A Robust and Fast Convergent Interval Analysis Method for the Calculation of Internally Controlled Switching Instants

Nicola Femia

Abstract— **An interval analysis-based method for the root-finding of nonmonotonic polynomials is presented in this paper. It has been developed for numerical time-domain analysis of switched nonlinear networks, where internally controlled switching instants must be calculated as zeros of strongly nonmonotonic nonlinear functions. The method in based on an interval extension of Newton's operator resulting from the application of the mean-value theorem (m.v.t.) at the highest order to the polynomials whose zeros are sought. It is demonstrated that such interval extension is the most efficient one with respect to not only all those derived from the application of m.v.t. at any order lower than the maximum one but also to that one obtained with the centered form of the first derivative of the polynomial. A recursive algorithm for roots finding is presented which uses this optimal interval Newton's contraction mapping. Some examples drawn from switching converters time-domain analysis are proposed to outline the robustness and the sharp convergence of the method and its improvements with respect to other interval operators.**

I. Introduction

ONE of the main problems encountered in numerical time-domain analysis of switched nonlinear networks consists in the calculation of the first (or earliest) zero t^* of a strongly nonlinear function $f(t)$, corresponding to a current or voltage, within a given time interval $I = [t_1, t_2]$. The interest in the first zero t^* only comes from the fact that t^* may be the instant where the state of some controlled switch (BJT, MOSFET, GTO, IGBT, etc.) or uncontrolled switch (diodes) changes because of either (a) a zero-voltage or a zero-current condition (spontaneous commutation) or (b) the action of a feedback circuitry (forced commutation) which suddenly drives a switch commutation if some voltage or current exceeds a given threshold. In both cases the first zero t^* of some voltage or current $f(t)$ corresponds to an Internally Controlled Switching Instant (ICSI), after which a new circuital configuration is entered which is modeled by a new set of equations. Some eventualities, occurring very frequently in switching converters and regulators, dramatically reduce the possibilities of computing ICSI's by means of the numerical methods based on real analysis: i) it is not known *a priori* if I includes zeros of $f(t)$; ii) $f(t)$ may be nonmonotonic within I; iii) I may contain more than one zero of $f(t)$; iv) the values of $f(t)$ at the extremes of I may have the same sign. Consequently, Newton's, bisection and other methods working on real funtions can merely fail, unless either an optimal guess is found or a large amount of computations is accepted. But, in addition to the uncertainty of the former hypothesis, the complexity of modern switching circuits makes unacceptable also the latter one if fast numerical simulation are looked for. Furthermore, some chaotic behaviors have been recently detected in switched nonlinear networks that can be properly observed only if a large number of switching periods (even thousands!) is run.

Manuscript received October 25, 1993; revised January 16, 1995 and May 17, 1995. The work was supported by M.U.R.S.T. 40% and 60% funds. This paper was recommended by Associate Editor A. Premoli.

The author is with the Dipartimento di Ingegneria Elettronica, Università di Salerno, I-84084, Fisciano (SA) Italy.

Publisher Item Identifier S 1057-7122(96)01363-3.

In this paper the author shows how Interval Analysis (IA) can be applied to solve efficiently the root-finding problem framed above. By means of IA robust and fast convergent algorithms can be settled which do not require guess solutions at all, even in presence of strong nonmonotonic nonlinearities and of the conditions i)–iv) listed above. In the books [1]–[9] an exhaustive treatment of IA theory and applications may be found. The essence of IA-based computations lies in the seek for the closest upper and lower extremes which bound the interval of values where the exact result of a computation certainly lies. Working with IA offers the advantage of getting as a result of each computation two informations at a time: an approximated value of the solution and an estimation of the actual worst case error of the computation. Major efforts have been made, since the birth of IA as an autonomous discipline around late sixties, to identifiy optimal interval forms, operators and decomposition techniques enabling the minimization of the width of intervals including the range of rational functions (see for example [15], [16]). Such topic is indeed of crucial importance for making interval computation-based algorithms efficient. IA has encountered until today large interest in the framework of numerical solution of nonlinear problems and it has been applied also to problems concerning analysis and design of linear and nonlinear circuits. Nowadays some programming languages including reliable intervals handling capabilities are also available [6]. The efficiency of interval computations is strictly connected to machine numbers rounding and is much sensitive to the form adopted for the calculation of the interval extension of rational and irrational functions. A large number of papers in literature is devoted to this last aspect (see for example [10]–[14]).

In Section II an algorithm for the first-zero-search is proposed, based on Newton's interval operator for interval con-

1057–7122/96$05.00 © 1996 IEEE

traction mapping. In Section III some fundamental properties of Newton's operator and the optimality of its maximum order m.v.t. form are outlined. In Section IV some applications of the method are proposed to show its performances. In the Appendix a simple criterion for the automatic calculation of centered forms of polynomials is given. Readers are supposed to know at least the elementary concepts of IA, otherwise they are addressed to the referenced literature.

