EMRP
European Metrology Research Programme
Programme of EURAMET

The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union

NPL REPORT TQE 10

Guidance for the use of compact covariance uncertainty software

HUMPHREYS, D A and HARRIS, P M

JULY 2014

National Measurement System

Guidance for the use of compact covariance uncertainty software

Humphreys, D A and Harris, P M
Time Quantum & Electromagnetics Division

ABSTRACT

This User Guide describes the operation of NPL's compact covariance uncertainty software and provides examples to illustrate the use of the software. The objective of the software is to provide a compact representation of the large covariance matrices associated with waveform measurements. The software also allows the propagation of the covariance matrix defined by various algebraic manipulations of those waveform measurements

ISSN 1754-2995

National Physical Laboratory
Hampton Road, Teddington, Middlesex, TW11 0LW

Approved on behalf of NPLML by Dr A J A Smith, Group Leader.

EXECTIVE SUMMARY

This User Guide outlines the covariance matrix compression algorithm implemented in NPL's compact covariance uncertainty software, developed under workpackage WP2 of Euramet Joint Research Project IND16 "Metrology for ultrafast electronics and high-speed communications" (Ultrafast).

The motivation for the work was to provide an approach whereby a covariance matrix can be stored in a format that grows proportional to the length of a measured waveform rather than the square of the length, and to allow uncertainties to be propagated between the time and frequency domains. In addition, modules have been developed to allow propagation of the covariance matrix defined by various algebraic manipulations of the waveform measurements so that impedance match corrections and calibration corrections can be applied.

The software is written in Matlab and this User Guide provides an outline of the definitions used and the software elements, together with examples. A separate software package based on these algorithms has been written in Python. No warranty or guarantee applies to the software described in this document, and therefore any users should satisfy themselves that it meets their requirements.

The software packages developed under Euramet Joint Research Project IND16 are available free of charge. To obtain a copy of the software please follow the registration instructions on the project website www.ptb.de/emrp/ultrafast.html .

CONTENTS

EXECUTIVE SUMMARY

1 INTRODUCTION

This User Guide is part of Deliverable D2.1 of Euramet project IND16 “Metrology for ultrafast electronics and high-speed communications” (Ultrafast) [1]. The purpose of this guide is to allow the covariance matrix compression algorithm implemented in NPL’s compact covariance uncertainty software to be evaluated. No warranty or guarantee applies to the software described in this document, and therefore any users should satisfy themselves that it meets their requirements. Full details of the compact covariance algorithm are available in Ref. [2].

2 MOTIVATION

Measurement uncertainties associated with a waveform measured in the time or frequency domain can be represented by a covariance matrix. The covariance matrix for a waveform measured in the frequency domain relates to the real and imaginary parts of the frequency components of the waveform and all the correlation relationships are captured in the matrix. The problem is that the size of the covariance matrix grows as the square of the trace length. Digital Sampling Oscilloscopes (DSOs) are the main instruments used to disseminate from the primary standard and typically provide waveforms with lengths of up to 5 000 points, yielding large, but manageable, covariance matrices. However, Real-Time Digital Oscilloscopes (RTDOs) can easily acquire waveforms with lengths of greater than 10^6 points giving impractically large covariance matrices that are difficult to store and manipulate. The compact covariance approach will allow the full-waveform calibration and uncertainties to be transferred to the user and applied to time-domain nor frequency-domain measurements.

The motivation for this work is to provide an approach whereby a covariance matrix can be stored in a format that grows proportional to the length of a measured waveform (rather than the square of the length) to allow practical dissemination and use in industry. The structure of the covariance matrix associated with waveform measurements provided by different instrument types has been assessed using measured and simulated data[3]. We analysed the structures present and proposed a strategy to store the matrix in compact form.

3 OVERVIEW

3.1 PURPOSE OF THE SOFTWARE

The purpose of the compact covariance uncertainty software is to provide a high-level language prototype implementation of the compression algorithm. The software is written in Matlab and made available to the project team for evaluation purposes. By writing in a commonly available language, the algorithm can be easily implemented and evaluated within other programs and languages.

3.2 SCOPE OF THE COMPRESSION ALGORITHM

The compression algorithm is designed for use with DSO or RTDO calibration waveforms and is applied in the frequency domain. The data is assumed to be real in the time-domain and is represented from the 1st harmonic to the n^{th} harmonic in the frequency domain, where n can correspond to the Nyquist limit (typically the case for a RTDO) or an arbitrary frequency (typically the case for a DSO). However, there is no reason why the algorithm could not be adapted for use with complex modulation data, in which case the frequency range of interest would extend from $-f_n$ to f_n, as opposed to f_1 to f_n.

3.3 FUNCTIONALITY OF THE SOFTWARE

The software performs the following functions:

1. Create a compact representation of a covariance matrix from measured frequency-domain waveform data:
    a. The mean components of *m* frequency-domain waveform measurements and the covariance matrix associated with the mean components are calculated;
    b. An approximation to the covariance matrix that has a compact representation is determined, and the mean components and the compact representation are stored;
    c. A test of the positive-definiteness of the approximation to the covariance matrix is undertaken, and the degrees of approximation and compression are reported.
2. Form a covariance matrix from its compact representation.
3. Perform propagation of covariance matrices stored in compact form defined by the operations of concatenation, de-embedding, addition and subtraction of two or more frequency-domain waveforms and return a compactly-represented covariance matrix for the result. The objective is to allow measurement results to be corrected for mismatch and the primary standard impulse response.
4. Evaluate the amplitude and phase response from a frequency-domain waveform and the associated standard uncertainties.
5. Evaluate a time-domain representation of a frequency-domain waveform and the associated standard uncertainties.
6. Perform propagation of covariance matrices stored in compact form defined by transformations of waveform data from the frequency-domain to the time-domain and vice-versa for evenly spaced time and frequency points.

3.4 BUG REPORTING

Please report errors (reproducible failures) arising with this software to:

Name	email	Telephone
D A Humphreys	david.humphreys@npl.co.uk	+44 208 943 6389
P M Harris	peter.harris@npl.co.uk	+44 208 943 6961

Please record the following information:

- The version of Matlab you are running;
- The release date/version of the modules;
- The nature of the failure;
- The raw input data;
- Details of the hardware (platform, operating system, memory size).

4 DEFINITIONS

4.1 COVARIANCE MATRIX

The covariance matrix is assumed to be associated with the mean or average of a set of m measured frequency-domain waveform data obtained independently. Let $\boldsymbol{X}$ denote the matrix whose m columns contain the (real and imaginary parts of the) frequency-domain waveform data, and $\boldsymbol{X}'$ the matrix obtained by correcting $\boldsymbol{X}$ for the means over all m measurements, i.e., the mean of the elements in the jth row is subtracted from all elements in that row. Then, the covariance matrix $\boldsymbol{V}$ is defined as

$$\boldsymbol{V} = \frac{1}{m}\boldsymbol{S},$$

where $\boldsymbol{S}$ is the sample covariance matrix defined by

$$\boldsymbol{S} = \frac{1}{m-1}\boldsymbol{X}'(\boldsymbol{X}')^{\mathrm{T}}.$$

4.2 FOURIER TRANSFORMS

There are different definitions of the (discrete) Fourier transform and its inverse that are in use in commercial software packages and instrumentation. They differ in terms of scaling and exponent sign.

