

TMsim: An Algorithmic Tool for the Parametric and Worst-Case Simulation of Systems with Uncertainties

Original

TMsim: An Algorithmic Tool for the Parametric and Worst-Case Simulation of Systems with Uncertainties / Trincherò, Riccardo; Manfredi, Paolo; Stievano, IGOR SIMONE. - In: MATHEMATICAL PROBLEMS IN ENGINEERING. - ISSN 1024-123X. - ELETTRONICO. - 2017:(2017), pp. 1-12. [10.1155/2017/6739857]

Availability:

This version is available at: 11583/2668130 since: 2017-03-31T07:46:51Z

Publisher:

Hindawi

Published

DOI:10.1155/2017/6739857

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Research Article

TMsim: An Algorithmic Tool for the Parametric and Worst-Case Simulation of Systems with Uncertainties

Riccardo Trincherio,¹ Paolo Manfredi,² and Igor S. Stievano¹

¹Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

²Department of Information Technology, IDLab, Ghent University-IMEC, iGent Tower, Technologiepark-Zwijnaarde 15, 9052 Ghent, Belgium

Correspondence should be addressed to Riccardo Trincherio; riccardo.trincherio@polito.it

Received 3 November 2016; Accepted 8 March 2017; Published 30 March 2017

Academic Editor: J.-C. Cortés

Copyright © 2017 Riccardo Trincherio et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a general purpose, algebraic tool—named TMsim—for the combined parametric and worst-case analysis of systems with bounded uncertain parameters. The tool is based on the theory of Taylor models and represents uncertain variables on a bounded domain in terms of a Taylor polynomial plus an interval remainder accounting for truncation and round-off errors. This representation is propagated from inputs to outputs by means of a suitable redefinition of the involved calculations, in both scalar and matrix form. The polynomial provides a parametric approximation of the variable, while the remainder gives a conservative bound of the associated error. The combination between the bound of the polynomial and the interval remainder provides an estimation of the overall (worst-case) bound of the variable. After a preliminary theoretical background, the tool (freely available online) is introduced step by step along with the necessary theoretical notions. As a validation, it is applied to illustrative examples as well as to real-life problems of relevance in electrical engineering applications, specifically a quarter-car model and a continuous-time linear equalizer.

1. Introduction

With the ever-growing complexity of modern design scenarios in every domain of engineering, today's applications are constantly facing the need for techniques that are capable of taking parameter uncertainty into account. Researchers in the field of uncertainty quantification have striven to find alternatives to state-of-the-art approaches, which often turn out to be computationally intensive or inaccurate. In this framework, the strategies can be classified into two groups: probabilistic approaches take the actual distribution of uncertain parameters into account and seek for a statistical characterization of the system variables of interest. A classical example that has been utilized in virtually every engineering problem is the Monte Carlo (MC) method [1]. An alternative and more efficient probabilistic approach that was proposed in relatively recent times is polynomial chaos [2, 3].

On the contrary, interval or worst-case (WC) approaches assume bounded uncertain input parameters and provide

a conservative estimation of the lower and upper bounds of the system variables without explicit information on the actual distribution. The calculations involved in the analysis are suitably modified to provide a conservative estimation of the bound of the result. These methods include interval arithmetic or interval analysis (IA) [4], which is however rather basic and often results in large overestimation since it is unable to track variable dependency [5]. Affine arithmetic (AA) is an improved approach that takes variable dependence into account by means of a linear parametrization [6]. Finally, the Taylor model (TM) utilizes a higher-order polynomial representation in conjunction with a residual interval remainder that takes truncation and round-off errors into account, yielding a well-defined and robust theoretical framework [7–9]. TM calculations are carried out by using standard operations over polynomials or, whenever a non-linear operator occurs, by the rules of Taylor expansion. The interval remainder is propagated and updated by means of Taylor remainder theory and IA. The TM algebra combines

the strength of a high-order parametric representation and of IA, while limiting the overestimation issue to the interval remainder. By adding the interval remainder to the bound of the polynomial part, a rigorous and conservative estimation of the overall (WC) bounds of the variable is obtained.

This paper presents an open source and general purpose algebraic tool, named TMsim, that implements TM algebra and proposes specific improvements in the calculation of remainders and an extension to some matrix operations. Specifically, the remainder of nonlinear operations, which is usually computed in Lagrange form [7], is now calculated also according to the Cauchy formulation. The tightness and convergence differ between the Lagrange and Cauchy forms; thus the best of the two is chosen. Moreover, for the special case of the multiplicative inverse, an alternative and *exact* formulation of the remainder is implemented. The tool also handles basic matrix operations and implements a recently proposed algorithm to implement in TM algebra the matrix inversion [10], which is a critical operation in the solution of many engineering problems.

A preliminary framework was introduced with specific application to the frequency-domain analysis of circuits [10] and transmission lines [11]. In the present paper, the tool is further extended and presented in a more systematic and comprehensive way. It is implemented in MATLAB and made available online [12].

2. Problem Statement

Given a set of d independent input variables $x = [x_1, \dots, x_d] \in D$, with associated uncertainty bounds $D = [a_1, b_1] \times \dots \times [a_d, b_d] \subset \mathbb{R}^d$, the purpose of the proposed TMsim tool is to algebraically and parametrically calculate a set of output variables and their corresponding uncertainty bounds. Each variable f that depends on the inputs is understood to be represented according to the TM formulation [7]

$$f(x) \in T_f := P_f^n(x - x_0) + I_f, \quad (1)$$

where x_0 is the center of the domain D , P_f^n is a n th-order multivariate polynomial, and $I_f = [a, b]$ is an interval remainder such that

$$P_f^n(x - x_0) + a \leq f(x) \leq P_f^n(x - x_0) + b \quad \forall x \in D. \quad (2)$$

The TM of $f(x)$ is therefore guaranteed to lie between the upper and lower bound defined by the value of the polynomial P_f^n and the corresponding remainder I_f . Ideally, if $I_f = [0, 0]$, the polynomial P_f^n provides an exact parametric representation.

Moreover, the information on the overall (WC) bounds of $f(x)$ is readily obtained as $B(P_f) + I_f$, where $B(\cdot)$ denotes the bound operator; that is,

$$B(f(x)) = \left[\min_x \{f(x)\}, \max_x \{f(x)\} \right] \quad (3)$$

and the addition between intervals is intended in the IA-sense [4]; that is,

$$[a, b] + [c, d] = [a + c, b + d]. \quad (4)$$

The determination of the bound of a multivariate polynomial is a nontrivial task that is handled as detailed in Section 5.

In practice, a common maximum order n is assumed for each TM variable. Hence, for the ease of notation, the superscript n is dropped from now on when denoting the polynomial part of a TM.

3. Basic Scalar Operations

With the previously introduced definitions, basic scalar operations like addition and subtraction between TMs are readily computed as

$$T_f \pm T_g := P_f(x - x_0) \pm P_g(x - x_0) + I_f \pm I_g, \quad (5)$$

where the operation over the remainders is again performed according to the IA, with the subtraction being defined as

$$[a, b] - [c, d] := [a - d, b - c]. \quad (6)$$

The multiplication between two TMs yields

$$T_f \cdot T_g := P_{f \cdot g}(x - x_0) + P_e(x - x_0) + P_f(x - x_0)I_g + P_g(x - x_0)I_f + I_f I_g, \quad (7)$$

where $P_{f \cdot g}$ is the part of the product $P_f(x - x_0) \cdot P_g(x - x_0)$ up to order n , whereas P_e is the remaining higher-order contribution. The higher-order part and the remaining terms are encompassed into an interval remainder calculated as

$$I_{f \cdot g} = B(P_e) + B(P_f)I_g + B(P_g)I_f + I_f I_g, \quad (8)$$

where the product of two intervals is defined in IA as

$$[a, b] \cdot [c, d] := [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]. \quad (9)$$

Complex calculations are handled by suitably operating on the real and imaginary part with real-valued computations [10].

4. TMsim Overview

TMsim is a MATLAB tool that implements TM algebra and defines a specific class of variables for which the standard basic operations (see Section 3) as well as other available nonlinear functions or matrix operations (see Sections 6 and 7, respectively) are replaced by the corresponding TM calculations.