It is worth remarking that numerical computation of ICSI's, which are zeros of currents and/or voltages whose waveforms are real-valued combinations of complex exponential functions [18], requires that any function is in polynomial form. Hence, in the following, we shall consider the problem of root finding of polynomials assuming that a suitable method for polynomial approximation of functions (e.g., Padé approximants or Chebyshev polinomials) is available.

II. First Root Finding of Polynomials by Means of Interval Contraction Mapping

Given the real polynomial $p(t) = p_0 + p_1 t + \cdots + p_N t^N$ in the real variable t let us suppose to be interested in the calculation of its *first root* t^* within the interval $I = [a, b]$: $t^* = \min\{t \in I | p(t) = 0\}$. Let $\overline{p}(I)$ and $\overline{p'}(I)$ be, respectively, the range of the polynomial $p(t)$ and of its first derivative $p'(t)$ on I. From the application of m.v.t. it follows that $t^* = \tau - p(\tau)/p'(\tau + \theta(t^* - \tau))$, for any $\tau \in I$ and for some $\theta \in [0, 1]$. Since neither the exact value of θ nor t^* are known, the *interval extension*

$$N(I) = \tau - p(\tau)/P'(I) \quad (1)$$

can be introduced, which represents an interval version of Newton's operator, where P' is an interval extension of p' on I. The interval Newton's operator (1) is defined provided the condition $0 \notin P'(I)$ is fulfilled. The point τ is usually chosen as the middle point of $I, m(I)$. Since $\overline{p'}(I) \subset P'(I)$ (see [1]) it follows that $t^* \in N(I)$. Consequently, given $I_0 = I$, the operator (1) can be used for the iterative *interval contraction mapping* towards t^* defined by the recursive formula:

$$I_n = I_{n-1} \cap N(I_{n-1}), \qquad n = 1, 2, \cdots. \quad (2)$$

The general properties of the contraction mapping (2) are deeply discussed in the books listed in the references. Herein we restrict the attention to few aspects which are important for the convergence in the root-finding problem (proofs can be found in [1]):

1) if an interval I contains a simple root t^* of p, then $t^* \in N(I)$;
2) whenever $N(I)$ is defined, i.e., $0 \notin P'(I)$, then either $N(I) \cap I = \emptyset$, in which case I does not contain a zero of p, or $N(I) \cap I$ is a nonempty interval which contains a zero of p if I does.

The property 1) assures that zeros can be always found by means of interval Newton's method, when they exist. In this case it has been shown ([1]) that the convergence of Newton's contraction mapping (2) is quadratic in exact interval arithmetic. The property 2) is very important because it can be used as an efficient nonexistence test of zeros. Such preliminar operation is really useful for fast circuit time-domain simulation since it permits to skip the root search if there are not zeros, i.e., internally controlled switching instants, in the interval under inspection. It has been shown in [17] that the convergence of interval Newton's method is quadratic even for this exclusion property.

From above it is clear that the condition $0 \in P'(I)$ is crucial since it makes $N(I)$ undefined. This can be the consequence of two distinct situations: (a) $0 \in \overline{p}'(I)$ or (b) $0 \notin \overline{p}'(I)$ but $0 \in P'(I)$. In the case (a) $p(t)$ is not monotonic on I and this requires a segmentation of I in two subintervals I_L and I_R such that $I_L \cup I_R = I$ and $I_L \cap I_R = \emptyset$. The iterative mapping (1), (2) can restart from I_L. If I_L does not contain zeros of p then I_R is inspected and further interval decompositions must be resorted to whenever Newton's operator (1) is not defined. In the case (b) the interval decomposition would not be stricIty necessary and it could be avoided by adopting a sufficiently narrow interval extension $P'(I)$, which bounds the exact range $\overline{p}'(I)$ as closely as possible. For this reason the miminization of the interval extensions of polynomials is always looked for. We will consider such problem in the next section. A further suppression of superfluous interval segmentations can be gained if a left-oriented self-calling strategy is adopted, according to the flow diagram of Fig. 1. Such algorithm, called in the sequel *first_root_search*(p, I), seeks the first zero of the polynomial p in I and stops when the *relative width* $\Delta_n = w(I_n)/m(I_n)$ of the last interval computed I_n, which includes the solution t^*, is smaller than the required one Δ. Some elements of the algorithm are worth to be discussed. If *path 2* is entered then a sequence of interval segmentations and *nested self calls* to *first_root_search* will occur until *path 1* is run. Throughout such *zooming* in a *left-priority* criterion is followed, namely after each interval segmentation, the left half is always inspected first. If the zero is found then the algorithm stops, otherwise the right half is inspected. If no zeros are found within a right half I_r of a segmented interval at any hierarchical level of recursive calls to *first_root_search*, then two possibilities arise: if the right end of I_r corresponds to the right end of I, then stop because I does not contain zeros of $p(t)$; otherwise move backwards and run a *first_root_search* call on the lowest order right-side subinterval among the preceding levels of segmentation. Such mechanism is illustrated in Fig. 2, where all the left and right halves of each interval up to the third segmentation level have been depicted for easy of explanation, while it must be clear that segmentations are actually performed only when needed.