Let $x_i, i = 1, \ldots, N$, denote a measured time-domain waveform and $X_j = a_j + \mathrm{j}b_j, j = 0, \ldots, n$, the corresponding frequency-domain waveform with $\mathrm{j}^2 = -1$ and $n = \lfloor N/2 \rfloor$, the integer part of $N/2$. Since the measured time-domain data is real-valued, it follows that $b_0 = 0$ (always), and $X_0 = a_0$ is the "DC-level" of the waveform. Furthermore, if N is even then $b_n = 0$ also, otherwise X_n is complex-valued.

Define angles

$$\theta_i = \frac{2\pi(i-1)}{N}, \qquad i = 1, \ldots, N,$$

corresponding to the time-domain measured values. By evaluating

$$a_0 = \frac{1}{N}\sum_{i=1}^{N} x_i,$$

$$a_j = \frac{2}{N}\sum_{i=1}^{N} x_i \cos(j\theta_i)\ , \qquad b_j = \frac{2}{N}\sum_{i=1}^{N} x_i \sin(j\theta_i)\,, \qquad j = 1, \ldots, n,$$

and, when N is even,

$$a_n = \frac{1}{N}\sum_{i=1}^{N} x_i \cos(n\theta_i)\ , \qquad b_n = 0,$$

it follows that

$$x_i = a_0 + \sum_{j=1}^{n} a_j \cos(j\theta_i) + \sum_{j=1}^{n} b_j \sin(j\theta_i), \qquad i = 1, \dots, N.$$

In these definitions, scaling is applied in the calculation of the frequency-domain waveform. An advantage is that the inverse discrete Fourier transform can be performed without knowledge of N (and whether it is even or odd).

In practice, the available frequency-domain data takes the form, for each measured waveform, of the frequencies $f_j, j = 0, \dots, n$, and the real and imaginary parts $a_j, b_j, j = 0, \dots, n$, of each frequency component. Then, the frequency-spacing is

$$\Delta f = f_1,$$

the sampling frequency f_s is

$$f_s = 2f_n = 2n\Delta f,$$

the time-spacing is

$$\Delta t = \frac{1}{f_s},$$

and the number of time-points is

$$N = \begin{cases} 2n, & b_n = 0, \\ 2n+1, & b_n \neq 0. \end{cases}$$

For comparison, for N odd, the Matlab command "fft($\boldsymbol{x}$)" returns the frequency-domain data:

$$\begin{bmatrix} a_0 N \\ (a_1 - jb_1)N/2 \\ \vdots \\ \vdots \\ (a_n - jb_n)N/2 \\ (a_n + jb_n)N/2 \\ \vdots \\ \vdots \\ (a_1 + jb_1)N/2 \end{bmatrix},$$

and, for N even, it returns

$$\begin{bmatrix} a_0 N \\ (a_1 - jb_1)N/2 \\ \vdots \\ \vdots \\ (a_{n-1} - jb_{n-1})N/2 \\ a_n N \\ (a_{n-1} + jb_{n-1})N/2 \\ \vdots \\ \vdots \\ (a_1 + jb_1)N/2 \end{bmatrix}.$$

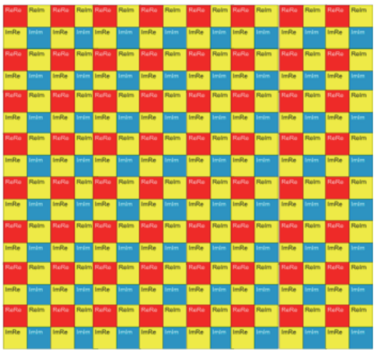
4.3 COVARIANCE MATRIX STRUCTURE

The covariance matrix is assumed to relate to frequency-domain waveform measurements. The reason for this is that time errors in the results reported by instruments may need correction prior to conversion to the frequency-domain. Timebase nonlinearity is a correctable effect for sampling oscilloscopes, and for a real-time oscilloscope the trigger time error between traces can be as high as $\pm\Delta t/2$, where Δt is the sampling interval, assuming that there are no other trigger jitter mechanisms. In addition, the clock accuracy of a real-time oscilloscope will introduce a slow timing error that differs on a "per trace" basis. For sampling oscilloscope timebase correction it is essential to achieve accurate results, and for real-time oscilloscopes averaging in the frequency domain with time alignment correction gives a better result than direct time-domain averaging.

Traditionally the covariance matrix structure for complex data interleaves the real and imaginary terms at each frequency:

$$\boldsymbol{V} = \begin{bmatrix} u^2(a_1) & u(a_1,b_1) & \cdots & u(a_1,a_n) & u(a_1,b_n) \\ u(b_1,a_1) & u^2(b_1) & & u(b_1,a_n) & u(b_1,b_n) \\ & \vdots & \ddots & \vdots & \\ u(a_n,a_1) & u(a_n,b_1) & \cdots & u^2(a_n) & u(a_n,b_n) \\ u(b_n,a_1) & u(b_n,b_1) & & u(b_n,a_n) & u^2(b_n) \end{bmatrix}.$$

Figure 1. Illustrates that the detailed evolution of the real, imaginary and cross-components with frequency will be obscured in this representation.



ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm
ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm
ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm
ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm
ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm
ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm
ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm
ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm
ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm
ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm
ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm
ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm
ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm
ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm
ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm	ReRe	ReIm
ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm	ImRe	ImIm

Figure 1 Traditional covariance matrix representation.

In the proposed algorithm, the real and imaginary parts of the frequency components are grouped together as this gives greater visibility to any underlying structure:

$$\boldsymbol{V} = \begin{bmatrix} u^2(a_1) & \cdots & u(a_1,a_n) & u(a_1,b_1) & \cdots & u(a_1,b_n) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u(a_n,a_1) & \cdots & u^2(a_n) & u(a_n,b_1) & \cdots & u(a_n,b_n) \\ u(b_1,a_1) & \cdots & u(b_1,a_n) & u^2(b_1) & \cdots & u(b_1,b_n) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u(b_n,a_1) & \cdots & u(b_n,a_n) & u(b_n,b_1) & \cdots & u^2(b_n) \end{bmatrix}.$$

Figure 2. Shows that all the similar terms are grouped together, allowing the feasibility of interpolation to be evaluated. It is important to note that the eigenvalues of the covariance matrix are unaffected by this re-ordering. This is clearly visible from the analysed results in Figure 3 to Figure 5 and in Ref. 4.

ReRe	ReIm
ImRe	ImIm

Figure 2 Proposed covariance matrix representation.