The tool requires an initialization that defines the order n of the TM representation and the number d of independent variables and creates some global auxiliary variables. An independent input variable is defined by its upper and lower bounds and the index $i \in \{1, \dots, d\}$. Each variable is represented as a structure with the field triplet (P, B, I) :

- (i) P is an array that collects the coefficients of the multivariate polynomial;

- (ii) B is an auxiliary two-element vector with the lower and upper bound of the polynomial;
- (iii) I is a two-element vector containing the lower and upper bound of the interval remainder.

To ease the calculation of the polynomial bounds (see Section 5), the coefficients of the polynomial are given in terms of rescaled variables $\tilde{x}_i = (2x_i - a_i - b_i)/(b_i - a_i) \in [-1, 1]$, $\forall i = 1, \dots, d$, instead of the original variables $x_i \in [a_i, b_i]$.

With the above definitions, the basic scalar operations are readily implemented as described in Section 3. These also include the raise to an integer power, which is implemented by means of iterative multiplications.

As a brief example, the code below defines two variables $x_1 = [2, 3]$ and $x_2 = [-1, 1]$ and calculates $y_1 = x_1^2 + x_2$, $y_2 = x_1 x_2^2$, and $z = x_2 \cdot y_1 + x_1 \cdot y_2$.

```
>> d = 2; % number of variables d_i=1,2
>> d_1 = 1; % index of variable x1
>> d_2 = 2; % index of variable x2
>> n = 3; % expansion order
>>
>> % Taylor model representation
>> initialize_TMsim(n,d);
>> x1=TaylorModel(2,3,d_1,'bounds');
>> x2=TaylorModel(-1,1,d_2,'bounds');
>> y1=x1^2+x2;
>> y2=x1*x2^2;
>> z=x1*y2+x2*y1;
```

The results are

$$\begin{aligned}
 y_1 &= \begin{cases} P = 6.25 + 2.5\tilde{x}_1 + \tilde{x}_2 + 0.25\tilde{x}_1^2, \\ B = [3, 10], \\ I = [0, 0] \end{cases} \\
 y_2 &= \begin{cases} P = 2.5\tilde{x}_2^2 + 0.5\tilde{x}_1\tilde{x}_2^2, \\ B = [-1, 3], \\ I = [0, 0] \end{cases} \\
 z &= \begin{cases} P = 6.25\tilde{x}_2 + 2.5\tilde{x}_1\tilde{x}_2 + 7.25\tilde{x}_2^2 + 0.25\tilde{x}_1^2\tilde{x}_2 + 2.5\tilde{x}_1\tilde{x}_2^2, \\ B = [-6.25, 18.75], \\ I = [-0.05, 0.25]. \end{cases}
 \end{aligned} \tag{10}$$

It is worth noting that the third-order representation is sufficient to represent y_1 and y_2 without error, but not z whose remainder is non-null (this would have been the case by setting $n = 4$). The estimated bound on z is $[-6.25, 18.75] + [-0.05, 0.25] = [-6.3, 19]$, which is a conservative estimation of the true bound $[-2.025, 19]$.

5. Polynomial Bounds

The exact calculation of the bounds of a multivariate polynomials can be performed exactly only for order $n \leq 2$. No rigorous approach exists for higher-order polynomials. TMsim implements a strategy, based on Bernstein polynomials, that allows for a conservative estimation of the bounds [7, 10]. Bernstein polynomials have the notable property of being bounded by their largest and smallest coefficient [13–16].

By means of a change of basis, a standard univariate polynomial $P(x) = \sum_{i=0}^n p_i x^i$ in the variable $x \in [a, b]$ is converted into the Bernstein form

$$P_B(x) = \sum_{i=0}^n b_i B_i(x), \tag{11}$$

where the Bernstein basis polynomials are defined as

$$B_i(x) = \frac{\binom{n}{i}}{(b-a)^n} (x-a)^i (b-x)^{n-i}, \quad 0 \leq i \leq n. \tag{12}$$

The corresponding Bernstein coefficients are

$$b_i = \sum_{j=0}^i \frac{\binom{i}{j}}{\binom{n}{j}} (b-a)^j \sum_{k=j}^n \binom{k}{j} a^{k-j} p_k, \tag{13}$$

from which

$$\begin{aligned}
 \min_i \{b_i\} &\leq \min_x \{P(x)\} \leq P(x) \leq \max_x \{P(x)\} \\
 &\leq \max_i \{b_i\}.
 \end{aligned} \tag{14}$$

The above expression emphasizes that the obtained bounds are not necessarily exact, but rather a conservative estimation of the actual bounds.

For the multivariate case, consider a polynomial in the form

$$P(x) = \sum_{\mathbf{k}} p_{\mathbf{k}} \prod_{i=1}^d \tilde{x}_i^{k_i}, \tag{15}$$

where $\tilde{x}_i \in [-1, 1] \forall i$ and $\mathbf{k} = [k_1, \dots, k_d]$ is a multi-index defining the exponents of each monomial, with $0 \leq k_i \leq n$ and $\sum_{i=1}^d k_i \leq n$. The vector of multivariate Bernstein coefficients is calculated as

$$\mathbf{b} = \sum_{\mathbf{k}} p_{\mathbf{k}} \cdot (\beta_{k_1} \otimes \beta_{k_2} \otimes \dots \otimes \beta_{k_d}), \tag{16}$$

where β_k is a vector with entries

$$\beta_{i,k} = \sum_{j=0}^{\min\{i,k\}} \frac{\binom{i}{j}}{\binom{n}{j}} 2^j \binom{k}{j} (-1)^{k-j}. \tag{17}$$

In order to conveniently calculate the Bernstein coefficients through (16), the independent variables are rescaled into the domain $D = [-1, 1]^d$. It is important to remark that, for a given order n , the vectors β_0 up to β_n for the conversion in (16) are predetermined and stored into a look-up table.

6. Nonlinear Functions

The TM resulting from the application of a nonlinear function $g(\cdot)$ is calculated by considering the *functional* Taylor expansion:

$$g(T_f(x)) = P_g(x) + I_g = \underbrace{\sum_{k=0}^n \frac{g^{(k)}(c_f)}{k!} (\bar{T}_f(x))^k}_{P_g(x) + I_p} + R_n(T_f(x)) \quad (18)$$

centered at $c_f = P_g(0)$. A shifted version $\bar{T}_f = T_f - c_f$ of the original TM has been introduced. Moreover, the function derivatives are defined as

$$g^{(k)}(c_f) := \left. \frac{dg(T_f(x))}{dT_f(x)} \right|_{x=c_f} \quad (19)$$

and $R_n(T_f(x))$ is the pertinent remainder in Lagrange or Cauchy formulation. It should be noted that the first term $P_g(x) + I_p$ in the r.h.s. of (18) merely involves sums and products of TMs, which are computed with the rules outlined in Section 3. The remainder I_p , resulting from these calculations, is added to the bound of the Lagrange/Cauchy remainder R_n to obtain the overall remainder I_g .

The Lagrange remainder of the n th-order Taylor expansion centered in x_0 of a function $g(x) \in C^{n+1}$ on $x \in D$ is defined as [17–19]

$$\begin{aligned} R_{L,n}(g(x)) &:= \frac{g^{(n+1)}(x^*)}{(n+1)!} (x - x_0)^{n+1} \\ &\equiv \frac{g^{(n+1)}(x_0 + \theta \bar{x})}{(n+1)!} \bar{x}^{n+1}, \end{aligned} \quad (20)$$

where $x^* \in [x_0, x] = x_0 + \theta \bar{x}$, with $\bar{x} = x - x_0$ and $\theta \in [0, 1]$.