III. Higher Order m.v.t. Forms of Interval Polynomials and Newton's Operator

The amount of segmentations and the number of iterations required by the the root search method illustrated above can be sensibly reduced by minimizing the width $w(N(I))$ of the interval Newton's operator (1). In addition to the basic form

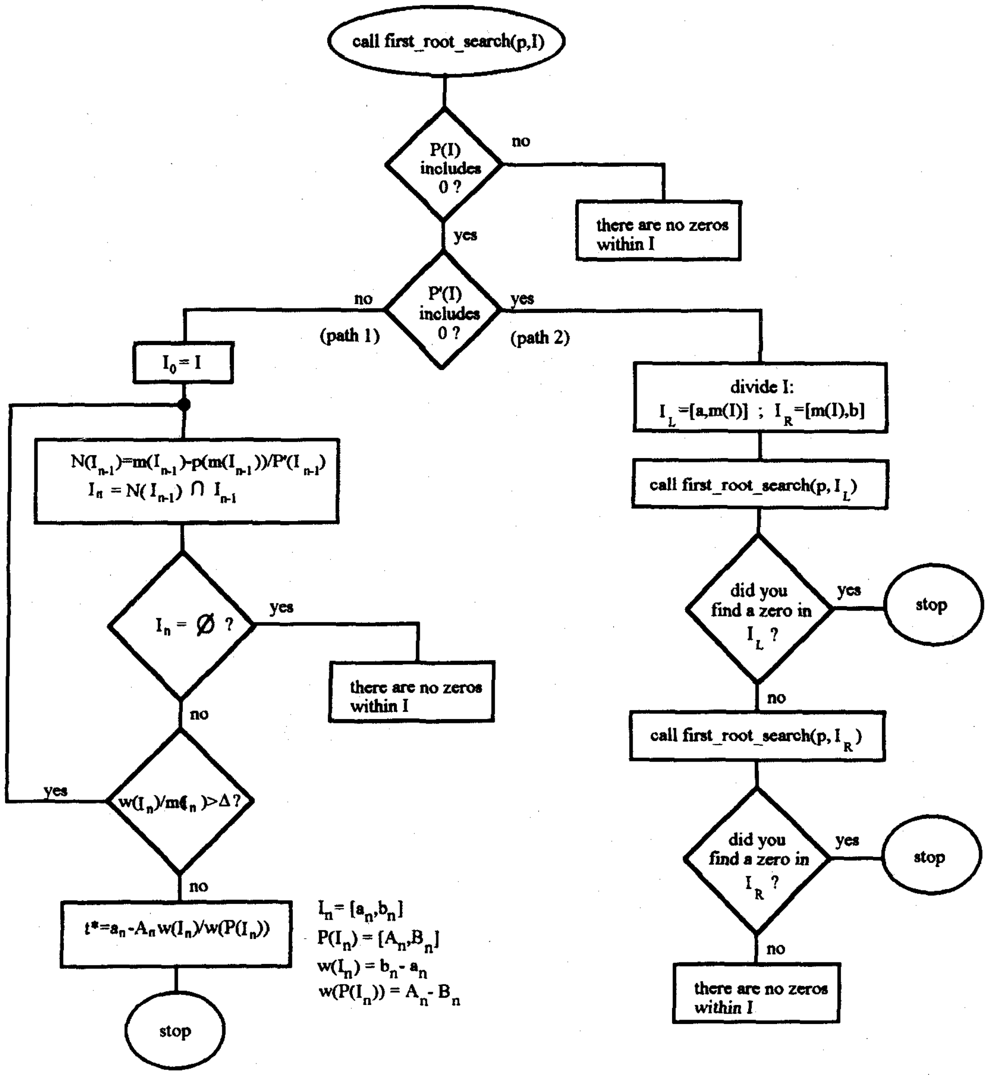


Fig. 1. Block diagram of the recursive algorithm for the search of the first root of $p(t)$ within I.

