigma_G \]

correct “by definition”

\[ e = 3\sigma = 3\sigma_U = \sqrt{3} \cdot b \]

Correct value: 0.9973 \(b\)

\(\approx 80\%\) systematic error

Even exceeds given bounds
Limitations of the Simplified Method (2)

Calculation of line-of-sight (LoS) errors:

• PEEH provides derived expression for instantaneous (or deterministic) LoS errors

\[ e_{\text{LOS}} = \sqrt{e_x^2 + e_y^2} \]

• ECSS provides approximate solution for „statistical“ errors (valid for zero-mean Gaussian with closely equal \( \sigma \))

\[ e_{\text{LOS}} = \max(\sigma_x, \sigma_y) \sqrt{-2 \log(1 - p_c)} \]

• Exact description via PDF

\[ e_{\text{LOS}} = \int_0^{\text{LoC}} p \left( \sqrt{e_x^2 + e_y^2} \right) de \]

“Careless” application of the first expressions to non-matching conditions again leads to systematic errors
**Limitations of the Simplified Method (3)**

Calculation of line-of-sight (LoS) errors: 68.3% LoC

<table>
<thead>
<tr>
<th>Case</th>
<th>$e_{\text{LoS}}$</th>
<th>PDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>X: $\mathcal{G}(0,1)$</td>
<td>Y: $\mathcal{G}(0,1)$</td>
<td>“Deterministic”</td>
</tr>
<tr>
<td>-7%</td>
<td>0%</td>
<td>1.5158</td>
</tr>
<tr>
<td>X: $\mathcal{G}(1,1)$</td>
<td>Y: $\mathcal{G}(1,1)$</td>
<td>30%</td>
</tr>
<tr>
<td>X: $\mathcal{G}(0,1)$</td>
<td>Y: $\mathcal{G}(0,2)$</td>
<td>-22%</td>
</tr>
<tr>
<td>X: $\mathcal{U}(-\sqrt{3}, \sqrt{3})$</td>
<td>Y: $\mathcal{G}(0,1)$</td>
<td>-10%</td>
</tr>
</tbody>
</table>
Implementation (1)

Analytical treatment (PDF convolution and CDF determination) in conflict with several constraints:

• SW requirement: No further MATLAB toolboxes shall be used (e.g. the Symbolic Toolbox)
• Even with symbolic computation, closed-form solutions could not be guaranteed (for arbitrarily complex systems)
• With a numerical description of the PDF convolution, the joint PDF of all error sources is required
  • Usually not known by the user
  • In best case, knowledge of (marginal) PDFs and correlation in terms of correlations coefficients expected
Implementation (2)

Therefore, the chosen approach is entirely numerical:

- Sample-based (around $1\times10^6$ samples per source, 10000 PDF bins)
- Dedicated inverse transform sampling method allows both „imprinting“ correlation and PDF information

\[\text{see paper for more information}\]

- Intrinsic drawback: loss of accuracy with respect to the analytical computation
  - Error expected to $< 1\%$ in entire computation chain
  - Thus safely negligible compared to potentially large systematic errors of simplified method
Conclusion
Conclusion

• PEET is a tool to accurately compute statistical and spectral error budgets (release: mid 2016).

• PEET is not restricted to pointing applications, but can also be used in other engineering fields.

• PEET is well-suited for integration in analysis tool chains with the available Excel & Matlab interfaces.

• PEET files provide much better transparency (model assumptions) and flexibility (model adaption) than purely “tabular” budgets.

• Limitation: PEET cannot explicitly account for non-linearities and transient system behavior, i.e. it shall not be understood as a replacement for E2E simulators.