An alternative formulation of the remainder is the Cauchy form, which reads [19]

$$\begin{aligned} R_{C,n}(g(x)) &:= \frac{(x - x^*)^n}{n!} (x - x_0) g^{(n+1)}(x^*) \\ &\equiv \frac{(1 - \theta)^n}{n!} \bar{x}^{n+1} g^{(n+1)}(x_0 + \theta \bar{x}). \end{aligned} \quad (21)$$

It is important to remark that both the Lagrange and the Cauchy remainder provide a conservative estimation of the truncation error. However, as described in the following, they have in general different bounds, and one of the two may even diverge as the order n is increased. Therefore, a single definition of the remainder does not guarantee a tight estimation of the error. For this reason, the TMsim tool calculates the remainder in both Lagrange and Cauchy form and selects the one with the smaller bound.

The remaining part of this section details the implementation of a representative set of general purpose nonlinear functions, including the logarithm, the square root, the multiplicative inverse, and the sin and cosine functions. The

implementation of the exponential function is well documented in the literature [7–9]. All the aforementioned functions are available in the current version of the tool. Other nonlinear functions can be implemented by following a similar procedure.

6.1. Logarithm. Consider the logarithmic function $g(T_f(x)) = \log(T_f(x))$. According to [7], the resulting TM $T_g(x)$ is computed as

$$\begin{aligned} \log(T_f(x)) &= \log(c_f) + \sum_{k=1}^n (-1)^{k+1} \frac{1}{k} \frac{\bar{T}_f^k(x)}{c_f^k} \\ &\quad + R_n(T_f(x)), \end{aligned} \quad (22)$$

where the remainder is given in the Lagrange formulation (20) as

$$R_{L,n}(T_f(x)) = \frac{(-1)^{n+2}}{n+1} \frac{\bar{T}_f^{n+1}(x)/c_f^{n+1}}{(1 + \theta \bar{T}_f(x)/c_f)^{n+1}}. \quad (23)$$

The bound of the remainder is obtained by means of the IA as follows:

$$B(R_{L,n}(T_f(x))) = \frac{(-1)^{n+2}}{n+1} \frac{B(\bar{T}_f(x))^{n+1}/c_f^{n+1}}{B(1 + \bar{T}_f(x)/c_f)^{n+1}}, \quad (24)$$

which is equivalent to

$$B(R_{L,n}(T_f(x))) = \frac{(-1)^{n+2}}{n+1} \frac{B(\bar{T}_f)^{n+1}}{B(T_f)^{n+1}}. \quad (25)$$

Unfortunately, the Lagrange remainder does not always converge to zero when the expansion order n is increased [19]. In particular, from (24),

$$\begin{aligned} R_{L,n}(T_f(x)) &\xrightarrow{n \rightarrow \infty} 0 \iff \\ \left| B\left(\frac{\bar{T}_f/c_f}{1 + \bar{T}_f(x)/c_f}\right) \right| &< 1 \end{aligned} \quad (26)$$

which implies that the Lagrange remainder converges only if

$$0 \leq \frac{\bar{T}_f(x)}{c_f} \leq 1. \quad (27)$$

However, since $\bar{T}_f(x)$ represents the variation of a parameter around its means value, its codomain can include negative values.

The Cauchy remainder (21) allows extending the convergence region, thus alleviating the limitations of the Lagrange remainder. Starting with the simple equality

$$\log(T_f(x)) = \log(c_f) + \log(1 + \bar{T}_f(x)), \quad (28)$$

where $\tilde{T}_f(x) = \bar{T}_f(x)/c_f$, the Cauchy remainder is expressed as

$$R_{C,n}(T_f(x)) = (-1)^{n+1} \left[\frac{(1-\theta)}{1+\theta\tilde{T}_f(x)} \right]^n \frac{\tilde{T}_f^{n+1}(x)}{1+\theta\tilde{T}_f(x)}. \quad (29)$$

Since $0 < \theta < 1$ and under the (quite practical) assumption $|\tilde{T}_f| < 1$ (i.e., the variation of a variable is smaller than its central value) [19],

$$\left| \frac{(1-\theta)}{1+\theta\tilde{T}_f(x)} \right| < 1 \quad (30)$$

which leads to

$$R_{C,n}(T_f(x)) \leq (-1)^{n+1} \frac{\tilde{T}_f^{n+1}(x)}{1+\theta\tilde{T}_f(x)}. \quad (31)$$

From the above expression,

$$R_{C,n}(T_f(x)) \xrightarrow{n \rightarrow \infty} 0 \iff |\tilde{T}_f(x)| \leq 1 \quad (32)$$

which implies that the convergence region is $-1 \leq \bar{T}_f(x)/c_f \leq 1$ as opposed to (27).

As in the case of the Lagrange remainder, the bound of the Cauchy remainder is calculated using IA as

$$B(R_{C,n}) = (-1)^{n+1} \frac{B(\tilde{T}_f(x))^{n+1}}{B(1+\tilde{T}_f(x))}. \quad (33)$$

6.2. Square Root. Consider now the square root function $g(T_f(x)) = \sqrt{T_f(x)}$. The resulting TM $T_g(x)$ is [7]

$$\sqrt{T_f(x)} = \sqrt{c_f} + \sum_{k=1}^n \frac{(-1)^{k-1} \sqrt{c_f} (2k-3)!!}{k! 2^k} \frac{\bar{T}_f^k(x)}{c_f^k} + R_n(T_f(x)), \quad (34)$$

where the remainder in Lagrange form is

$$R_{L,n}(T_f(x)) = C_L \frac{\bar{T}_f^{n+1}(x)/c_f^{n+1}}{(1+\theta\bar{T}_f(x)/c_f)^{n+1/2}}, \quad (35)$$

$$C_L = \frac{((-1)^n \sqrt{c_f} (2k-1)!!)}{((n+1)! 2^{n+1})}.$$

The bound of the above remainder is computed by mean of IA as

$$B(R_{L,n}(T_f(x))) = C_L \frac{B(\bar{T}_f(x))^{n+1}/c_f^{n+1}}{(1+B(\bar{T}_f(x)/c_f))^{n+1/2}}. \quad (36)$$

However, also in this case

$$R_{L,n}(T_f(x)) \xrightarrow{n \rightarrow \infty} 0 \iff 0 \leq \frac{\bar{T}_f(x)}{c_f} \leq 1. \quad (37)$$

The use of the Cauchy remainder, which reads [19]

$$R_{C,n}(T_f(x)) = C_C \frac{\tilde{T}_f^{n+1}(x)}{\sqrt{1+\theta\tilde{T}_f(x)}} \left(\frac{1-\theta}{1+\theta\tilde{T}_f(x)} \right)^n, \quad (38)$$

where

$$C_C = (-1)^n \sqrt{c_f} \frac{(2n-1)!! (n+1)}{(n+1)! 2^{n+1}}, \quad (39)$$

allows again mitigating the convergence issue. Under the assumptions that lead to (30), the corresponding bound is obtained as

$$R_{C,n}(T_f(x)) = C_C \frac{B(\tilde{T}_f(x))^{n+1}}{\sqrt{1+B(\tilde{T}_f(x))}}. \quad (40)$$

6.3. Multiplicative Inverse. Consider the function $g(T_f(x)) = 1/T_f(x)$. The resulting TM $T_g(x)$ is [7]

$$\frac{1}{T_f(x)} = \sum_{k=0}^n (-1)^k \frac{\bar{T}_f^k}{c_f^{k+1}} + R_n(T_f(x)), \quad (41)$$

where in principle the remainder can be computed via either the Lagrange or Cauchy formulation. However, in this special case, an *exact* remainder is calculated based on the properties of geometric series [17].

To begin with, the Taylor expansion of $1/(1-x)$ up to the order n is considered

$$\frac{1}{1-x} = \sum_{k=0}^n x^k + R_n(x). \quad (42)$$

The exact remainder for $|x| < 1$ is obtained via the identity

$$R_n(x) = \sum_{k=n+1}^{\infty} x^k = \lim_{N \rightarrow \infty} \frac{x^{n+1} - x^{N+1}}{1-x} = \frac{x^{n+1}}{1-x}. \quad (43)$$

The above equation is recast to obtain the remainder of $1/T_f(x)$ as

$$R_n(T_f(x)) = \frac{1}{c_f} \frac{\bar{T}_f^{n+1}(x)}{(1-\bar{T}_f(x))}, \quad (44)$$

where in this case $\tilde{T}_f = -\bar{T}_f/c_f$. The bound of the above remainder is computed as

$$B(R_n(T_f(x))) = \frac{1}{c_f} \frac{B(\bar{T}_f(x))^{n+1}}{B(1-\bar{T}_f(x))} \quad (45)$$

for $|\tilde{T}_f| < 1$.

