Uncertainty propagation on a nonlinear measurement model based on Taylor expansion

Min-Hee Gu1,2, Chihyun Cho1, Hahng-Yun Chu2, No-Weon Kang1 and Joo-Gwang Lee1

Abstract

In this paper, the propagation of uncertainty on a nonlinear measurement model is presented using a higher-order Taylor series. As the derived formula is based on a Taylor series, it is necessary to compute the partial derivatives of the nonlinear measurement model and the correlation among the various products of the input variables. To simplify the approximation of this formula, most previous studies assumed that the input variables follow independent Gaussian distributions. However, in this study, we generate multivariate random variables based on copulas and obtain the covariances among the products of various input variables. By applying the derived formula to various cases regardless of the error distribution, we obtained the results that coincide with those of a Monte-Carlo simulation. To apply high-order Taylor expansion, the nonlinear measurement model should be continuous within the range of the input variables to allow for differentiation, and be an analytic function in order to be represented by a power series. This approach may replace some time-consuming Monte-Carlo simulations by choosing the appropriate order of the Taylor series, and can be used to check the linearity of the uncertainty.

Keywords
uncertainty, nonlinear, Monte-Carlo, Taylor

Introduction

In describing the propagation of uncertainty through a system, the first-order Taylor series is highly recommended by the Guide to the Expression of Uncertainty in Measurement (GUM).1 However, it is difficult to accurately predict the uncertainty of a nonlinear measurement model using a first-order Taylor expansion. Thus, the GUM suggests adding higher-order terms of the Taylor expansion when the model cannot be represented linearly. The high-order terms are obtained from the second order Taylor expansion with the assumption that the input variables have normal distribution.

Monte-Carlo (MC) simulation is widely used for nonlinear measurement model since it can predict distributions as well as standard deviations of measurement model outputs. It is strongly recommended that sufficient number of simulations are required to achieve high accuracy of the estimation. For example, the GUM recommends a minimum of 200,000 simulations to obtain a 95% confidence.2 Some simulations such as finite element method (FEM) numerical analysis, statistical parameter estimations, and complex signal processing often take hours to days.3–5 Therefore, in these cases, MC simulation is not suitable for the uncertainty propagation on the nonlinear measurement model. Various studies on uncertainty propagation have been conducted, and they are well summarized in Liu et al.,6,7 Meng et al.,8 and Ouyang et al.9 Recent studies showed that the probability density function (pdf) of the output of the measurement model could be predicted even through the input has arbitrary distribution. Most studies, however, assume independent input variables or approximate by truncating high orders terms.

Recently, Wang and Iyer10 added third-order terms to reduce the error produced by the truncation of the Taylor expansion, and showed that the uncertainty of the nonlinear measurement model could be predicted...
more accurately than using the technique recommended in GUM. Zhang extended the Taylor series to an infinite number of terms and proposed some methods to calculate the variance of the system.\textsuperscript{11} Mekid and Vaja\textsuperscript{12} calculated the uncertainty of the measurement model up to the third order using the skewness and kurtosis of the input variables. Xue et al.,\textsuperscript{13} derived the variances and covariances of the model based on the infinite expansion of the Taylor series. These approaches assume that the input variables are independent (or dependent) normal distributions, thus removing terms that are difficult to calculate analytically. Thus, they cannot be applied when the input variables do not obey a normal distribution and/or are correlated with each other.

In this paper, we re-derive the Taylor series for the covariance of a nonlinear measurement model regardless of distribution and correlation of the variables. We also show the method how to apply the derived formula to simple numerical simulation. In the numerical approach, we calculated the non-analytic terms in the formula (such as the covariance between products of input variables) from the multivariate random variables generated based on copulas. To obtain the partial derivatives of the system, an n-point stencil is applied across the multiple dimensions. As a result, the formula based on the higher-order Taylor expansion can be fully applied to covariance calculations on the nonlinear measurement model, irrespective of the distribution of the input variables.

This paper is organized as follows. In Section 2, the covariance of the nonlinear measurement model is re-derived using a Taylor series expansion. In Section 3, we show examples of applying the derived formula by simple numerical approach. Section 4 discusses the limitations of Taylor approximation, and Section 5 presents the conclusions to this study.