4.4 CONCATENATED RESULTS

In a calibration or during a measurement there will be one or more de-embedding contributions, such as impedance match corrections. A calibration correction is applied to the measured results as a multiplication in the frequency domain or a convolution in the time domain. In the frequency domain,

$$C_i = A_i B_i,$$

where $A_i = a_{R,i} + \mathrm{j}a_{I,i}$ is a complex-valued component corresponding to the frequency f_i, and similarly for B_i and C_i. Since

$$c_{R,i} = a_{R,i} b_{R,i} - a_{I,i} b_{I,i}$$

and

$$c_{I,i} = a_{R,i} b_{I,i} + a_{I,i} b_{R,i},$$

it follows that the covariance matrix $V_{C,i}$ for C_i is given by

$$V_{C,i} = J_{A,i} V_{A,i} J_{A,i}^{\mathrm{T}} + J_{B,i} V_{B,i} J_{B,i}^{\mathrm{T}},$$

where $^{\mathrm{T}}$ denotes matrix transposition. Here, $V_{A,i}$, $V_{B,i}$ and $V_{C,i}$ are, respectively, the covariance matrices of dimension 2×2 for A_i, B_i and C_i, and $J_{A,i}$ and $J_{B,i}$ are sensitivity matrices given by

$$J_{A,i} = \begin{pmatrix} b_{R,i} & -b_{I,i} \\ b_{I,i} & b_{R,i} \end{pmatrix}$$

and

$$J_{B,i} = \begin{pmatrix} a_{R,i} & -a_{I,i} \\ a_{I,i} & a_{R,i} \end{pmatrix}.$$

The implication is that, provided the de-embedding covariance matrix has similar structural properties, the data storage requirement will grow as the number of stages times the record length.

5 GETTING STARTED

5.1 BACKROUND

In this section we describe the Matlab software, focussing on the modules that perform the main functions of the software. We also describe the input data files used by the software and the output results files generated by the software.

An example showing the use of the software is presented in section 7.

5.2 INSTALLATION AND UNINSTALLATION

There are no specific installation or uninstallation instructions. It is not necessary for the software and data and results files to exist within the same directory. It is necessary that the software is in the Matlab working directory or a directory included on the Matlab path. Please note that the software calls “xlsread” and requires a valid Excel licence.

5.3 MATLAB SOFTWARE

The software was developed using Matlab R2013a. The software modules are listed below:

Section	Module name	Function	Release date
6.1	CreateCompactCV_eig.m CreateCompactCV_eig_compact.m	Create a compact representation of a covariance matrix.	30/6/14
6.2	FormCovarianceMatrixFromCompactCV_eig.m	Form a covariance matrix from its compact representation.	30/6/14
6.3	FormFreqRespFromCompactCV_eig.m FormFreqRespFromCompactCV_eig_MC.m	Form the magnitude and phase of a frequency response and the associated uncertainties.	30/6/14
6.4	FormTimeSeriesFromCompactCV_eig.m	Form a time series from a frequency response and the associated uncertainties.	30/6/14
6.5	FormMultiplyFromCompactCV_eig.m	Multiply two frequency responses and evaluate the covariance matrix for the result (stored in compact form).	30/6/14
6.6	FormDivideFromCompactCV_eig.m	Divide two frequency responses and evaluate the covariance matrix for the result (stored in compact form).	30/6/14
5.10	FormAddFromCompactCV_eig.m	Add two frequency responses and evaluate the covariance matrix for the result (stored in compact form).	30/6/14

Section	Module name	Function	Release date
6.8	FormSubtractFromCompactCV_eig.m	Subtract two frequency responses and evaluate the covariance matrix for the result (stored in compact form).	30/6/14
6.9	FormConstantCompactCV_eig.m	Forms the compact representation of the covariance matrix for a user-supplied frequency response.	30/6/14

6 SOFTWARE ROUTINES

6.1 CREATECOMPACTCV_EIG.M

The module is called as follows:

```
RESULT = CreateCompactCV_eig(icontrol);
```

The module performs the following operations:

i. Load frequency-domain waveform data corresponding to repeated measurements (see section 6.11);
ii. Optionally, pre-process the frequency-domain waveform data to apply a correction to each set of data in order to align (approximately) the waveform measurements;
iii. Evaluate the average frequency-domain waveform in terms of the (corrected) frequency-domain data;
iv. Evaluate the full covariance matrix $\boldsymbol{V}$ in terms of the (corrected) frequency-domain data and the average waveform;
v. Evaluate the first ne_0 eigenvalues in an eigendecomposition of $\boldsymbol{V}$, and decide the number p of eigencomponents to be used to define a compact representation of $\boldsymbol{V}$;
vi. Store data defining the average frequency-domain transform and a compact representation of an approximation $\boldsymbol{V}_0$ to the associated covariance matrix in the output parameter *RESULT* and in an output results file (see section 6.12);
vii. Compare the full covariance matrix $\boldsymbol{V}$ with the approximate covariance matrix $\boldsymbol{V}_0$ constructed from its compact representation;
viii. Optionally, provide graphical visualisations of the two covariance matrices and their differences.

Notes:

- The input parameter *icontrol* is a vector of dimension 2×1, the first elements of which controls whether the frequency-domain waveform data is pre-processed at step ii, and the second controls whether the module provides graphical output at step viii. Each elements takes the value one (for true) or zero (for false). If the module is called with no input parameter, each element is assumed to take a default value of one.
- At step iv, only the values defining the “ReRe”, “ImIm” and “ImRe” blocks of the covariance matrix are stored.
- At step v, ne_0 is set to be the smaller of $2(n-1)$ and $m-1$, where n is the number of frequencies and m is the number of repeated measurements.
- At step v, p is selected as the number for which the values of the eigenvalues become essentially constant.
- At step v, Matlab’s “eigs” function is used to undertake the eigendecomposition.
- At step v, a graph is provided showing the first ne_0 eigenvalues and the choice of p.
- At step vi, the covariance matrix $\boldsymbol{V}$ has eigendecomposition

$$\boldsymbol{V} = \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}, \quad \boldsymbol{Q} = \boldsymbol{E}\boldsymbol{\Lambda}^{1/2},$$

where the matrix $\boldsymbol{E}$ of dimension $2n \times 2n$ contains the eigenvectors in its columns and the matrix $\boldsymbol{\Lambda}$ of dimension $2n \times 2n$ is a diagonal matrix containing the corresponding eigenvalues, which are assumed to be in non-increasing order. Since $\boldsymbol{V}$ is a symmetric, positive-definite matrix it follows that $\boldsymbol{\Lambda}_{2n,2n} > 0$, and the positive square roots are taken in the matrix decomposition. An approximation $\boldsymbol{V}_0$ to $\boldsymbol{V}$ that has a compact representation is given by

$$\boldsymbol{V}_0 = \boldsymbol{Q}_0\boldsymbol{Q}_0^{\mathrm{T}} + \boldsymbol{D},$$

where the matrix $\boldsymbol{Q}_0$ of dimension $2n \times p$ contains the first p columns of $\boldsymbol{Q}$, and the matrices $\boldsymbol{Q}_0$ and $\boldsymbol{D}$ of dimension $2n \times 2n$ have the forms

$$\boldsymbol{Q}_0 = \begin{pmatrix} \boldsymbol{Q}_{0,R} \\ \boldsymbol{Q}_{0,I} \end{pmatrix}, \qquad \boldsymbol{D} = \begin{pmatrix} \mathrm{diag}\{\boldsymbol{d}_1\} & \mathrm{diag}\{\boldsymbol{d}_2\} \\ \mathrm{diag}\{\boldsymbol{d}_2\} & \mathrm{diag}\{\boldsymbol{d}_3\} \end{pmatrix}$$

for vectors $\boldsymbol{d}_1, \boldsymbol{d}_2$ and $\boldsymbol{d}_3$ of dimension $n \times 1$. The matrix $\boldsymbol{D}$ is chosen such that

$$[\boldsymbol{V}]_{i,i} = [\boldsymbol{V}_0]_{i,i}, \qquad [\boldsymbol{V}]_{n+i,n+i} = [\boldsymbol{V}_0]_{n+i,n+i}, \quad [\boldsymbol{V}]_{i,n+i} = [\boldsymbol{V}_0]_{i,n+i},$$

for $i = 1, \dots, n$, i.e., the variances and covariances associated with the real and imaginary parts of each frequency component are preserved exactly in the approximation.