(1) a number of modified versions of Newton's operator and other interval operators for contraction mappings have been proposed in literature. One of the most popular is Krawczyk's operator ([10], [11]), defined as follows:

$$K(I) := \tau - \frac{p(\tau)}{p'(\tau)} + \left(1 - \frac{P'(I)}{p'(\tau)}\right)(I - \tau), \qquad \tau = m(I)$$

Other versions of Krawczyk's operator have been also proposed for systems of nonlinear equations in more than one variable ([12], [13]). Krawczyk's operator brings the advantage of presenting the interval extension $P'(I)$ at the numerator of a fraction; however, the condition $0 \notin P'(I)$ must be fulfilled to get the inclusion $K(I) \cap I \subset I$, which is necessary for interval contraction. The performances of Krawczyk's and other operators have been considerably experienced especially in the framework of tests of existence, uniqueness and convergence in nonlinear problems. They are somehow more complicated than basic Newton's operator (1) but offer small width and avoid the presence of the inverse interval $(P'(I))^{-1}$, even if segmentations are required whenever $0 \in P'(I)$. Newton's operator may offer indeed faster convergence if an appropriate form for $P'(I)$, or even an alternative interval extension, is adopted in (1). For example, Oliveira has proposed in [14] an alternative form of interval Newton's operator where the interval extension $O(I) = p'(\tau) + 1/2(P''(I)(I - \tau))$ is used instead of $P'(I), P''(I)$ being the interval extension of the second derivative p''. $O(I)$ can be obtained by applying m.v.t. at second-order to the polynomial p. It makes the width of Newton's operator $N(I)$ narrower than that of Krawczyk's one $K(I)$. Notice that the above interval extension $O(I)$ does not correspond to the interval extension which can be obtained by applying the m.v.t. at first-order to the derivative $p', P'(I) = p'(\tau) + P''(I)(I - \tau)$. In this section we will stress this last point and will show that the improvement of Newton's operator can be further sharpened, accounting for some properties of interval

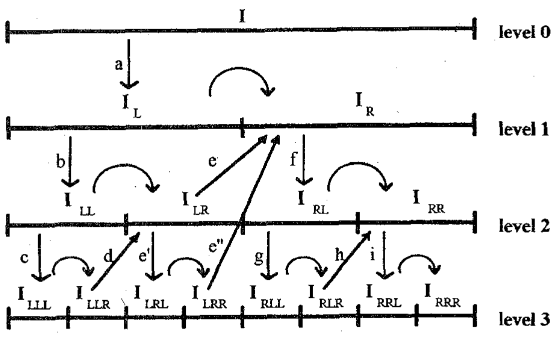


Fig. 2. Search mechanism with selective interval segmentation.

extensions obtained by means of m.v.t.. Let define the nth *order m.v.t. form* of a polynomial of degree N as follows:

$$p(t) = \sum_{k=0}^{n-1} \frac{1}{k!} p^{(k)}(\tau)(t-\tau)^k + \frac{1}{n!} p^{(n)}(\tau + \theta(t-\tau))(t-\tau)^n \quad (3)$$

$\forall t \in I$ and for some $\theta \in [0,1]$, with $\tau = m(I)$ and $n \leq N$. If $p(t^*) = 0$, then from (3) we have (see below).

As far as $n < N$, the exact value of θ which fulfils the (3) is unknown, while t^* is the unknown zero that we are seeking. Consequently, the nth *order m.v.t.* interval Newton's operator can be defined as follows:

$$N_n(I) = \tau - \frac{p(\tau)}{\sum_{k=1}^{n-1} \frac{1}{k!} p^{(k)}(\tau)(I-\tau)^{k-1} + \frac{1}{n!} P^{(n)}(I)(I-\tau)^{n-1}} \quad (4)$$

$N_n(I)$ is a more general form of Newton's operator (1) and includes as special cases both the basic operator (1), for $n = 0$, and Oliveira's one, for $n = 1$. Since the interval extension $P^{(n)}$ of the nth derivative $p^{(n)}$ in (4) is a rational interval function of degree $N-n$, it might be expected that $w(N_n(I))$ decreases as n increases. But this is not always true. In fact, let us consider the nth and $(n+q)$th-order m.v.t. forms of interval Newton's operator and interval extension of the polinomial p, for any positive integer q such that $(n+q) \leq N$

$$\begin{aligned} N_n(I) &= \tau - p(\tau)\left(\sum_{k=1}^{n-1} \frac{p^{(k)}(\tau)}{k!} R^{k-1} + \frac{P^{(n)}(I)}{n!} R^{n-1}\right)^{-1} \\ &= \tau - p(\tau)[\overline{P}'_{n-1}(I)]^{-1} \\ P_n(I) &= p(\tau) + \left(\sum_{k=1}^{n-1} \frac{p^{(k)}(\tau)}{k!} R^{k-1} + \frac{P^{(n)}(I)}{n!} R^{n-1}\right) R \\ &= p(\tau) + [\overline{P}'_{n-1}(I)]R \end{aligned} \quad (5)$$