**Theory**

In probability theory and statistics, covariance is a measure of the degree of correlation between two random variables, \(x_1\) and \(x_2\). By applying the linearity of expectations, this can be represented as the “expected value of their product” minus the “product of their expected values.”

\[
\text{cov}(x_1, x_2) = \langle x_1 - \langle x_1 \rangle \rangle \langle x_2 - \langle x_2 \rangle \rangle \nonumber \\
= \langle x_1 x_2 \rangle - \langle x_1 \rangle \langle x_2 \rangle - \langle x_1 \rangle \langle x_2 \rangle + \langle x_1 \rangle \langle x_2 \rangle 
\]

(1)

where \(\langle \cdot \rangle\) denotes the expectation of the argument. The approximate output of a system using the first-order Taylor expansion is

\[
y_i = f(x_1 + z_1, \ldots, x_m + z_m) \approx f(x_1, \ldots, x_m) + \sum_{k=1}^{m} \frac{\partial f(x_1, \ldots, x_m)}{\partial x_k} z_k
\]

(2)

where \(x_k\) is the input variable, \(z_k\) is the error in \(x_k\), and \(y\) is the output of the function. \(f\) is a real-valued function that is differentiable on \(x_1, \ldots, x_m\); for simplicity, \(f(x_1, \ldots, x_m)\) will be represented as \(f\). Thus, the covariance using the first-order Taylor expansion is given by:

\[
\text{cov}(y_i, y_j) = \text{cov} \left( f_i + \sum_{k=1}^{m} \frac{\partial f_i}{\partial x_k} z_k, f_j + \sum_{p=1}^{m} \frac{\partial f_j}{\partial x_p} z_p \right) 
\]

\[
= \text{cov} \left( \sum_{k=1}^{m} \frac{\partial f_i}{\partial x_k} z_k, \sum_{p=1}^{m} \frac{\partial f_j}{\partial x_p} z_p \right) 
\]

\[
= \sum_{k=1}^{m} \frac{\partial f_i}{\partial x_k} \sum_{p=1}^{m} \frac{\partial f_j}{\partial x_p} \langle z_k z_p \rangle 
\]

\[
= \sum_{k=1}^{m} \frac{\partial f_i}{\partial x_k} \frac{\partial f_j}{\partial x_p} \text{cov}(z_k, z_p)
\]

(3)

If there are \(m\) random variables, an \(m \times m\) covariance matrix \(\Sigma\) can be obtained.

\[
\Sigma_{i,j} = \text{cov}(y_i, y_j) \quad (i,j = 1, \ldots, m)
\]

(4)

The covariance matrix \(\Sigma^y\) on the outputs \(y = (y_1, \ldots, y_n)\) can be expressed as a product of matrices using the above approximate equation.

\[
\Sigma^y = \mathbf{J} \Sigma \mathbf{J}^T
\]

(5)

\[
\mathbf{J} = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_m}
\end{bmatrix}
\]

(6)

Here, \(\mathbf{J}\) is the Jacobian matrix of \(f = (f_1, \ldots, f_n)\), and \(\mathbf{J}^T\) is the transpose of \(\mathbf{J}\). This formula is known to describe the propagation of uncertainty in linear measurement model.\textsuperscript{1}

Equation (5), however, is not appropriate when the measurement model \(f\) is not linear. Thus, we require an accurate model for the nonlinear \(f\) and a method of propagating the uncertainty through it. In this paper, we use a higher-order Taylor expansion to obtain the covariance matrix of the nonlinear measurement model. It is not guaranteed that the covariance matrix will be expressed as a simple product of matrices, similar to equation (5). The approximation of the outputs using the second-order Taylor expansion is given by

\[
y_i = f_i(x_1 + z_1, \ldots, x_m + z_m) \approx f_i(x_1, \ldots, x_m) + \sum_{k=1}^{m} \frac{\partial f_i}{\partial x_k} z_k + \frac{1}{2} \sum_{k=1}^{m} \frac{\partial^2 f_i}{\partial x_k^2} z_k^2
\]

(7)
The additive law of covariance is useful to simplify the Taylor expansion in equation (1).

\[
\text{cov}(ax_1 + bx_2, cx_3 + dx_4) = ac\text{cov}(x_1, x_3) + ad\text{cov}(x_1, x_4) + bc\text{cov}(x_2, x_3) + bd\text{cov}(x_2, x_4)
\]