- At step vii, the following output is provided:
    - A measure of the compression achieved, i.e.,

      $$S = 100 \times \frac{2np + 3n}{(2n)^2};$$

    - A measure of the approximation, i.e.,

      $$E = 100 \times \sqrt{\frac{\sum_{ij}\left([\boldsymbol{V}]_{ij} - [\boldsymbol{V}_0]_{ij}\right)^2}{\sum_{ij}[\boldsymbol{V}]_{ij}^2}};$$

    - Information about whether the approximate covariance matrix is positive definite.
- At step viii, the following graphs are provided:
    - A graph showing as a surface plot the full covariance matrix $\boldsymbol{V}$;
    - A graph showing as a surface plot the covariance matrix $\boldsymbol{V}_0$ constructed from its compact representation;
    - A graph showing as a surface plot the differences between corresponding elements of $\boldsymbol{V}$ and $\boldsymbol{V}_0$.
- The output of the module is a single column array, called *RESULT*, which contains the information defining the average frequency-domain waveform and the associated covariance matrix $\boldsymbol{V}_0$ (see section 6.12).
- The function "CreateCompactCV_eig_compact.m" performs the same operations as "CreateCompactCV_eig.m" except that the full covariance matrix $\boldsymbol{V}$ is not explicitly formed, but its rows are stored in temporary files and accessed as necessary. The function is slow to execute, but can be used to generate the compact representation of an approximation to the covariance matrix determined from repeated measured frequency-domain waveform data for problems for which n and m are very large (and it is not possible to store the full covariance matrix).

6.2 FORMCOVARIANCEMATRIXFROMCOMPACTCV_EIG.M

The module is called as follows:

```
[DC, f0, Rzref, Izref, CVx] = …
        FormCovarianceMatrixFromCompactCV_eig(RESULT);
```

The module performs the following operations:

1. If the module is called with no input parameter, load from a previously created output results file data defining an average frequency-domain waveform and a compact representation of the associated covariance matrix;
2. Extract *DC* containing the DC component of the average frequency-domain waveform;
3. Extract f_0 defining the frequency spacing Δf for the frequency-domain waveform;
4. Extract data *Rzref* and *Izref* defining, respectively, the real and imaginary components of the average frequency-domain waveform;
5. Form *CVx*, the full covariance matrix $\boldsymbol{V}_0$ associated with the average frequency-domain waveform.

Notes:

- The module is used to unpack the compact representation of an average frequency-domain waveform and the associated covariance matrix.
- Optionally, the module may be called with an input parameter *RESULT* (see section 6.12).
- The outputs of the module are two scalars *DC* and f_0, two column arrays *Rzref* and *Izref* of length n, and an array *CVx* of dimension $2n \times 2n$.

6.3 FORMFREQRESPFROMCOMPACTCV_EIG.M

The module is called as follows:

```
[Aref, uAref, Pref, uPref] = …
      FormFreqRespFromCompactCV_eig(RESULT);
```

The module performs the following operations:

1. If the module is called with no input parameter, load from a previously created output results file data defining an average frequency-domain waveform and a compact representation of the associated covariance matrix;
2. For each frequency f_k, evaluate the amplitude $Aref_k$ of the average frequency-domain waveform corresponding to that frequency and the standard uncertainty $uAref_k$ associated with the amplitude;
3. For each frequency f_k, evaluate the phase $Pref_k$ of the average frequency-domain waveform corresponding to that frequency and the standard uncertainty $uPref_k$ associated with the phase;
4. Display the amplitude and phase responses with their associated standard uncertainties.

Notes:

- At steps ii and iii, no explicit construction of the covariance matrix $\boldsymbol{V}_0$ from its compact representation is undertaken.
- At steps ii and iii, the standard uncertainties are evaluated using the law of propagation of uncertainty, which is approximate for the calculations of amplitude and phase, but is undertaken in terms of the elements of $\boldsymbol{V}_0$ that reproduce exactly the corresponding elements of the full covariance matrix $\boldsymbol{V}$.
- At steps ii and iii, each standard uncertainty is returned as a “NaN” (not-a-number) in the case that the amplitude is zero.
- At step iv, the following output is provided:
    - A graph showing as a function of frequency the values of the amplitude of the average frequency-domain waveform;
    - A graph showing as a function of frequency the standard uncertainties associated with the values of amplitude;

  - A graph showing as a function of frequency the values of the phase of the average frequency-domain waveform;
  - A graph showing as a function of frequency the standard uncertainties associated with the values of phase.
- Optionally, the module may be called with an input parameter *RESULT* (see section 6.12).
- The outputs of the module are four column arrays of dimension $n \times n$.
- The function "FormFreqRespFromCompactCV_eig_MC.m" performs the same operations as "FormFreqRespfromCompactCV_eig.m" except that a Monte Carlo method is used to evaluate the amplitudes and phases and the associated standard uncertainties. By using a Monte Carlo method, the approximation inherent in applying the law of propagation of uncertainty is avoided, and more reliable results can be expected to be obtained.

6.4 FORMTIMESERIESFROMCOMPACTCV_EIG.M

The module is called as follows:

```
[t, tref, utref] = FormTimeSeriesFromCompactCV_eig(n0, s0,
    RESULT);
```

The module performs the following operations:

i. If the module is called without the input parameter *RESULT*, load from a previously created output results file data defining an average frequency-domain waveform and a compact representation of the associated covariance matrix;
ii. If the module is called with input parameter *n0* equal to zero, the number n_t of points is set according to the number n of harmonics and is even or odd according to whether *Izref*$_n$ is zero or not;
iii. Evaluate time samples t_k for $k = 1, \dots, n_t$;
iv. For each time sample t_k, evaluate the value *tref*$_k$ of the average time-domain waveform and the standard uncertainty *utref*$_k$ associated with the waveform value;
v. Display the average time-domain waveform and the associated standard uncertainties.

Notes:

- At steps ii and iii, if *n0* is equal to zero, then (cf. section 4.2)

$$n_t = \begin{cases} 2n, & Izref_n = 0, \\ 2n+1, & Izref_n \neq 0, \end{cases}$$

  and

$$t_k = (k-1)\Delta t, \qquad k = 1, \dots, n_t.$$

- At steps ii and iii, if *n0* is not equal to zero, then

$$n_t = n_0,$$

  and

$$t_k = \begin{cases} (k-1)\dfrac{L}{n_t}, & Izref_n = 0, \\ (k-1)\dfrac{L+\Delta t}{n_t}, & Izref_n \neq 0, \end{cases}$$

  where

$$L = \frac{1}{\Delta f}$$

is the duration of the measured waveform.

- At step iv, no explicit construction of the covariance matrix $\boldsymbol{V}_0$ from its compact representation is undertaken.
- At step iv, the definition of the inverse discrete Fourier transform (in terms of the sign of the exponent and the multiplying factor) is as described in section 4.2.
- At step iv, the angles in the evaluation of the inverse discrete Fourier transform are defined by (cf. section 4.2)

$$\theta_i = -\frac{2\pi s_0}{n_t} + \frac{2\pi(i-1)}{n_t}, \qquad i = 1, \dots, n_t,$$

to incorporate a user-defined phase-shift expressed in terms of a number *s0* of time-pojnts.