$$\begin{aligned} N_{n+q}(I) &= \tau - p(\tau)\left(\sum_{k=1}^{n+q-1} \frac{p^{(k)}(\tau)}{k!} R^{k-1} + \frac{P^{(n+q)}(I)}{(n+q)!} R^{n+q-1}\right)^{-1} \\ &= \tau - p(\tau)[\overline{P}'_{n+q-1}(I)]^{-1} \\ &= \tau - p(\tau)\left(\sum_{k=1}^{n-1} \frac{p^{(k)}(\tau)}{k!} R^{k-1} + \frac{1}{n!} \cdot \left(\sum_{k=n}^{n+q-1} \frac{n! p^{(k)}(\tau)}{k!} R^{k-n} + \frac{n! P^{(n+q)}(I)}{(n+m)!} R^q\right) R^{n-1}\right)^{-1} \\ P_{n+q}(I) &= p(\tau) + \left(\sum_{k=1}^{n+q-1} \frac{p^{(k)}(\tau)}{k!} R^{k-1} + \frac{P^{(n+q)}(I)}{(n+q)!} R^{n+q-1}\right) R \\ &= p(\tau) + [\overline{P}'_{n+q-1}(I)]R \end{aligned} \quad (6)$$

where $R = [-r, r] = (I - \tau) = (I - m(I))$. The widths of interval extensions $N_n(I), P_n(I)$, and $N_{n+q}(I), P_{n+q}(I)$ decrease according to the widths of interval extensions $\overline{P}'_{n-1}(I)$ and $\overline{P}'_{n+q-1}(I)$, respectively. The interval extension

$$\overline{P}'_{n-1}(I) = \sum_{k=1}^{n-1} \frac{p^{(k)}(\tau)}{k!} R^{k-1} + \frac{P^{(n)}(I)}{n!} R^{n-1}$$

deserves some attention. It is obtained from the nth-order m.v.t. form $P_n(I)$ by cutting off the term $p(\tau)$ and shifting the coefficients of powers from 1 to n into the coefficients of powers from 0 to $n-1$. Consequently, the coefficients of $\overline{P}'_{n-1}(I)$ are all smaller than the corresponding ones of the $(n-1)$th-order m.v.t. form of the derivative p',

$$P'_{n-1}(I) = \sum_{k=1}^{n-1} \frac{p^{(k)}(\tau)}{(k-1)!} R^{k-1} + \frac{P^{(n)}(I)}{(n-1)!} R^{n-1}$$

and then

$$\begin{aligned} w(\overline{P}'_{n-1}) &= \sum_{k=1}^{n-1} \frac{|p^{(k)}(\tau)|}{k!} r^{k-1} + \frac{w(P^{(n)}(I))}{n!} r^{n-1} \\ &< \sum_{k=1}^{n-1} \frac{|p^{(k)}(\tau)|}{(k-1)!} r^{k-1} + \frac{w(P^{(n)}(I))}{(n-1)!} r^{n-1} \\ &= w(P'_{n-1}) \end{aligned} \quad (7)$$

$\forall n \leq N$. For this reason we shall call $\overline{P}'_{n-1}(I)$ *reduced* $(n-1)$*th order m.v.t. interval extension of* p'. To compare the

$$t^* = \tau - \frac{p(\tau)}{\sum_{k=1}^{n-1} \frac{1}{k!} p^{(k)}(\tau)(t^*-\tau)^{k-1} + \frac{1}{n!} p^{(n)}(\tau + \theta(t^*-\tau))(t^*-\tau)^{n-1}}$$

widths of $N_n(I), N_{n+q}(I), P_n(I)$ and $P_{n+q}(I)$ we adopt the highest order m.v.t. form (centered form) for $P^{(i)}(I)$, namely

$$P^{(i)}(I) = \sum_{k=0}^{N-i} \frac{p^{(k+i)}(\tau)}{k!} R^k, \qquad i = n, n+q$$

and assume

$$\Phi = P^{(n)}(I);$$

$$\Psi = \sum_{k=n}^{n+q-1} \frac{n! p^{(k)}(\tau)}{k!} R^{k-n} + \frac{n! P^{(n+q)}(I)}{(n+q)!} R^q.$$

Since $[a,b] \cdot R^k = [a,b] \cdot [-r^k, r^k] = [-r', r'], \forall a, b$ and k, we get

$$w(\Phi) = 2 \sum_{k=1}^{N-n} \frac{|p^{(k+n)}(\tau)|}{k!} r^k \tag{8}$$

$$\begin{aligned} w(\Psi) &= 2\left(\sum_{k=n+1}^{n+q-1} \frac{n! |p^{(k)}(\tau)|}{k!} r^{k-n} + \frac{n!}{(n+q)!} r^q \right. \\ &\quad \left. \cdot \sum_{k=0}^{N-n-q} \frac{|p^{(k+n+q)}(\tau)|}{k!} r^k \right) \\ &= 2\left(\sum_{k=1}^{q-1} \frac{n!}{(k+n)!} |p^{(k+n)}(\tau)| r^k \right. \\ &\quad \left. + \sum_{k=q}^{N-n} \frac{n!}{(n+q)!(k-q)!} |p^{(k+n)}(\tau)| r^k \right). \end{aligned} \tag{9}$$