6.4. Sine and Cosine. For the case of sine and cosine functions, the coefficients $g^{(k)}(c_f)$ in (18) are given by [7]

$$g^{(k)}(c_f) = \begin{cases} (-1)^{k/2} \sin(c_f) & k \text{ even} \\ (-1)^{(k-1)/2} \cos(c_f) & \text{otherwise,} \end{cases} \quad (46)$$

$$g^{(k)}(c_f) = \begin{cases} (-1)^{k/2} \cos(c_f) & k \text{ even} \\ (-1)^{(k-1)/2+1} \sin(c_f) & \text{otherwise,} \end{cases} \quad (47)$$

respectively.

The bounds of the Lagrange remainder are calculated as

$$B(R_n(T_f(x))) = \frac{B(g^{(n+1)}(T_f(x)))}{(n+1)!} B(\bar{T}_f(x))^{n+1}. \quad (48)$$

By substituting (46) or (47) in (48), the determination of the bounds requires the calculation of the bounds $B(\sin(T_f))$ or $B(\cos(T_f))$ in the pertinent domain of $T_f(x)$. The estimation of these bounds is complicated by the periodic behavior of the sine and cosine functions. Since the sine and cosine are bounded functions, a naive solution would be to simply assume $B(\sin(T_f)) = B(\cos(T_f)) = [-1, 1]$. However, tighter bounds can be obtained by considering the function as locally monotonic and by including the effect of a possible slope change only when a stationary point falls into the domain of $\zeta \in B(T_f(x))$. This leads to [11]

$$\begin{aligned} & \min \{\sin(\zeta)\} \\ &= \begin{cases} -1 & \frac{3\pi}{2} + 2\pi k \in B(T_f) \\ \min \{\sin(B(T_f(x)))\} & \text{otherwise} \end{cases} \quad (49a) \end{aligned}$$

$$\begin{aligned} & \max \{\sin(\zeta)\} \\ &= \begin{cases} +1 & \frac{\pi}{2} + 2\pi k \in B(T_f) \\ \max \{\sin(B(T_f(x)))\} & \text{otherwise} \end{cases} \quad (49b) \end{aligned}$$

$$\begin{aligned} & \min \{\cos(\zeta)\} \\ &= \begin{cases} -1 & \pi + 2\pi k \in B(T_f) \\ \min \{\cos(B(T_f(x)))\} & \text{otherwise} \end{cases} \quad (49c) \end{aligned}$$

$$\begin{aligned} & \max \{\cos(\zeta)\} \\ &= \begin{cases} +1 & 2\pi k \in B(T_f) \\ \max \{\cos(B(T_f(x)))\} & \text{otherwise,} \end{cases} \quad (49d) \end{aligned}$$

with $k \in \mathbb{Z}$.

6.5. Illustrative Example. As an illustrative example, the function $g(x) = \sin(1/\log(\sqrt{x}))$ with $x \in [3, 9]$ is considered. The corresponding TM is computed by means of the MATLAB code

```
>> initialize_TMsim(9,1);
```

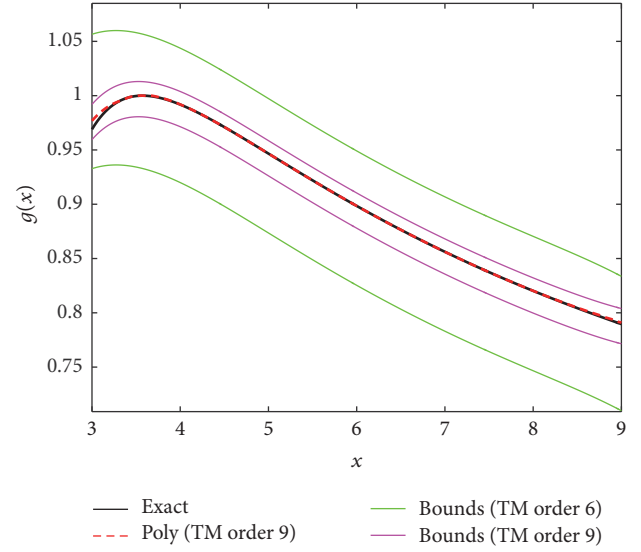


FIGURE 1: TM approximation of the function $g(x)$. Black line: actual value; dashed red line: polynomial approximation provided by a ninth-order TM; green and magenta lines: bounds obtained with orders 6 and 9, respectively.

```
>> x=TaylorModel(3,9,1,'bounds');
>> g=sin(inv(log(sqrt(x))));
```

The result is shown in Figure 1. The black curve is the actual value of the function $g(x)$. The dashed red curve is the approximation provided by the polynomial part of a ninth-order TM. Finally, the green and magenta lines are the upper and lower bounds obtained with the inclusion of the interval remainder for an expansion order of $n = 6$ and $n = 9$, respectively. It is appreciated that increasing the order yields tighter bounds. However, both bounds are conservative over the entire domain D of the input variable x .

7. Matrix Operations

A matrix TM is a matrix where all the entries are defined via the standard TM representation (1) and is here denoted by means of bold quantities as $\mathbf{T}_f(x) = \mathbf{P}_f(x) + \mathbf{I}_f$. The basic scalar calculations introduced in Section 3 are readily generalized to the corresponding matrix computations by operating element-wise. For instance, the product of two matrix TMs merely involves additions and multiplications of scalar TMs.

On the other hand, despite a generalization of the Taylor expansion to matrices being defined, the implementation of nonlinear matrix operations in the TM algebra is hindered by the fact that no corresponding definition of the remainder is available. A rigorous solution to handle the matrix inversion was proposed in [10] based on the Sherman-Morrison (SM) formula [20]. The implementation of other nonlinear matrix operations, such as the exponentiation, is still under investigation. An envisaged approach is to seek for an eigenvalue decomposition in TM algebra in order to reduce the problem to a scalar nonlinear operation over the eigenvalues.

7.1. Matrix Inversion. The SM formula provides an *exact* solution for the computation of the inversion $(\mathbf{A} + \mathbf{B})^{-1}$ when \mathbf{B} has unitary rank. Under this assumption, the resulting inverse matrix is

$$(\mathbf{A} + \mathbf{B})^{-1} = \mathbf{A}^{-1} - \frac{1}{1 + \alpha} (\mathbf{A}^{-1} \mathbf{B} \mathbf{A}^{-1}), \quad (50)$$

where $\alpha = \text{trace}(\mathbf{A}^{-1} \mathbf{B})$ and only the inversion of matrix \mathbf{A} is required.

For its application to a matrix TM \mathbf{T}_f , this is first split into a constant matrix and a shifted TM, as was done in Section 6 for the scalar case:

$$\mathbf{T}_f = \mathbf{c}_f + [\bar{\mathbf{P}}_f(x - x_0) + \mathbf{I}_f] = \mathbf{c}_f + \bar{\mathbf{T}}_f, \quad (51)$$

where $\mathbf{c}_f = \mathbf{P}_f(0)$ is a standard matrix and $\bar{\mathbf{P}}_f(x - x_0) = \mathbf{P}_f(x - x_0) - \mathbf{c}_f$. If the nonconstant part $\bar{\mathbf{T}}_f$ has unitary rank, then (52) is applied directly and the inverse TM is

$$\mathbf{T}_g = \mathbf{T}_f^{-1} = (\mathbf{c}_f + \bar{\mathbf{T}}_f)^{-1} = \mathbf{c}_f^{-1} - \frac{1}{1 + \alpha} (\mathbf{c}_f^{-1} \bar{\mathbf{T}}_f \mathbf{c}_f^{-1}), \quad (52)$$

with $\alpha = \text{trace}(\mathbf{c}_f^{-1} \bar{\mathbf{T}}_f)$. Hence, the SM formula reduces the inversion of a matrix TM to standard and available calculations, namely,

- (1) sums and multiplications of matrix TMs;
- (2) the inversion of the constant matrix \mathbf{c}_f , which is carried out according to classical algebra;
- (3) the calculation of the trace operator, which merely involves the scalar sum of the TMs on the matrix diagonal;
- (4) the calculation of the multiplicative inverse $1/(1 + \alpha)$, which is carried out as described in Section 6.3.