(8)

where \(a, b, c,\) and \(d\) are constants. Using equation (8), we can obtain approximations of the covariance of two outputs of the system:

\[
\text{cov}(y_i, y_j) \approx \text{cov} \left( f_i + \sum_{k=1}^{m} \frac{\partial f_i}{\partial x_k} z_k + \frac{1}{2} \sum_{k, h=1}^{m} \frac{\partial^2 f_i}{\partial x_k \partial x_h} z_k z_h, f_j + \sum_{p=1}^{m} \frac{\partial f_j}{\partial x_p} z_p + \frac{1}{2} \sum_{p, q=1}^{m} \frac{\partial^2 f_j}{\partial x_p \partial x_q} z_p z_q \right)
\]

\[
= \text{cov} \left( \sum_{k=1}^{m} \frac{\partial f_i}{\partial x_k} z_k + \frac{1}{2} \sum_{k, h=1}^{m} \frac{\partial^2 f_i}{\partial x_k \partial x_h} z_k z_h, \sum_{p=1}^{m} \frac{\partial f_j}{\partial x_p} z_p + \frac{1}{2} \sum_{p, q=1}^{m} \frac{\partial^2 f_j}{\partial x_p \partial x_q} z_p z_q \right) + \frac{1}{4} \sum_{k, h, p, q=1}^{m} \frac{\partial^2 f_i}{\partial x_k \partial x_h} \frac{\partial^2 f_j}{\partial x_p \partial x_q} \text{cov}(z_k, z_h, z_p, z_q)
\]

(9)

Similarly, the covariance of two outputs for the \(r\)-th order can be obtained based on the \(r\)-th order Taylor expansion.

\[
\text{cov}(y_i, y_j) \approx \sum_{p, q=1}^{m} \frac{1}{p! q!} \prod_{i=1}^{p} \frac{\partial^p f_i(x_1, \ldots, x_n)}{\partial x_{i_1} \cdots \partial x_{i_p}} \frac{\partial^q f_j(x_1, \ldots, x_n)}{\partial x_{j_1} \cdots \partial x_{j_q}} \text{cov}(z_{i_1}, \ldots, z_{i_p}, z_{j_1}, \ldots, z_{j_q})
\]

(10)

As a result, the covariance of the nonlinear measurement model can be represented as the covariance of the various products of input variables with their partial derivatives. Note that the analytic solution for the covariance of the product of variables has not known when they are not normal distribution. This makes equation (10) difficult to apply to the real problem. Equation (10), however, can be applied using a numeric calculation if we can compute the numerical partial derivatives and the numerical covariance for the product of input variables. We will show this in following section.

**Numerical approach**

As mentioned in the previous section, in order to apply equation (10), the partial difference of the measurement model and the covariance between the products of the input variables are required. There are various ways to find precisely the partial difference. In this study, we simply applied \(n\)-point stencil to focus on the applicability of the derived equation. It is beyond the scope of this paper to analytically calculate the covariance between the product of the input variables. Instead, we calculated it numerically and applied it to equation (10).

**Numerical partial differentiation**

The \(n\)-point stencil is widely used to obtain numerical derivatives. For example, the first partial derivative using a 5-point stencil is as follows:

\[
\frac{\partial f(x)}{\partial x} \approx \frac{1}{12h} (f(x - 2h) - 8f(x - h) + 8f(x + h) - f(x + 2h))
\]

(11)

The \(n\)-point stencil can also be used to compute derivatives on the multivariable. However, we use a multidimensional \(m\)-point stencil to simplify the numerical implementation, as shown in Figure 1. For \(O\) variables, we require \(O^O\) points. For the first-order partial derivatives, equation (11) is applied along each axis. As the order of the differentiation increases, such as for
\[ \sigma^2 f/\sigma^2 x, \] equation (11) is modified appropriately; the formula can be found in Fornberg.\textsuperscript{14} Note that this approach does not require additional induction to derive the appropriate formulas.

**Covariance of various products of input variances**

Equation (10) requires the covariances of various products of input variables, such as \( \text{cov}(z_1 \cdot \ldots \cdot z_p, z_1 \cdot \ldots \cdot z_q) \).

Although there have been some studies on the covariance between products of random variables,\textsuperscript{15,16} we found that these approximations are not sufficiently accurate for this study. Instead, we generated multivariate random variables from the covariances of input variables using the relevant copula and found the covariances of the products of various combinations.