Due to the equality $m(\Phi) = m(\Psi) = p^{(n)}(\tau)$ and to the property of inclusion monotonicity of interval arithmetics [1], we have $w(N_{n+q}(I)) < w(N_n(I))$ and $w(P_{n+q}(I)) < w(P_n(I))$ if and only if $w(\Psi) < w(\Phi)$. The first sum appearing in the expression of $w(\Psi)$ is always smaller than the sum of the first $q-1$ terms of $w(\Phi)$, as

$$\frac{n!}{(k+n)!} \le \frac{n!}{k! n!} = \frac{1}{k!}.$$

The second sum in the expression of $w(\Psi)$, instead, may be greater than the sum of the corresponding terms in the expression of $w(\Phi)$, as the inequality

$$\frac{n!}{(k-q)!(n+q)!} \le \frac{n!}{k! n!} = \frac{1}{k!}$$

is not necessarily fulfilled, unless $n+q = N$, namely when the m.v.t. is applied up to the maximum order (the degree N of $p(t)$). In this case, in fact, the second sum in the expression of $w(\Psi)$ reduces to one term only, corresponding to $k = N - n$, so that

$$\frac{n!}{(k-q)!(n+q)!} = \frac{n!}{N!} < \frac{n!}{(N-n)! n!} = \frac{1}{k!}$$

and then $w(\Psi) < w(\Phi)$. Notice that if N is large there are more possibilities that $w(\Psi) > w(\Phi)$, especially if n is small. This is the case encountered in the time domain simulation of switching converters, where voltage and current waveforms may have a bandwidth next to (or even greater than) the switching frequency so that high order approximating polynomials are required for correct waveforms representation.

From above we conclude that higher-order m.v.t. forms of polynomial interval extensions and interval Newton's operator are not always better than first-order ones, unless the maximum order N corresponding to the polynomial degree is resorted to. Furthermore, the maximum order m.v.t. interval extension $P_N(I)$ corresponds to a *centered form*, whose optimality for bounding the range of polynomials has been already shown in literature ([15], [16]), whereas the optimal Newton's operator (4) defined according to the application of m.v.t. at maximum order to the polynomial p is obtained by means of the reduced $(N-1)$th-order m.v.t. form

$$\overline{P}'_{N-1}(I) = \sum_{k=1}^{N-1} \frac{p^{(k)}(\tau)}{k!} R^{k-1}$$

instead of the $(N-1)$th-order m.v.t. form (centered form)

$$P'_{N-1}(I) = \sum_{k=1}^{N-1} \frac{p^{(k)}(\tau)}{(k-1)!} R^{k-1}$$

of the derivative p'. The improvement achieved using $\overline{P}'_{N-1}(I)$ increases with the polynomial degree N. In conclusion:

1) The width of polynomial interval extensions and Newton's operators descending from the application of m.v.t. at nth-order, $0 \le n \le N$, to the polynomial p is not always monotonically decreasing with respect to the order n;
2) polynomial interval extension and Newton's operator descending from the application of m.v.t. at maximum order ($n = N$) to the polynomial p always guarantees the smallest width with respect to those ones of all lower order m.v.t. forms ($n < N$);
3) the maximum order m.v.t. form of interval Newton's operator is always better than Newton's operator using the centered form for the interval extension of the derivative p'.

Before closing the discussion it is worth stressing some computational aspects connected to the use of lower order m.v.t. Newton's operators. We recall that we are interested in Interval Analysis for applications where large numbers of root searches are required, as in switching circuit time-domain analysis, under the worst case conditions i)–iv) listed in the Introduction. To this regard the main demand is for a roots-finding technique running as fast as possible, taking into account the great variety of situations that can occur. In general, the lower the number of arithmetic operations, the faster the algorithm. We have seen above that higher order reduced m.v.t. forms for p' in Newton's operator $N_n(I)$ certainly involve less interval segmentations and ensures sharper contractions, but require the calculation of higher derivatives of $p(t)$. Does this involve additional computations? As far as a centered form is adopted for the residual interval extension $P^{(n)}(I)$ in (5) the answer is no, because even in this case all derivatives of p must be computed indeed. If a natural interval extension is adopted for $P^{(n)}(I)$, instead, one could expect that there could