It is worth noting that in practice the matrix \mathbf{c}_f is always invertible as it corresponds to the matrix for the central value of the variable x .

The generalization of (52) to the case of full-rank matrices requires an iterative application of the algorithm [10]. The matrix $\bar{\mathbf{T}}_f$ is first split into the sum of rank-one matrices:

$$\bar{\mathbf{T}}_f = \bar{\mathbf{T}}_{f,1} + \bar{\mathbf{T}}_{f,2} + \bar{\mathbf{T}}_{f,3} + \dots, \quad (53)$$

where $\bar{\mathbf{T}}_{f,i}$ is defined as a matrix TM with null entries except for the i th column, which coincides with the i th column of the original TM $\bar{\mathbf{T}}_f$, thus being rank-1 by construction. The following iterative procedure is then put forward:

- (1) the inversion $(\mathbf{c}_f + \bar{\mathbf{T}}_{f,1})^{-1}$ is computed via (50) by considering $\mathbf{A} = \mathbf{c}_f$ and $\mathbf{B} = \bar{\mathbf{T}}_{f,1}$;
- (2) the inversion $(\mathbf{c}_f + \bar{\mathbf{T}}_{f,1} + \bar{\mathbf{T}}_{f,2})^{-1}$ is calculated by letting $\mathbf{A} = \mathbf{c}_f + \bar{\mathbf{T}}_{f,1}$ and $\mathbf{B} = \bar{\mathbf{T}}_{f,2}$. The inverse of \mathbf{A} , as needed in (50), is available from the previous step.

The above procedure is iterated until all terms in (53) are accounted for, thus yielding the end result.

7.2. Illustrative Example. The outlined procedure is general and allows inverting an arbitrary matrix whose elements are nonlinear functions. As an example, consider the following matrix:

$$\mathbf{M}(x) = \begin{bmatrix} \exp(1 + 0.3x) & \log(3 + x) & \sin\left(\pi + \frac{\pi}{4}x\right) \\ \log(3 + x) & \exp(1 + 0.3x) & \log\left(\pi + \frac{\pi}{4}x\right) \\ \sin\left(\pi + \frac{\pi}{4}x\right) & \log\left(\pi + \frac{\pi}{4}x\right) & \exp(1 + 0.3x) \end{bmatrix}, \quad (54)$$

where $x \in D = [-1, 1]$.

The above matrix belongs to the class of Toeplitz matrices, which are rather common in many fields of engineering as they result from Fourier expansions [21–23]. It is important to remark that $\det(\mathbf{M}(x)) \neq 0 \forall x \in [-1, 1]$.

The following MATLAB script calculates the inverse of $\mathbf{M}(x)$ by means of TMsim:

```
>> initialize_TMsim(3,1);
>> x_TM = TaylorModel(-1,1,1,'bounds');
>> a_TM = exp(1+0.3*x_TM);
>> b_TM = log(3+x_TM);
>> c_TM = sin(pi+pi/4*x_TM);
>> M_TM = [a_TM,b_TM,c_TM;
           b_TM,a_TM,b_TM;
           c_TM,b_TM,a_TM];
>> invM_TM = inv(M_TM);
```

The resulting matrix TM $\mathbf{T}_M^{-1}(x)$ provides a parametric representation of the nonlinear matrix $\mathbf{M}^{-1}(x)$ over the entire domain D and a conservative estimation of its upper and lower bounds. The minimum and maximum values of the entries of $\mathbf{M}^{-1}(x)$, estimated based on 10'000 MC samples, are

$$\begin{aligned} \min(\mathbf{M}^{-1}(x)) &= \begin{bmatrix} 0.3754 & -0.2252 & -0.1598 \\ -0.2252 & 0.4216 & -0.2252 \\ -0.1598 & -0.2252 & 0.3754 \end{bmatrix}, \\ \max(\mathbf{M}^{-1}(x)) &= \begin{bmatrix} 0.6055 & -0.1534 & 0.1487 \\ -0.1534 & 0.6022 & -0.1534 \\ -0.1487 & -0.1534 & 0.6055 \end{bmatrix}, \end{aligned} \quad (55)$$

respectively.

These bounds are also estimated from the TM by considering that $[\mathbf{M}(x)^{-1}]_{ij} \in B([\mathbf{T}_M^{-1}(x)]_{ij})$. This, for an expansion order $n = 6$, leads to

$$\begin{aligned} \min(B(\mathbf{T}_M^{-1}(x))) &= \begin{bmatrix} 0.3229 & -0.2831 & -0.1720 \\ -0.2855 & 0.3967 & -0.2820 \\ -0.1795 & -0.2863 & 0.3220 \end{bmatrix}, \\ \max(B(\mathbf{T}_M^{-1}(x))) &= \begin{bmatrix} 0.6230 & -0.1127 & 0.2542 \\ -0.1103 & 0.6614 & -0.1137 \\ 0.2617 & -0.1095 & 0.6238 \end{bmatrix}. \end{aligned} \quad (56)$$

Furthermore, Figure 2 compares the behavior of $[\mathbf{M}^{-1}(x)]_{11}$ (black line) with the polynomial model provided by a TM of order $n = 6$ (dashed red line). The bounds for expansion orders of $n = 3$ and $n = 6$ are indicated by the green and magenta curves, respectively. This example highlights the capability of TMsim to provide a parametric approximation of the entries of the inverse of a matrix with nonlinear elements as well as a conservative estimation of their upper and lower bounds.

It is interesting to calculate the product between \mathbf{T}_M and its inverse. In classical algebra, this would result in the identity matrix. With the proposed TM framework and an expansion order $n = 3$, the result is

$$\mathbf{T}_M(x) \cdot \mathbf{T}_M^{-1}(x) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{P}(x)} + \underbrace{\begin{bmatrix} \pm [1.262] & \pm [1.226] & \pm [1.190] \\ \pm [1.306] & \pm [1.305] & \pm [1.274] \\ \pm [1.289] & \pm [1.268] & \pm [1.252] \end{bmatrix}}_{\mathbf{I}}. \quad (57)$$

It is remarkable that the polynomial part still defines an identity matrix. This property stems from the exactness of the SM formula. All the approximations arising from the TM operations are included in the interval remainder. By increasing the expansion order, the interval remainder is reduced. For instance, using $n = 6$ yields

$$\mathbf{I} = \begin{bmatrix} \pm [0.095] & \pm [0.086] & \pm [0.075] \\ \pm [0.108] & \pm [0.096] & \pm [0.086] \\ \pm [0.106] & \pm [0.095] & \pm [0.087] \end{bmatrix}. \quad (58)$$

8. Application Example

In this section, the accuracy and the strength of TMsim are further investigated by considering two real-life application examples, namely, a quarter-car model and a continuous-time linear equalizer (CTLE). The code for these and other application examples is available online [12].

8.1. Quarter-Car Model. The first example considers the quarter-car model depicted in Figure 3 which is widely used for the analysis of the dynamical behavior of vehicle suspensions under the assumption that the front and the rear part of the vehicle are uncoupled. The model has two degrees of freedom, specifically, the vertical displacements of the chassis z_s and of the tire z_u from the reference frame. The road profile h is considered as the excitation of the system.

The proposed quarter-car model is defined by the following parameters, the masses of the chassis m_s and of the tire m_u , the spring coefficients of the suspension K and of the tire P , and the damping coefficients of the suspension C and of the tire C_t .