In applying the copula, multivariate normal random numbers are generated using Cholesky decomposition and converted to uniform random vectors.\textsuperscript{17} These vectors are then transformed back to random numbers with the target marginal distributions using the inverse cumulative distribution function in MATLAB.\textsuperscript{18} Depending on the marginal distribution, there may be a difference in the correlations between the target and the generated variables because of the nonlinearity of the transform. To overcome this issue, we apply Spearman's ranked correlation. In a normal distribution, the Pearson correlation coefficient is related to Spearman’s rho as:\textsuperscript{19}

\[ \rho = 2 \sin \left( \frac{\pi}{6} \rho_s \right) \]  

where \( \rho_s \) and \( \rho \) are Spearman’s and Pearson’s correlation coefficients, respectively. In this paper, to focus on the numerical approach, equation (12) is applied repeatedly until the generated random variables meet the target covariances. The detailed procedure is described in the appendix.

**Results**

We tested the covariances of the nonlinear measurement model derived from the previous section using four nonlinear functions and five random variable distributions. The nonlinear functions are \( y_1 = x_1^2 + x_2^2 + 2, \ y_2 = x_1^2 + x_2^2, \ y_3 = x_1^2 x_2, \) and \( y_4 = x_1 x_2 \). The distributions considered here are the normal, exponential, uniform, gamma, and Rayleigh distributions; details are summarized in Table 1. We calculated the covariance of \( y_1 \) and \( y_2 \) and the covariance of \( y_3 \) and \( y_4 \) for the various distributions of input variables using a MC simulation and first-, second-, and third-order Taylor approximations. The MC simulations consisted of 1,000,000 runs; all results are summarized in Table 2.

---

### Table 1. Distribution of test variable.

| Distribution | Denote | Mean | Variance |
|--------------|--------|------|----------|
| Normal       | \( N(\mu, \sigma^2) \) | \( \mu \) | \( \sigma^2 \) |
| Exponential  | \( \text{Exp}(\lambda) \) | \( \lambda^{-1} \) | \( \lambda^{-2} \) |
| Uniform      | \( U(a, b) \) | \( \frac{1}{2}(a + b) \) | \( \frac{1}{12}(b - a)^2 \) |
| Gamma        | \( \Gamma(k, \theta) \) | \( k \theta \) | \( k \theta^2 \) |
| Rayleigh     | \( R(\sigma) \) | \( \sigma \sqrt{\frac{\pi}{2}} \) | \( \frac{4}{27} \sigma^2 \) |