TABLE I
ROOT SEARCH BY MEANS OF KRAWCZYK'S OPERATOR, SECOND- AND EIGHTH-ORDER m.v.t. FORM OF NEWTON'S OPERATOR (EXAMPLE 1)

iteration n	I_n	$w(I_n)$	$w(I_n)/m(I_n)$
	Krawczyk's operator (speed up factor: 1)		
1	[0.00000000000000 0.43237002816005]	0.43237002816005	2.00000000000000
2	[0.04305082165896 0.38232834667416]	0.33927752501520	1.59517696338860
3	[0.11991569404060 0.30543868341355]	0.18552298937295	0.87232199411396
4	[0.19020324981479 0.23515112732385]	0.04494787750906	0.21134319957598
5	[0.21157906899495 0.21377530814369]	0.00219623914875	0.01032663241187
6	[0.21267470540409 0.21267967173455]	0.00000496633046	0.00002335149571
7	[0.21267718855666 0.21267718858198]	0.00000000002532	0.00000000011908
8	[0.21267718856932 0.21267718856932]	0	0
	Newton's operator: 2nd order m.v.t. form (speed up factor: 1.5)		
1	[0.07955033185890 0.22977275973426]	0.15022242787536	0.97129785624246
2	[0.20951277690184 0.22785204075746]	0.01833926385562	0.08386254730671
3	[0.21265597128011 0.21276900875401]	0.00011303747390	0.00053140966893
4	[0.21267718779991 0.21267719189710]	0.00000000409719	0.00000001926483
5	[0.21267718856932 0.21267718856932]	0	0
	Newton's operator: 8th order m.v.t. form (speed up factor: 2.2)		
1	[0.19519984495482 0.22301453975115]	0.02781469479633	0.13301644234875
2	[0.21263764305055 0.21274502474053]	0.00010738168998	0.00050487101666
3	[0.21267718799493 0.21267718955449]	0.00000000155957	0.00000000733303
4	[0.21267718856932 0.21267718856932]	0	0

be a convenience in terms of reduced derivative computations. But the coefficients of $p^{(n)}$ must be calculated in any case even to evaluate the natural interval form of $P^{(n)}(I)$, which certainly yields wider interval Newton's operator and then involves more segmentations and causes slower contraction. In addition, the width of $N_n(I)$ does not monotonically decrease with respect to the order n of the m.v.t. form. It could be interesting to establish exactly which is the minimum intermediate order m.v.t. form resulting more convenient than first-order m.v.t. form by a computational point of view. This is very difficult since all depends on the location and size of the interval I along the real axis and on the complex influence of several factors, like the intervals width, the amount of segmentations and the amount of real and interval arithmetic operations. However, with regard to the intervals width, maximum order m.v.t. interval extensions of polynomials and maximum order m.v.t. interval Newton's operator give always the best results if adopted, respectively, for $P(I)$ in the test of root existence and for interval contraction mapping (1)–(2), without involving more computations than first-order form. Consequently, computational optimality, i.e. smaller computing times, is also expected by maximum order m.v.t. forms. The results presented in the next section confirm such prediction.

Further reduction of interval width can be gained in several ways. For example, the powers of R in (5) can be nested to obtain the *nested* forms

$$\begin{aligned}\overline{P}'_{N-1}(I) &= c_1 + R(c_2 + \cdots + R(c_{N-1} + c_N R)\cdots);\\ P_N(I) &= c_0 + R\overline{P}'_{N-1}(I),\\ c_k &= \frac{1}{k!}p^{(k)}(\tau), \qquad k = 0, \cdots, N \end{aligned} \tag{10}$$

where the interval R appears always raised to the first power; this usually yields a narrower interval extension [1]. In the Appendix it is shown how the coefficients of centered forms can be easily calculated by means of Tartaglia's triangle. An alternative way of getting narrower interval extensions (5) consists in performing exact calculation of the powers R^k, which are made easier by the simmetry of the interval R. The applications discussed in the next section have been carried out by adopting the nested-forms (10) throughout the block diagram of Fig. 1.

IV. APPLICATIONS

The algorithm *first_root_search* has been developed and applied by the author in the framework of switched nonlinear networks analysis [18]; the examples proposed in the following are drawn from that application, which can be considered as a worst case since nonmonotonic and fast oscillating waveforms are usually present. A rounded machine interval arithmetic has been implemented under MATLAB® environment which computes separately intervals endpoints [3].

Example 1: Let us first consider a polynomial of eighth degree, whose coefficients $[p_0, p_1, \cdots, p_8]$ are

$$[1.82507, -4.99980, -16.90853, -7.32341, 35.81879, 7.94845, -37.15860, 10.20117, 4.11101]$$

for which the first root within the interval $I = [0, 0.5]$ is sought. This polynomial approximates the current of the diode in the ON state in a Cuk converter working in discontinuous quasi-resonant mode [18]; when this current drops to zero the diode spontaneously turns OFF. The results shown in Table IV have been obtained by means of the contraction mappings using, respectively, Krawczyk's operator ([10], [11]) and second- and eighth-order m.v.t. form of Newton's operator (5). Clearly eighth-order m.v.t. form of Newton's operator offers best convergence. Speed up factors have been extimated by running 100 consecutive root searches. To this regard it must be noted that the better performances of eighth-order