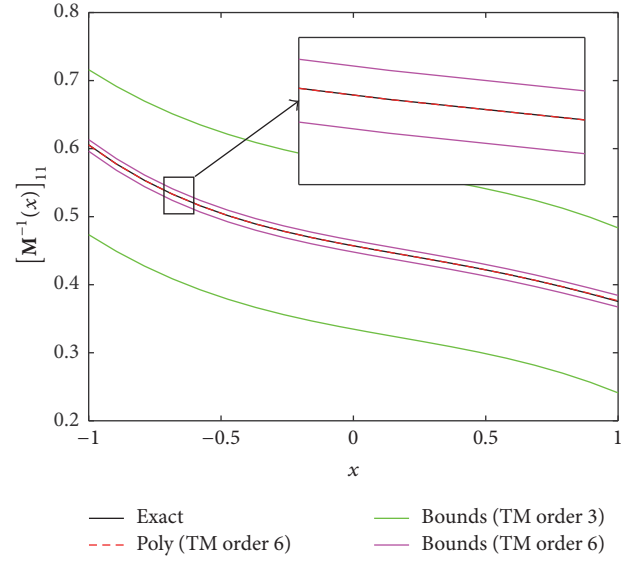


FIGURE 2: TM approximation of $[\mathbf{M}^{-1}(x)]_{11}$. Black line: actual value; dashed red line: polynomial approximation provided by a sixth-order TM; green and magenta lines: bounds obtained with orders 3 and 6, respectively.

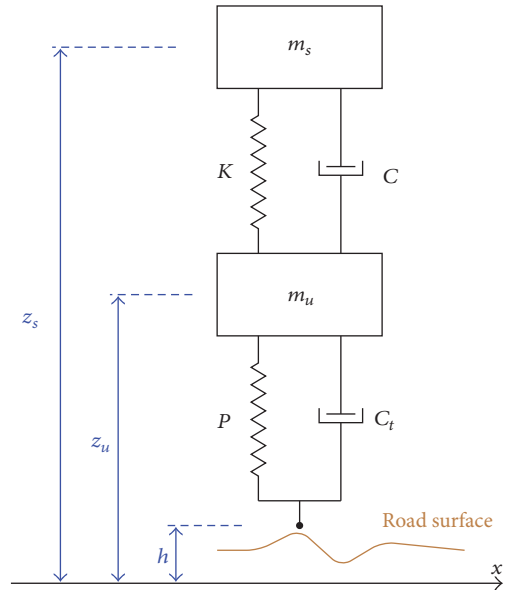


FIGURE 3: Schematic of the quarter-car model considered in the first application, with definition of parameters and state variables.

According to [24], the quarter-car model is described by the following matrix equation:

$$\begin{bmatrix} m_s & 0 \\ 0 & m_u \end{bmatrix} \begin{bmatrix} \ddot{z}_s(t) \\ \ddot{z}_u(t) \end{bmatrix} + \begin{bmatrix} C & -C \\ -C & C \end{bmatrix} \begin{bmatrix} \dot{z}_s(t) \\ \dot{z}_u(t) \end{bmatrix} + \begin{bmatrix} K & -K \\ -K & K + P \end{bmatrix} \begin{bmatrix} z_s(t) \\ z_u(t) \end{bmatrix} = \begin{bmatrix} 0 \\ C_t \dot{h}(t) + P h(t) \end{bmatrix}, \quad (59)$$

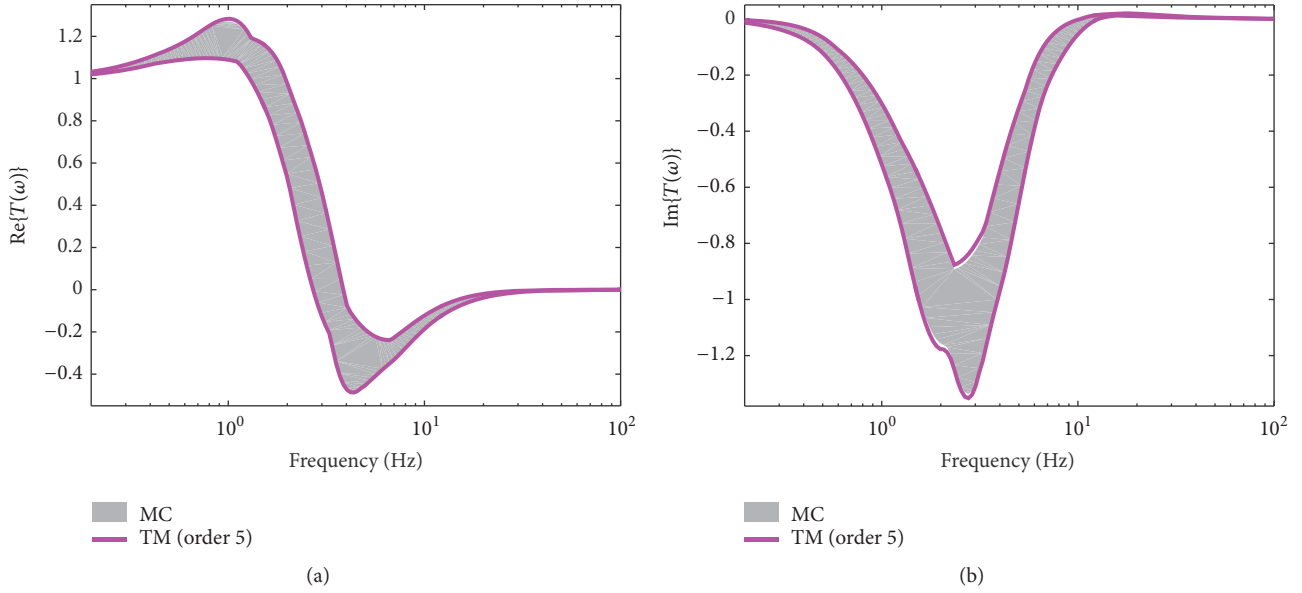


FIGURE 4: Real and imaginary parts of the transfer function $T(\omega) = Z_s(\omega)/H(\omega)$. Gray area: spread of the response resulting from parameter uncertainty and obtained with MC; magenta lines: WC bounds estimated from a fifth-order TM.

where z_s and z_u are the displacements from the static equilibrium position with respect to an inertial frame.

After some straightforward manipulation, the above equation is rewritten in its state-space controllable form, which reads

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (60)$$

where the column vector $\mathbf{x}(t) = [v_s(t), v_u(t), z_s(t), z_u(t)]^T \in \mathbb{R}^4$ collects the state variables of the system and $\mathbf{u}(t) = [h(t), \dot{h}(t)]^T \in \mathbb{R}^2$ is a vector defining the sources of the system. The matrix $\mathbf{A} \in \mathbb{R}^{4 \times 4}$ and the column vector $\mathbf{B} \in \mathbb{R}^{4 \times 2}$ are defined as

$$\mathbf{A} = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{C} & -\mathbf{M}^{-1}\mathbf{K} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad (61)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{M}^{-1}\mathbf{F} \\ \mathbf{0} \end{bmatrix},$$

where \mathbf{I} and $\mathbf{0}$ denote the identity matrix and the null matrix of dimensions 2×2 , respectively. The remaining terms are defined as follows:

$$\mathbf{M} = \begin{bmatrix} m_s & 0 \\ 0 & m_u \end{bmatrix} \quad (62a)$$

$$\mathbf{C} = \begin{bmatrix} C & -C \\ -C & C + C_t \end{bmatrix} \quad (62b)$$

$$\mathbf{K} = \begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \quad (62c)$$

$$\mathbf{F} = \begin{bmatrix} 0 & 0 \\ P & C_t \end{bmatrix}. \quad (62d)$$

The frequency-domain response of the system, denoted as $\mathbf{X}(\omega)$, is obtained via the Fourier transform of (3) and by considering as excitation vector $\mathbf{u}(t) = 1/(2\pi) \exp(j\omega t)[1, j\omega]^T$:

$$\mathbf{X}(\omega) = (j\omega\mathbf{I} - \mathbf{A})^{-1} \mathbf{B}\mathbf{U}(\omega), \quad (63)$$

where $\mathbf{U}(\omega) = [1, j\omega]^T$.