---

### Table 2. Comparison of covariance using the MC simulation and the Taylor expansion.

| Type (\( \text{var} \)) | Distribution of \( x_1 \) and \( x_2 \) | Covariance of input | Monte-Carlo | First order | Second order | Third order |
|-------------------------|------------------------------------------|--------------------|-------------|-------------|--------------|-------------|
| \( \Sigma^{(n \times n)} \) | \( N(0, 2), N(0, 3) \) | | 2.0031 | 1.2050 | 137.82 | 20.84 | 4.45 | 4.33 | 10^{-5} | 18.09 | 20.98 | 137.82 | 20.84 | \n| | \( \text{Exp}(1/\sqrt{2}), \text{Exp}(1/\sqrt{3}) \) | | 2.0010 | 1.2166 | 6308.6 | 1146.5 | 158.9 | 107.3 | 10 | 107.3 | 107.3 | 1416.1 | 645.7 | 6308.6 | 1146.5 | \n| | \( U(-\sqrt{6}, \sqrt{6}), U(-3, 3) \) | | 2.0013 | 1.2032 | 38.17 | 8.24 | 2.56 | 3.02 | 10^{-5} | 7.21 | 8.22 | 38.17 | 8.24 | \n| | \( \Gamma(5, \sqrt{2}/3), \Gamma(3, 1) \) | | 2.0009 | 1.2086 | 8031.0 | 1675.0 | 2353.1 | 753.6 | 280.7 | 1472.7 | 486.9 | 1675.0 | 486.9 | \n| | \( R\left(\frac{\sqrt{4}}{\pi} - \theta\right), R\left(\frac{\sqrt{4}}{\pi} - \theta\right) \) | | 2.0016 | 1.2260 | 4339.9 | 1135.1 | 1455.1 | 592.4 | 278.5 | 968.9 | 402.8 | 1135.1 | 402.8 | \n| \( \Sigma^{(n \times n)} \) | \( N(0, 2), \text{Exp}(1/\sqrt{3}) \) | | 1.9979 | 1.2009 | 140.44 | 20.00 | 0.00 | -0.00 | 23.89 | 8.50 | 140.44 | 42.00 | \n| | \( \text{Exp}(1/\sqrt{2}), U(-3, 3) \) | | 2.0037 | 1.1713 | 477.39 | 71.90 | 12.01 | 8.50 | 74.07 | 31.26 | 477.39 | 71.90 | \n| | \( U(-\sqrt{6}, \sqrt{6}), \Gamma(3, 1) \) | | 1.1713 | 3.0000 | 71.90 | 14.35 | 8.50 | 6.00 | 31.26 | 14.35 | 71.90 | 14.35 | \n| | \( \Gamma(5, \sqrt{2}/3), R\left(\frac{\sqrt{4}}{\pi} - \theta\right) \) | | 2.0003 | 1.2096 | 66.46 | 27.19 | 0.00 | -0.04 | 28.78 | 6.86 | 66.46 | 27.19 | \n| | \( R\left(\frac{\sqrt{4}}{\pi} - \theta\right), N(0, 3) \) | | 2.0021 | 1.2202 | 4610.6 | 650.2 | 1686.6 | 354.4 | 571.8 | 6460.6 | 650.2 | \n| | \( \text{Exp}(1/\sqrt{2}), U(-3, 3) \) | | 1.2096 | 3.0030 | 27.19 | 25.72 | -0.04 | 18.01 | 8.66 | 25.72 | 27.19 | 25.72 | \n| | \( \Gamma(5, \sqrt{2}/3), R\left(\frac{\sqrt{4}}{\pi} - \theta\right) \) | | 1.2202 | 2.9946 | 650.2 | 102.1 | 354.4 | 77.4 | 561.8 | 102.1 | 650.2 | 102.1 | \n| | \( R\left(\frac{\sqrt{4}}{\pi} - \theta\right), N(0, 3) \) | | 2.0019 | 1.2265 | 730.04 | 140.57 | 160.36 | 59.31 | 389.57 | 105.27 | 730.04 | 140.57 | \n| | \( \text{Exp}(1/\sqrt{2}), U(-3, 3) \) | | 1.2265 | 3.0025 | 140.57 | 30.79 | 59.31 | 21.94 | 105.27 | 30.79 | 140.57 | 30.79 |
All random variables were generated to have the covariance matrix
\[
\begin{bmatrix}
2 & 1.2 \\
1.2 & 3
\end{bmatrix}
\]
. The covariances of the generated random variables (denoted as “Covariance of input” in Table 2) exhibit only small differences from the target covariances according to Spearman’s ranked correlation. As expected, the first- and second-order Taylor approximations are quite different from the MC simulation results, but the covariances of the MC simulations and the third-order Taylor approximation are exactly the same, regardless of the distribution of random variables. Note that the MC simulation results for the variance of \(y_2\) and \(y_4\) are the same as those given by the second-order Taylor approximation, and they do not change as the order of the Taylor approximation increases. This means that we can determine the order of Taylor expansion, although the exact mathematical definition of the function is unknown. This will be described in more detail in the following section.

\section*{Limitation of Taylor approximation}

In this section, we study the limitation of the covariance based on Taylor approximation. The function \(f(x) = 1/(1 - x)\) is used as the nonlinear function. For the accuracy of calculation and convenience of analysis, the derivative of this function has been derived analytically and only one random variable is examined. The results of the Taylor series expansion of \(f\) are shown in Figure 2, where \(x = 2\) and \(z = 0.9\) and 1.01 (see equation (7)). For \(z = 0.9\), the Taylor approximation converges to a value of \(1/(1 - 2.9)\) when the differential order exceeds 40. For \(z = 1.01\), however, the Taylor approximation diverges gradually. This is because the function \(f\) becomes discontinuous when the denominator is zero. Therefore, for the above function to be expressed as a Taylor series, the value of \(z\) should be between \(-1\) and 1. \(^{20}\) This corresponds well to the fact that the differentiable functions only can be represented as Taylor series. Figure 3 shows the variance of \(f\) as a function of the Taylor expansion’s order. When the variance of the input variable is 0.03, the variance of \(f\) using the proposed method approaches the result of the MC simulations (dashed line). However, it diverges from the MC simulation results for the case of 0.05, because the distribution of the random variable \(f(x)\) deviates from the continuous range, as shown in Figure 4(b). Therefore, the nonlinear measurement model approximating with the Taylor series should be a local continuous function that can be differential within the input variable range. It should also be an analytic function that can be expressed as the sum of the power series. Figure 3 additionally reveals that in the case of the measurement models that are difficult to describe mathematically, such as EVM and eye pattern measurement of complex signals, \(^{5,21}\) increasing the order of...
Taylor series and examining its convergence will be helpful in determining the differential order. Note that equations (9) and (10) show rapidly increasing number of calculation as the order of the Taylor series increase. For example, as the order increases to $r$, the amount of partial differential computation increases to $\sum_{k=1}^{r} 2(k + 1)$. Therefore, it is important to choose the appropriate order when applying the Taylor approximation.