- At step iv, the standard uncertainties are evaluated using the law of propagation of uncertainty, which is exact for the calculation of the inverse discrete Fourier transform but is undertaken in terms of the approximation $\boldsymbol{V}_0$ to the full covariance matrix $\boldsymbol{V}$.
- The outputs of the module are three column arrays of dimension $n_t \times 1$ containing the results $tref_k$ and $utref_k$ at times t_k.

6.5 FORMMULTIPLYFROMCOMPACTCV_EIG.M

The module is called as follows:

```
RESULT = FormMultiplyFromCompactCV_eig;
```

The module performs the following operations:

i. Load from a previously created output results file data defining the first average frequency-domain waveform and a compact representation of the associated covariance matrix;
ii. Load from a previously created output results file data defining the second average frequency-domain waveform and a compact representation of the associated covariance matrix;
iii. Check the compatibility of the two average frequency-domain waveforms, which must be defined by the same number of frequency components and the same sampling frequency;
iv. For each frequency f_k, evaluate the product of the coefficients for the two average frequency-domain waveforms corresponding to that frequency;
v. Evaluate in a compact form the covariance matrix associated with coefficients defining the product of the average frequency-domain waveforms;
vi. Store data defining the product of the average frequency-domain waveforms and a compact representation of the associated covariance matrix in the output parameter *RESULT* and in an output results file (see section 6.12).

Notes:

- At step v, no explicit construction of the covariance matrices for the two average frequency-domain waveforms from their compact representations is undertaken.
- At step v, the elements of the covariance matrix are evaluated using the law of propagation of uncertainty, which is approximate for the calculation of the products of the coefficients of average frequency-domain waveforms.
- At step vi, the name of the file containing the data defining the product of the average frequency-domain waveforms and a compact representation of the associated covariance

matrix is of the form "file1 MULT file2.mat", where "file1.mat" and "file2.mat" are the names of the files containing the data defining the individual average frequency-domain waveforms.
- The output of the module is a single column array, called *RESULT*, which contains the information defining $\boldsymbol{V}_0$ (see section 6.12), to which the operations of calculating a frequency response, forming the time-domain representation, etc., may then be applied.

6.6 FORMDIVIDEFROMCOMPACTCV_EIG.M

The divide function in the frequency-domain performs a deconvolution in the time-domain. Deconvolution is an ill-posed problem, which will amplify the uncertainties at certain frequencies within the waveform. Weiner filtering has been included to increase stability, and we recommend that for general users this routine is used for impedance match correction only.

The module is called as follows:

```
RESULT = FormDivideFromCompactCV_eig(sig2);
```

The module performs the following operations:

i. Load from a previously created output results file data defining the first average frequency-domain waveform and a compact representation of the associated covariance matrix;
ii. Load from a previously created output results file data defining the second average frequency-domain waveform and a compact representation of the associated covariance matrix;
iii. Check the compatibility of the two average frequency-domain waveform, which must be defined by the same number of frequency components and the same sampling frequency;
iv. If the module is called with no input parameter, set the value of the input parameter *sig2* to be zero;
v. For each frequency f_k, evaluate the quotient of the coefficients for the two average frequency-domain waveforms corresponding to that frequency;
vi. Evaluate in a compact form the covariance matrix associated with coefficients defining the quotient of the average frequency-domain waveforms;
vii. Store data defining the quotient of the average frequency-domain waveforms and a compact representation of the associated covariance matrix in the output parameter *RESULT* and in an output results file (see section 6.12).

Notes:

- At step vi, no explicit construction of the covariance matrices for the two average frequency-domain waveforms from their compact representations is undertaken.
- At step vi, the elements of the covariance matrix are evaluated using the law of propagation of uncertainty, which is approximate for the calculation of the quotients of the coefficients of the average frequency-domain waveforms.
- At step vii, the name of the file containing the data defining the quotient of the average frequency-domain waveforms and a compact representation of the associated covariance matrix is of the form "file1 DIV file2.mat", where "file1.mat" and "file2.mat" are the names of the files containing the data defining the individual average frequency-domain waveforms.
- Optionally, the module may be called with an input parameter σ^2. Then, the quotient of two complex-valued coefficients z_1 and z_2 is evaluated as

$$\frac{z_1 \bar{z}_2}{z_2 \bar{z}_2 + \sigma^2},$$

where $\bar{z}$ is the complex conjugate of z.

- The output of the module is a single column array, called *RESULT*, which contains the information defining $\boldsymbol{V}_0$ (see section 6.12), to which the operations of calculating a frequency response, forming the time-domain representation, etc., may be applied.

6.7 FORMADDFROMCOMPACTCV_EIG.M

The module is called as follows:

```
RESULT = FormAddFromCompactCV_eig;
```

The module performs the following operations:

i. Load from a previously created output results file data defining the first average frequency-domain waveform and a compact representation of the associated covariance matrix;
ii. Load from a previously created output results file data defining the second average frequency-domain waveform and a compact representation of the associated covariance matrix;
iii. Check the compatibility of the two average frequency-domain waveforms, which must be defined by the same number of frequency components and the same sampling frequency;
iv. For each frequency f_k, evaluate the sum of the coefficients for the two average frequency-domain waveforms corresponding to that frequency;
v. Evaluate in compact form the covariance matrix associated with coefficients defining the sum of the average frequency-domain waveforms;
vi. Store data defining the sum of the transforms and a compact representation of the associated covariance matrix in the output parameter *RESULT* and in an output results file (see section 6.12).

Notes:

- At step v, no explicit construction of the covariance matrices for the two average frequency-domain waveforms from their compact representations is undertaken.
- At step v, the elements of the covariance matrix are evaluated using the law of propagation of uncertainty, which is exact for the calculation of the sums of the coefficients of the average frequency-domain waveforms.
- At step vi, the name of the file containing the data defining the sum of the average frequency-domain waveforms and a compact representation of the associated covariance matrix is of the form “file1 ADD file2.mat”, where “file1.mat” and “file2.mat” are the names of the files containing the data defining the individual average frequency-domain waveforms.
- The output of the module is a single column array, called *RESULT*, which contains the information defining $\boldsymbol{V}_0$ (see section 6.12), to which the operations of calculating a frequency response, forming the time-domain representation, etc., may be applied.

6.8 FORMSUBTRACTFROMCOMPACTCV_EIG.M

The module is called as follows:

```
RESULT = FormSubtractFromCompactCV_eig;
```

The module performs the following operations:

i. Load from a previously created output results file data defining the first average frequency-domain waveform and a compact representation of the associated covariance matrix;
ii. Load from a previously created output results file data defining the second average frequency-domain waveform and a compact representation of the associated covariance matrix;

iii. Check the compatibility of the two average frequency-domain waveforms, which must be defined by the same number of frequency components and the same sampling frequency;
iv. For each frequency f_k, evaluate the difference of the coefficients for the two average frequency-domain waveforms corresponding to that frequency;
v. Evaluate in compact form the covariance matrix associated with coefficients defining the difference of the average frequency-domain waveforms;
vi. Store data defining the difference of the average frequency-domain waveforms and a compact representation of the associated covariance matrix in the output parameter *RESULT* and in an output results file (see section 6.12).