For the simulation, it is assumed $m_s = 332.5$ kg, $m_u = 38$ kg, and $C_t = 0$ Ns/m. In addition, three independent uncertainty variables are considered; that is, $C = (6000 \pm 20\%)$ Ns/m, $K = (22000 \pm 20\%)$ N/m, and $P = (20000 \pm 20\%)$ N/m.

Figure 4 shows the variation (gray area) of the real (top panel) and imaginary (bottom panel) part of the transfer function $T(\omega) = Z_s(\omega)/H(\omega)$, related to the state variable $Z_s(\omega)$, computed with 10'000 MC samples over 100 frequency points. The WC bounds obtained with a fifth-order analysis in TMsim (magenta lines) compare very well with the spread given by the MC samples. Figure 5 provides an analogous comparison for the magnitude of the transfer function. The excellent agreement validates the capability of TMsim to accurately deal with large parameter variations in solving a matrix system of equations.

8.2. Continuous-Time Linear Equalizer. The second proposed application example deals with the analysis of three CTLEs whose electrical circuits are depicted in Figure 6. CTLEs are commonly used in telecommunications to compensate for high-frequency channel losses with a “high-pass” type of response. The impact of parameter uncertainty in these CTLE configurations was analyzed in [25]. The circuit components are as follows:

- (a) $R = 35 \Omega \pm 20\%$, $C = 1$ pF $\pm 20\%$, $L = 3$ nH $\pm 20\%$;
- (b) $R = 100 \Omega \pm 20\%$, $C = 1$ pF $\pm 20\%$, $L = 3$ nH $\pm 20\%$;
- (c) $R = 100 \Omega \pm 20\%$, $C = 1$ pF $\pm 20\%$.

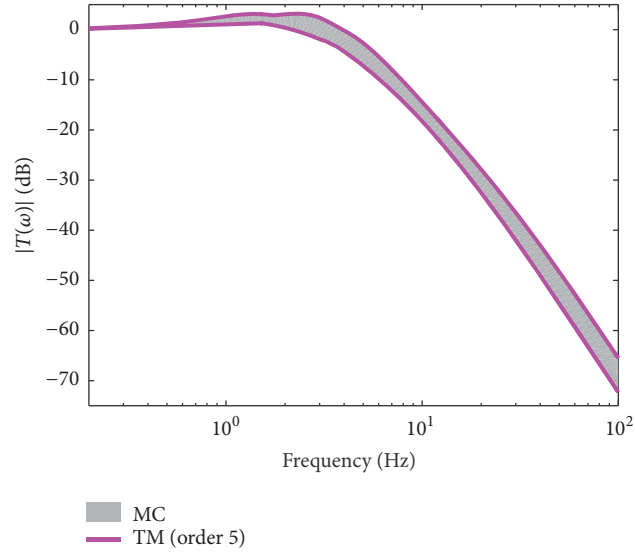


FIGURE 5: Magnitude of the transfer function $T(\omega) = Z_s(\omega)/H(\omega)$. Line identification is as in Figure 4.

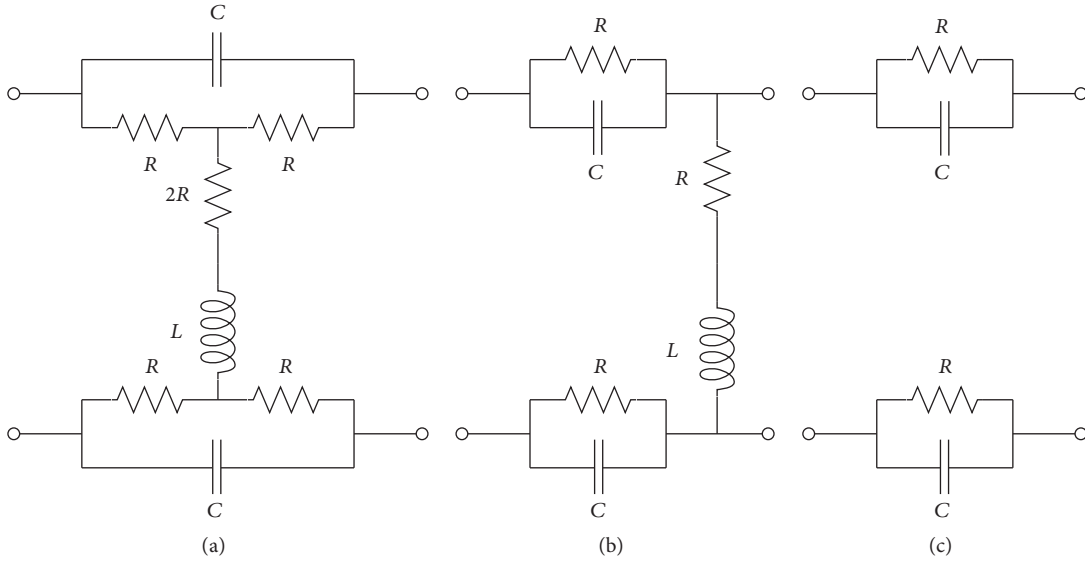


FIGURE 6: Schematics of the three different CTLE configurations considered.

The impact of the uncertainty in the circuit components on the transmission from the input port (left side) to the output port (right side), commonly denoted in electrical engineering as “ S_{21} ”, is investigated by analyzing the three circuits of Figure 6 by means of the modified nodal analysis (MNA) [10, 26]. The results for the real part, imaginary part, and magnitude of S_{21} for the three CTLE configurations are collected in the first, second, and third row of Figure 7, respectively. The gray areas indicate the spread of the response resulting from a MC analysis with 10'000 samples, whereas the magenta lines are the WC bounds computed from a fourth-order TM. This second example confirms the excellent accuracy provided by the TM analysis despite the large variability of the circuit components.

9. Conclusions

This paper presents an open source and freely available algebraic tool, named TMsim, that allows for the parametric and WC analysis of systems with bounded uncertain parameters. TMsim implements TM algebra, which represents uncertain variables as Taylor expansions complemented by an interval remainder accounting for approximation errors. While the Taylor expansion provides a parametric model of the variable, the information on the remainder allows obtaining a conservative estimation of the WC bounds. The TM representation is propagated from input to output uncertainties via a suitable redefinition of the linear and nonlinear operations involved in the system analysis.