**Conclusion**

In this paper, we have re-derived a formula for estimating the covariances of nonlinear measurement model using high-order Taylor expansions. The presence of these high-order terms means that the expression is analytically intractable. Many previous studies have simplified these terms by assuming that the input variables obey normal distributions, whereas we generated multivariate random variables with arbitrary distributions and calculated these terms numerically. The estimation results were shown to be exactly the same as those given by MC simulations. We also showed that the nonlinear measurement model should be an analytic function that can be differentiated within the range of the input variables to apply high-order Taylor approximation.

The method used in Section 3 (Numerical Approach) may be used instead of time-consuming MC simulations such as the uncertainty evaluation on the finite element method (FEM) numerical analysis, statistical parameter estimations, and complex signal processing. It can also be used to evaluate the linearity of the uncertainty. This paper has focused on showing that the uncertainty propagation of nonlinear measurement model can be represented as high-order Taylor expansion and it can be applied in real problem using numerical approaches. Thus, it is left to the future studies that how to precisely generate multivariable random variables.

**Author’s note**

Min-Hee Gu is now affiliated with Pixoneer geomatics.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by Physical Metrology for National Strategic Needs funded by Korea Research Institute of Standards and Science [KRISS-2019-GP2019-0005].

**ORCID iD**

Chihyun Cho https://orcid.org/0000-0003-2506-576X

**References**

1. BIPM, IEC, IFCC, ISO, IUPAC, IUPAP and OIML. Guide to the Expression of Uncertainty in Measurement (GUM:1995 with minor corrections), Bureau International des Poids et Mesures, JCGM 100, 2008.
2. BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP and OIML. Evaluation of measurement data—supplement 1 to the guide to the expression of uncertainty in measurement—propagation of distributions using a Monte Carlo method, Bureau International des Poids et Mesures, JCGM 101, 2008.
3. Hoffmann JP, Leuchtmann P and Vahlbruch R. Pin gap investigations for the 1.85mm coaxial connector. In: 37th European microwave conference, Munich, 8–12 October 2007, pp.388–391. IEEE.
4. Hoffmann J, Leuchtmann P, Rufeneracht J, et al. A stable bayesian vector network analyzer calibration algorithm. IEEE Trans Microw Theory Tech 2009; 57: 869–880.
5. Cho C, Lee JG, Kim JH, et al. Uncertainty analysis in EVM measurement using a Monte Carlo simulation. IEEE Trans Instrum Meas 2015; 64: 1413–1418.
6. Liu J, Meng X, Xu C, et al. Forward and inverse structural uncertainty propagations under stochastic variables with arbitrary probability distributions. Comput Methods Appl Mech Eng 2018; 342: 287–320.
7. Liu J, Cao L, Jiang C, et al. Parallelepiped-formed evidence theory model for quantifying uncertainties with correlation. Appl Math Model 2020; 77: 32–48.
8. Meng X, Liu J, Cao L, et al. A general frame for uncertainty propagation under multimodally distributed random variables. Comput Methods Appl Mech Eng 2020; 367: 113109.
9. Ouyang H, Liu J, Han X, et al. Correlation propagation for uncertainty analysis of structures based on a non-probabilistic ellipsoidal model. Appl Math Model 2020; 88: 190–207.
10. Wang CM and Iyer HK. On higher-order corrections for propagating uncertainties. Metrologia 2005; 42: 406–410.
11. Zhang J. The calculating formulae, and experimental methods in error propagation analysis. IEEE Trans Reliab 2006; 55: 169–181.
12. Mekid S and Vaja D. Propagation of uncertainty: expressions of second and third order uncertainty with third and fourth moments. Measurement 2008; 41: 600–609.
13. Xue J, Leung Y and Ma J-H. High-order Taylor series expansion methods for error propagation in geographic information systems. J Geogr Syst 2015; 17: 187–206.
14. Fornberg B. Generation of finite difference formulas on arbitrarily spaced grids. Math Comput 1988; 51: 699–706.
15. Wilkinson JW. Exact expressions for the covariances between products of random variables. Technical report ORNL-TM-1977, Oak Ridge National Laboratory, Atomic Energy Commission, USA, 1967.
16. Bohrnstedt GW and Goldberger AS. On the exact covariance of products of random variables. J Am Stat Assoc 1969; 64: 1439–1442.
17. Genz A. Numerical computation of multivariate normal probabilities. J Comput Graph Stat 1992; 1(2): 141–149.
18. MATLAB and Statistics Toolbox Release 2019b. Natick, MA, The Math Works, Inc., 2019.
19. Hotelling H and Pabst MR. Rank correlation and tests of significance involving no assumption of normality. Ann Math Statist 1936; 7: 29–43.
Appendix