Notes:

- At step v, no explicit construction of the covariance matrices for the two average frequency-domain waveforms from their compact representations is undertaken.
- At step v, the elements of the covariance matrix are evaluated using the law of propagation of uncertainty, which is exact for the calculation of the differences of the coefficients of the average frequency-domain waveforms.
- At step vi, the name of the file containing the data defining the sum of the average frequency-domain waveforms and a compact representation of the associated covariance matrix is of the form “file1 SUB file2.mat”, where “file1.mat” and “file2.mat” are the names of the files containing the data defining the individual average frequency-domain waveforms.
- The output of the module is a single column array, called *RESULT*, which contains the information defining $\boldsymbol{V}_0$ (see section 6.12), to which the operations of calculating a frequency response, forming the time-domain representation, etc., may be applied.

6.9 FORMCONSTANTCOMPACTCV_EIG.M

The module is called as follows:

```
RESULT = FormConstantCompactCV_eig(DC, f0, zref, uzref, n);
```

The module performs the following operations:

i. If the input includes vectors *zref* and *uzref* of complex-values:
    i) The number *n* of frequencies is defined by the length of the vector *zref* (or *uzref*);
    ii) The coefficients of the frequency-domain waveform are defined by the real and imaginary parts of *zref*;
    iii) The standard uncertainties associated with the coefficients are defined by the real and imaginary parts of *uzref*;
ii. If the input includes scalars *zref* and *uzref*, each a complex-value:
    i) The number *n* of frequencies is also an input to the module;
    ii) The real and imaginary parts of all the coefficients of the frequency-domain waveform are defined to be the real and imaginary pats of *zref*, respectively;
    iii) The standard uncertainties associated with the real and imaginary parts of all the coefficients are defined to be the real and imaginary parts of *uzref*, respectively;
iii. Create in compact form the covariance matrix associated with coefficients defining the frequency-domain waveform;
iv. With the additional input parameters *DC* and *f0*, store data defining the frequency-domain waveform and a compact representation of the associated covariance matrix in the output parameter *RESULT* and in an output results file (see section 6.12).

Notes:

- At step iv, the name of the file containing the data defining the waveform transform is of the form "CONST $f0$ GHz n.mat", where $f0$ is the frequency-spacing (provided as input) and n is the number of frequencies (defined by the inputs to the module).
- The inputs to the module are DC, $f0$, $zref$ and $uzref$ (when $zref$ and $uzref$ are complex-valued vectors of length greater than one).
- The inputs to the module are DC, $f0$, $zref$, $uzref$ and n (when $zref$ and $uzref$ are complex-valued scalars).
- The output of the module is a single column array, called *RESULT*, which contains the information defining $\boldsymbol{V}_0$ (see section 6.12), to which the operations of calculating a frequency response, forming the time-domain representation, etc., may be applied.

6.10 OTHER MODULES

In addition to the above modules, the following Matlab modules are used:

- "alignment.m";
- "alignment_compact.m";
- "CVapp_fun.m";
- "CVfun.m";
- "CVfun_compact.m";
- "CVgen.m";
- "ExcelCol.m" (from Matlab Exchange);
- "fullcovariance.m";
- "LoadCompactCV_eig.m";
- "Qgen.m";
- "Qgen_compact.m".

It is assumed that all Matlab modules are in the Matlab working directory or a directory included on the Matlab path.

6.11 INPUT DATA FILES

The Matlab script "CreateCompactCV_eig.m" reads data from:

1. A comma separated file (with extension "csv") using Matlab's "xlsread.m" function*;

2. A delimited ascii file (with extension "dat") using Matlab's "importdata.m" function.

In either case, the file is assumed to contain the following information arranged into columns:

- Column 1: Uniformly-spaced frequencies f_k, $k = 0, \ldots, n$, in Hz, with the first frequency value equal to 0 Hz;

- For $i = 1, \ldots, m$:

    - Column $2i$: Values $\Re_{i,k}$, $k = 0, \ldots, n$, for the real part of the frequency-domain waveform corresponding to frequency k and waveform i;

    - Column $2i + 1$: Values $\Im_{i,k}$, $k = 0, \ldots, n$, for the imaginary part of the frequency-domain waveform corresponding to frequency k and waveform i, with $\Im_{i,0} = 0$, $i = 1, \ldots, m$.

---

* This requires a licensed copy of Excel.

The Matlab script “CreateCompact_CV_eig_compact.m” on;y reads data from comma separated files.

6.12 OUTPUT RESULTS FILES

The Matlab script “CreateCompactCV_eig.m” (or “CreateCompactCV_eig_compact.m”) writes results to a Matlab MAT-file with the same name as the input data file from which data is read but with the extension “mat”.

The file contains a single column array, called *RESULT*, which is the output parameter of the module and contains the following information:

1. The number of auxiliary parameters;

2. Auxiliary parameters, consisting of:
   a) The number n of frequencies;
   b) The DC (zero frequency) value of the average frequency-domain waveform;
   c) The number p of eigencomponents used to define the compact representation of the covariance matrix $\boldsymbol{V}_0$;
   d) The frequency spacing in GHz;

3. The real parts of the coefficients of the average frequency-domain waveform corresponding to frequencies f_k, $k = 1, \ldots, n$;

4. The imaginary parts of the coefficients of the average frequency-domain waveform corresponding to frequencies f_k, $k = 1, \ldots, n$;

5. The elements $d_{1,k}$, $k = 1, \ldots, n$, used to create the diagonals in the “ReRe” block of $\boldsymbol{V}_0$;

6. The elements $d_{3,k}$, $k = 1, \ldots, n$, used to create the diagonals in the “ImIm” block of $\boldsymbol{V}_0$;

7. The elements $d_{2,k}$, $k = 1, \ldots, n$, used to create the diagonals in the “ImRe” block of $\boldsymbol{V}_0$;

8. The elements of the columns of the matrix $\boldsymbol{Q}_0$ defined by the p eigenvectors and eigenvalues in an eigendecomposition of the full covariance matrix $\boldsymbol{V}$.

All other Matlab modules read data from the MAT-files generated by “CreateCompactCV_eig.m” or “CreateCompactCV_eig_compact.m”.

7 EXAMPLES

The Matlab script “TestScript.m” illustrates the use of the Matlab modules described in section 6. Each function may be also called separately at the command line as described in the following sections. The results described are obtained using the input data files “FFT_data_positive.csv” and “FFT_data_negative.csv”.

7.1 CALCULATE COMPRESSED COVARIANCE MATRIX

The command

```
CreateCompactCV_eig;
```

or

```
RESULT = CreateCompactCV_eig;
```

followed by the selection of the input data file “FFT_data_positive.csv” generates the graphical output shown in Figure 3 to Figure 5 as well as the output results file “FFT_data_positive.mat”. In this example the input data file contains data for *n* = 99 frequencies and *m* = 400 waveforms. Figure 3**Error! Reference source not found.** shows the full covariance matrix and Figure 4 the covariance matrix formed from a compact representation of an approximation to the full covariance matrix defined by *p* = 9 eigenvectors and eigenvalues in an eigendecomposition of the covariance matrix (Figure 6). The differences between the full and approximate covariance matrices are shown in Figure 5. In this example the compression achieved in *S* = 5.3 % and the approximation achieved is *E* = 9.2 %.