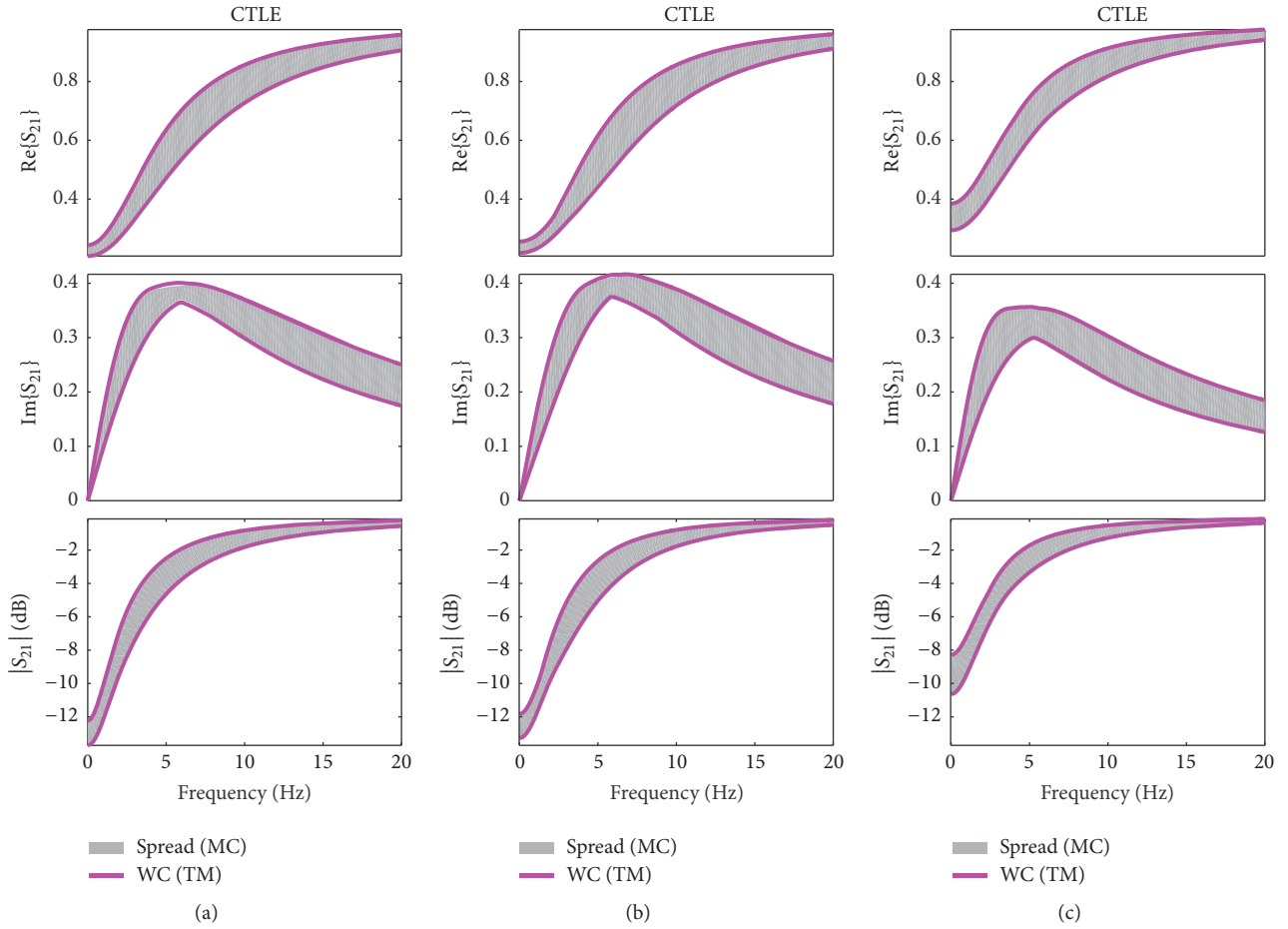


FIGURE 7: Real part, imaginary part, and magnitude of the transmission coefficient S_{21} in the three CTLE configurations of Figure 6. Gray area: spread of the response from a MC analysis. Magenta lines: WC bounds obtained with a fourth-order TM.

TMsim implements special improvements in the estimation of remainders and extensions to matrix operations. Specifically, a more accurate calculation of the remainders is obtained by choosing the tightest option between the Lagrange and Cauchy form. Moreover, an exact determination of the remainder is accomplished for the relevant case of the multiplicative inverse function. The tool also implements an effective algorithm for the inversion of a matrix TM.

The tool is validated by means of ad hoc illustrative examples as well as real-life application examples, that is, a quarter-car model and three CLTE electrical circuits. Excellent accuracy in the conservative estimation of the WC bounds of the output variables is achieved.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

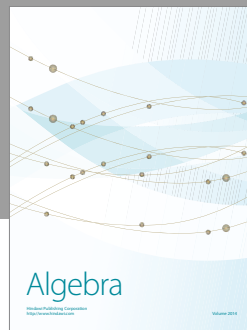
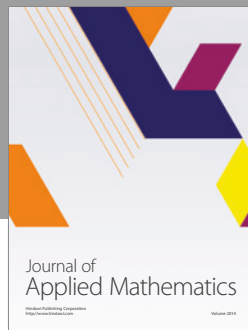
The authors would like to thank Luca Castellazzi, Politecnico di Torino, for providing the mechanical example on the quarter-car model. This work was supported in part by

the Research Foundation Flanders (FWO), of which Dr. Manfredi is a Postdoctoral Research Fellow.

References

- [1] G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer, New York, NY, USA, 1996.
- [2] D. Xiu and G. E. Karniadakis, "The Wiener-Askey polynomial chaos for stochastic differential equations," *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002.
- [3] R. G. Ghanem and P. D. Spanos, *Stochastic Finite Elements. A Spectral Approach*, Springer, New York, NY, USA, 1991.
- [4] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1966.
- [5] T. Ding, R. Trinchero, P. Manfredi, I. S. Stievano, and F. G. Canavero, "How affine arithmetic helps beat uncertainties in electrical systems," *IEEE Circuits and Systems Magazine*, vol. 15, no. 4, pp. 70–79, 2015.
- [6] L. H. de Figueiredo and J. Stolfi, "Affine arithmetic: concepts and applications," *Numerical Algorithms*, vol. 37, no. 1–4, pp. 147–158, 2004.
- [7] K. Makino and M. Berz, "Remainder differential algebras and their applications," in *Computational Differentiation: Techniques, Applications, and Tools*, pp. 63–74, SIAM, 1996.

- [8] M. Berz and G. Hoffstätter, "Computation and application of Taylor polynomials with interval remainder bounds," *Reliable Computing*, vol. 4, no. 1, pp. 83–97, 1998.
- [9] K. Makino and M. Berz, "Taylor models and other validated functional inclusion methods," *International Journal of Pure and Applied Mathematics*, vol. 4, no. 4, pp. 379–456, 2003.
- [10] R. Trinchero, P. Manfredi, T. Ding, and I. S. Stievano, "Combined parametric and worst case circuit analysis via Taylor models," *IEEE Transactions on Circuits and Systems. I. Regular Papers*, vol. 63, no. 7, pp. 1067–1078, 2016.
- [11] P. Manfredi, R. Trinchero, F. G. Canavero, and I. S. Stievano, "Application of Taylor models to the worst-case analysis of stripline interconnects," in *Proceedings of the 20th IEEE Workshop on Signal and Power Integrity (SPI '16)*, pp. 1–4, Turin, Italy, May 2016.
- [12] <https://www.tmsim.polito.it/>.
- [13] A. P. Smith, "Fast construction of constant bound functions for sparse polynomials," *Journal of Global Optimization*, vol. 43, no. 2-3, pp. 445–458, 2009.
- [14] J. Garloff, "Convergent bounds for the range of multivariate polynomials," in *Proceedings of the International Symposium on Interval Mathematics*, pp. 37–56, September 1985.
- [15] J. Garloff, "The Bernstein algorithm," *Interval Computations*, vol. 4, pp. 154–168, 1993.
- [16] G. G. Lorentz, *Bernstein Polynomials*, Chelsea, New York, NY, USA, 1986.
- [17] B. S. Thomson, J. B. Bruckner, and A. M. Bruckner, *Elementary Real Analysis*, Prentice Hall (Pearson), 2nd edition, 2008.
- [18] N. P. Bali and M. Goyal, *A Textbook of Engineering Mathematics*, Laxmi Publications, 2016.
- [19] B. C. Bhui and D. Chatterjee, *Mathematics I*, Vikas Publishing, 2014.
- [20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, New York, NY, USA, 3rd edition, 2007.
- [21] R. Trinchero, I. S. Stievano, and F. G. Canavero, "Steady-state response of periodically switched linear circuits via augmented time-invariant nodal analysis," *Journal of Electrical and Computer Engineering*, vol. 2014, Article ID 198273, 11 pages, 2014.
- [22] R. Trinchero, I. S. Stievano, and F. G. Canavero, "Steady-state analysis of switching power converters via augmented time-invariant equivalents," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, pp. 5657–5661, 2014.
- [23] R. Trinchero, P. Manfredi, I. S. Stievano, and F. G. Canavero, "Steady-state analysis of switching converters via frequency-domain circuit equivalents," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 8, pp. 748–752, 2016.
- [24] G. Genta, *Motor Vehicle Dynamics: Modeling and Simulation*, vol. 43 of *Series on Advances in Mathematics for Applied Sciences*, Prentice-Hall, Upper Saddle River, NJ, USA, 1966.
- [25] J. B. Preibisch, T. Reuschel, K. Scharff, and C. Schuster, "Impact of continuous time linear equalizer variability on eye opening of high-speed links," in *Proceedings of the 20th IEEE Workshop on Signal and Power Integrity (SPI '16)*, pp. 1–4, Turin, Italy, May 2016.
- [26] C.-W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems*, vol. 22, no. 6, pp. 504–509, 1975.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