Figure 5 shows the procedure and MATLAB code for generating multivariate random numbers used in both MC simulation and calculation of the covariance between products of input variables. First, the correlation coefficient $\text{CorCoeff}$ is calculated by dividing the target covariance $\text{TargetCov}$ by each standard deviation. Then, based on the calculated CorCoeff, multivariate numbers $z$ having mean of 0 are generated. The generated $z$ is then converted back into a uniform random vector $u$. In order to convert $u$ back to a random number $R$ with the desired target marginal distribution, we use MATLAB's inverse cumulative distribution function (CDF) function. According to the inversion method, applying the inverse CDF of the distribution $F$ to the uniform random variable $U(0,1)$ produces a random number with exactly $F$ distribution, which maintains the correlation coefficient of $z$.

If the covariance of the random number $R$ is similar to that of TargetCov, these numbers are used both MC simulation and calculation of the covariance between product of input variables. If the difference of two covariance is large, apply equation (6) and repeat the above procedure. Thus, there are slightly differences between “covariance of input” and “Target covariance” as shown in Table 2.

| Procedure                                                                 | MATLAB code                              |
|---------------------------------------------------------------------------|------------------------------------------|
| Calculates correlation coefficient                                         | $\text{TargetCov} = [2.1.2; 1.2 3]$;     |
|                                                                           | $\text{Nsim} = 1000000$;                |
| Generates normal random number with zero mean                             | $\text{Std}1 = \sqrt{\text{TargetCov}(1,1)}$; |
|                                                                           | $\text{Std}2 = \sqrt{\text{TargetCov}(2,2)}$; |
|                                                                           | $\text{CorCoeff} = \frac{\text{TargetCov}}{\text{Std}1 \times \text{Std}1 \times \text{Std}2}$; |
|                                                                           | $\text{Std}1 \times \text{Std}2 \times \text{Std}2$; |
|                                                                           | $z = \text{mvnrnd}([0 0], \text{CorCoeff}, \text{Nsim})$; |
|                                                                           | $\ast \text{mvnrnd is the Matlab function that generates multivariate normal}$ |
|                                                                           | $\text{random number based on Cholesky decomposition.}$ |
| Converst uniform random vector                                             | $u = \text{normcdf}(z)$;                |
|                                                                           | $\ast \text{normcdf Matlab function that returns the cumulative distribution}$ |
|                                                                           | $\ast \text{cdf of the normal distribution at } z$. |
| Transforms back to random numbers with the target marginal distributions   | $R = [\text{norminv}(u(;,1), 0, \text{Std}1) \ \text{expinv}(u(;,2),\text{Std}2)]$; |
|                                                                           | $\ast \text{norminv, expinv, unifinv, gaminv, raylinv are Matlab functions}$ |
|                                                                           | $\text{that returns the inverse cumulative distribution function (cdf) of}$ |
|                                                                           | $\ast \text{each distribution at } u$. |
|                                                                           | $\text{while Decision == no}$             |
|                                                                           | $\text{CorCoeff} = 2 \times \sin(\text{CorCoeff} \times \pi / 6)$; |
|                                                                           | $\text{end}$ |

Figure 5. Procedure and MATLAB code for generating multivariable random numbers.