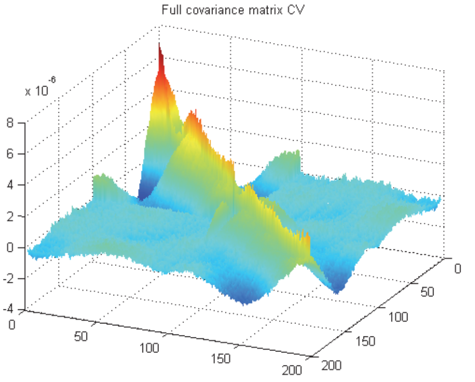


Figure 3 Full covariance matrix corresponding to input data file “FFT_data_positive.csv”.

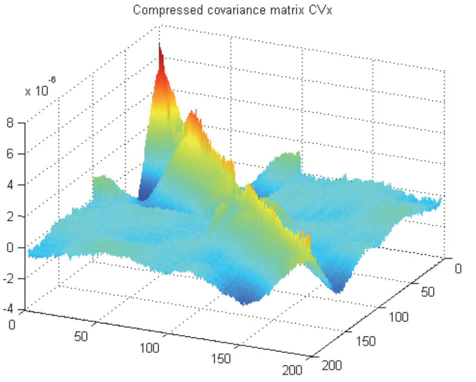


Figure 4 Covariance matrix formed from a compact representation of the matrix shown in Figure 3.

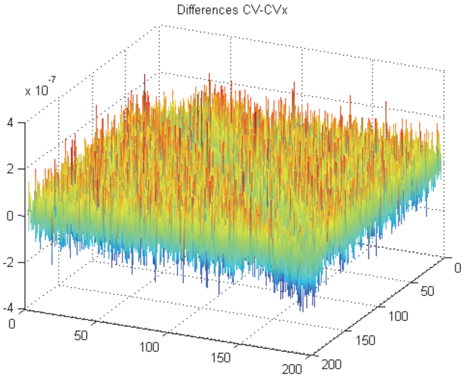


Figure 5 Differences between the covariance matrices shown in Figure 3 and Figure 4.

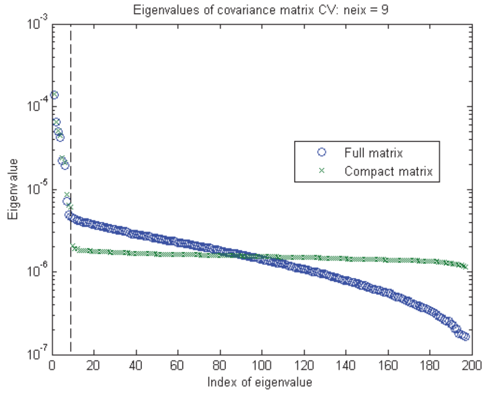


Figure 6 Eigenvalues of the covariance matrices shown in Figure 3 and Figure 4.

The command

```
[DC, f0, Rzref, Izref, CVx] = …
      FormCovarianceMatrixFromCompactCV_eig;
```

or

```
[DC, f0, Rzref, Izref, CVx] = …
      FormCovarianceMatrixFromCompactCV_eig(RESULT);
```

places into the Matlab workspace the GC value, the sampling frequency, the real and imaginary parts of the components of the average frequency-domain waveform, and the associated covariance matrix obtained from the compact representation of the matrix.

7.2 CALCULATE AMPLITUDE AND PHASE RESPONSES

The command

```
FormFreqRespFromCompactCV_eig;
```

or

```
[Aref, uAref, Pref, uPref] = FormFreqRespFromCompactCV_eig;
```

followed by the selection of the output results file “FFT_data_positive.mat” generates the graphical output shown in Figure 7 to Figure 10. These figures show the estimates of amplitude and phase response, with the associated standard uncertainties, defined by the average frequency-domain waveform and its corresponding covariance matrix stored in the output results file “FFT_data_positive.mat”.

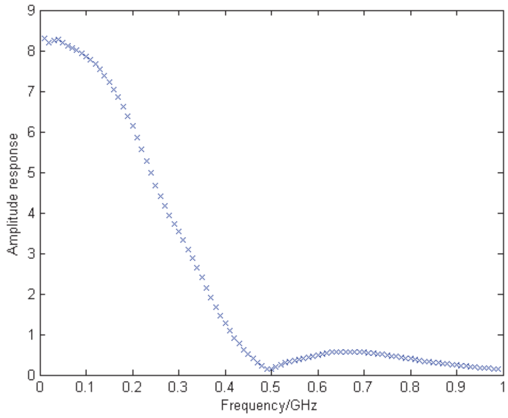


Figure 7 Estimate of the amplitude response corresponding to input data file “FFT_data_positive.csv”.

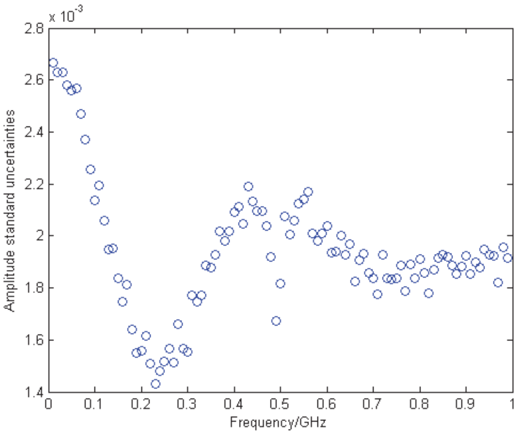


Figure 8 Standard uncertainties associated with the amplitude response shown in Figure 7.

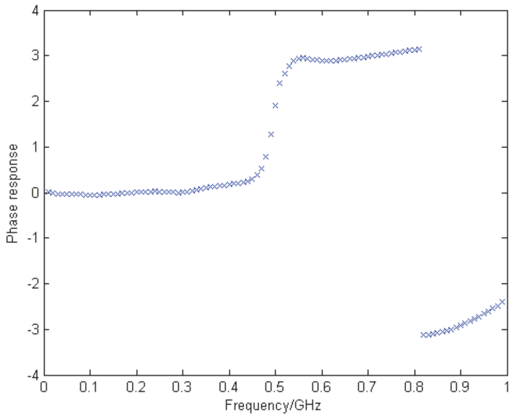


Figure 9 Estimate of the phase response corresponding to input data file “FFT_data_positive.csv”.

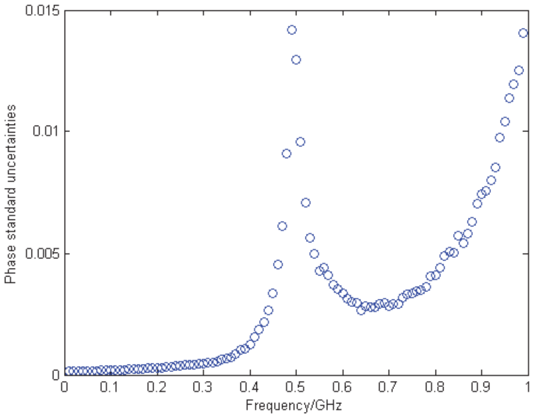


Figure 10 Standard uncertainties associated with the phase response shown in Figure 9.

7.3 CALCULATE TIME-DOMAIN REPRESENTATION

The command

```
FormTimeSeriesFromCompactCV_eig(0, 0);
```

or

```
[t, tref, utref] = FormTimeSeriesFromCompactCV_eig(0, 0);
```

followed by the selection of the output results file "FFT_data_positive.mat" generates the graphical output shown in Figure 11 and Figure 12 to Figure 14. These figures show the estimates of time-domain representation, with the associated standard uncertainties, defined by the average frequency-domain waveform and its corresponding covariance matrix stored in the output results file "FFT_data_positive.mat". These figures are obtained using the default set of time values.

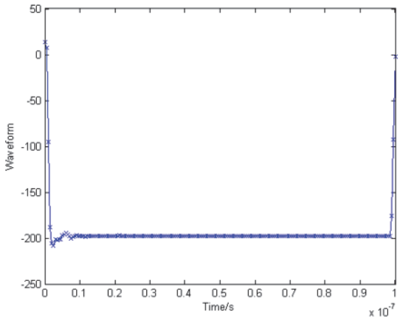
Alternatively, the commands

```
FormTimeSeriesFromCompactCV_eig(0, 99);
```

or

```
[t, tref, utref] = FormTimeSeriesFromCompactCV_eig(0, 99);
```

generates the graphical output shown in Figure 13and Figure 14. These figures are obtained accounting for a phase-shift.



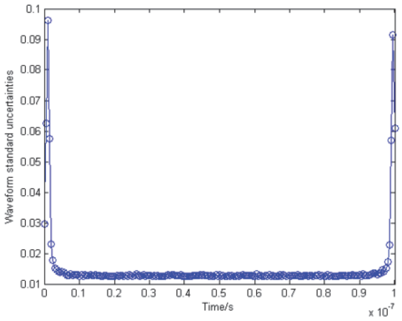
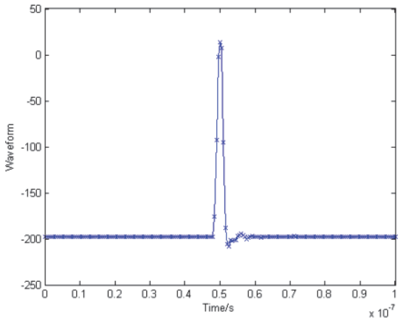


Figure 11 Estimate of the time series corresponding to input data file "FFT_data_positive.csv" for a default set of time values.

Figure 12 Standard uncertainties associated with the time series shown in Figure 11



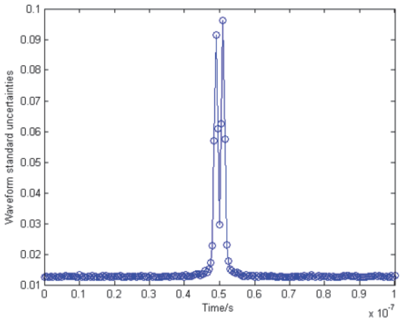


Figure 13 Estimate of the time series corresponding to input data file "FFT_data_positive.csv" for a user-supplied time-shift.

Figure 14 Standard uncertainties associated with the time series shown in Figure 13.

7.4 CALCULATE PRODUCT OF TWO FREQUENCY-DOMAIN WAVEFORMS

The command

```
FormMultiplyFromCompactCV_eig;
```

or

```
RESULT = FormMultiplyFromCompactCV_eig;
```

followed by the selection of the output results files "FFT_data_positive.mat" and "FFT_data_negative.mat" generates the (further) output results file "FFT_data_positive MULT FFT_data_negative.mat", which contains the frequency-by-frequency product of the average frequency-domain waveforms and a compact representation of the covariance matrix associated with the product.

7.5 CALCULATE QUOTIENT OF TWO FREQUENCY-DOMAIN WAVEFORMS

The command

```
FormDivideFromCompactCV_eig;
```

or

```
RESULT = FormDivideFromCompactCV_eig;
```

followed by the selection of the output results files "FFT_data_positive.mat" and "FFT_data_negative.mat" generates the (further) output results file "FFT_data_positive DIV FFT_data_negative.mat", which contains the frequency-by-frequency quotient of the average frequency-domain waveforms and a compact representation of the covariance matrix associated with the quotient.

7.6 CALCULATE SUM OF TWO FREQUENCY-DOMAIN WAVEFORMS

The command

```
FormAddFromCompactCV_eig;
```

or

```
RESULT = FormAddFromCompactCV_eig;
```

followed by the selection of the output results files "FFT_data_positive.mat" and "FFT_data_negative.mat" generates the (further) output results file "FFT_data_positive ADD FFT_data_negative.mat", which contains the frequency-by-frequency sum of the average frequency-domain waveforms and a compact representation of the covariance matrix associated with the sum.

7.7 CALCULATE DIFFERENCE OF TWO FREQUENCY-DOMAIN WAVEFORMS

The command

```
FormSubtractFromCompactCV_eig;
```

or

```
RESULT = FormSubtractFromCompactCV_eig;
```

followed by the selection of the output results files “FFT_data_positive.mat” and “FFT_data_negative.mat” generates the (further) output results file “FFT_data_positive SUB FFT_data_negative.mat”, which contains the frequency-by-frequency difference of the average frequency-domain waveforms and a compact representation of the covariance matrix associated with the difference.

7.8 FORMING COVARIANCE MATRIX FOR A USER-SUPPLIED TRANSFORM

The commands

```
DC = 0;
f0 = 0.1;
zref = complex(1, 0);
uzref = complex(0.01, 0);
n = 511;
FormConstantCompactCV_eig(DC, f0, zref, uzref, n);
```

generates the output results file “CONST 0.1 GHz 511.mat” containing a frequency-domain waveform with $DC = 0$, $n = 511$ frequencies, sampling frequency $f_0 = 0.1$ GHz and a compact representation of the associated covariance matrix, such that the coefficients of the frequency-domain waveform all have real part 1 with associated standard uncertainty 0.01 and imaginary part 0 with associated standard uncertainty 0.

8 ACKNOWLEDGEMENTS

This work was funded by the UK National Measurement Office under the Physical Programme and by Euramet under the EMRP Industry 2010 Programme.

9 REFERENCES

---

[1] EMRP IND16 “Ultrafast” JRP website, www.ptb.de/emrp/ultrafast.html

[2] D. A. Humphreys, P. M. Harris, M. Rodríguez-Higuero, F. Mubarak, D. Zhao, Member, and K. Ojasalo, “Principal Component Compression Method for Covariance Matrices Used for Uncertainty Propagation,” IEEE Trans. IM to be published, Digital Object Identifier 10.1109/TIM.2014.2340640, available at http://ieeexplore.ieee.org/ , Aug. 2014.

[3] D A Humphreys, P M Harris, J Miall, “Instrument related structure in covariance matrices used for uncertainty propagation,” Proc. 42nd European Microwave Conference, Amsterdam, 28th October to 1st November 2012. Poster01-54, pp. 1304-1307.

[4] K. Ojasalo, D. Humphreys, and P. Harris, “Preliminary report: compact covariance representation of waveform uncertainties,” Proc. XXXIII Finnish URSI Convention on Radio Science SMARAD Seminar, Dipoli, Otaniemi, Finland, 24 – 25 April 2013, pp.147-150.