TABLE II
INTERVAL EXTENSION $\overline{P}'_{n-1}(I)$ AS A FUNCTION OF THE ORDER n OF m.v.t. FORM NEWTON'S OPERATOR (EXAMPLE 2)

n	$\overline{P}'_{n-1}([0.3,0.5])\times10^{-5}$		$\overline{P}'_{n-1}([0,1])\times10^{-5}$		$w(\overline{P}'_{n-1}([0,1]))\times10^{-5}$
1	[-0.02225380457556	0.00824067728537]	[-0.74396902274725	0.71781837514818]	1.46178739789544
2	[-0.01812369396265	0.00411056667246]	[-1.66337658490038	1.63722593730131]	3.30060252220170
3	[-0.01536534878401	0.00135222149382]	[-2.27656509985747	2.25041445225840]	4.52697955211588
4	[-0.01402840053520	0.00001527324502]	[-2.11729929407766	2.09114864647859]	4.20844794055626
5	[-0.01372758653003	-0.00028554076016]	[-1.31718611727510	1.29103546967603]	2.60822158695113
6	[-0.01366215328463	-0.00035097400555]	[-0.61009843110183	0.58394778350276]	1.19404621460458
7	[-0.01365512623547	-0.00035800105472]	[-0.23368602840502	0.20753538080595]	0.44122140921097
8	[-0.01365419308381	-0.00035893420638]	[-0.16078355510459	0.13463290750552]	0.29541646261011

m.v.t. form of Newton's operator with respect to second order one descend exclusively from the smaller width of $\overline{P}'_7(I)$, since higher order m.v.t. forms do not involve additional computations whenever centered form is adopted for residual interval extension $P^{(n)}(I)$ in (5). Krawczyk's operator, instead, requires always more computations than Newton's one, whatever form is adopted for $P'(I)$ in the expression of $K(I)$. In Table I it is shown that eighth-order m.v.t. operator guarantees the excellent accuracy of $5 \cdot 10^{-4}$ within two iterations only. The actual accuracy of t^*, namely the relative error $\Delta_e = (t^* - t_e)/t_e$ with respect to the exact root t_e, can turn out even some orders of magnitude better. In fact, if I_n is sufficiently small then $p(t)$ is quasi-linear within I_n and $P(I_n) \cong \overline{p}(I_n)$, so that $\Delta_e \ll \Delta_n$.

Example 2: The second example refers to a more critical polynomial, whose coefficients are

$$10^5 \times [-0.00208, 0.00050, 0.06855, -0.02769, -0.38485, -0.35249, 2.90658, -3.53902, 1.33307].$$

This polynomial approximates the diode voltage in the OFF state for the Cuk converter mentioned before [18]; the diode turns ON when this voltage falls to zero. In this case Newton's operaton is undefined, since $0 \in \overline{P}'_{N-1}(I)$ for $I = [0, 0.5]$, while Krawczyk's operator gives a starting interval $K(I) = [-1.905028006521238, 2.295185527505336]$ which includes I. Therefore, both operators require interval segmentation. In the interval $I = [0.3, 0.5]$ Newton's operator is defined if the m.v.t. form is fifth-order at least, as shown in Table II. Therein it is shown also that, while for $I = [0.3, 0.5]$ the width $w(\overline{P}'_{n-1}(I))$ is monotonic with respect to the order n of the m.v.t. form, for $I = [0, 1]$ it is not so: only from sixth-order upwards the width is smaller than that one of first-order form. This affects directly computing times, which however greatly depend also on the location of the interval along the real axis. For example, in Table III the interval width and the times required to find the first root within $[-1, 0]$ and [0,1] by means of nth-order m.v.t. Newton's operators are compared, for $n = 1, \cdots, 8$. All values are scaled to get unity for $n = 8$. Note that t_n is monotonic for $I = [0, 1]$, but not for $I = [-1, 0]$, in spite of larger widths. In Table IV the results of the root search performed by means of Krawczyk's and eighth-order m.v.t. Newton's operators are compared. The algorithm *first_root_search* with eighth-order m.v.t. Newton's operator requires one segmentation only to find the root within the interval $I = [0, 0.5]$; moreover, no computations are performed on the left half interval $I_L = [0, 0.25]$ since $0 \notin P_N(I_L)$. The speed up factor of m.v.t. forms Newton's operator with respect to Krawczyk's one ranges from 1.35 (first order) to 2.5 (eighth order).

TABLE III
INTERVAL WIDTH AND NORMALIZED COMPUTING TIMES OF NEWTON'S METHOD FROM FIRST- TO EIGHTH- m.v.t. ORDER n (EXAMPLE 2)

	$I=[-1,0]$		$I=[0,1]$	
order n	$w(N_n)$	t_n	$w(N_n)$	t_n
1	3.61	1.03	4.87	1.55
2	5.40	1.17	11.00	1.35
3	4.66	1.21	15.10	1.24
4	2.76	1.17	14.03	1.21
5	1.50	1.15	8.70	1.17
6	1.08	1.14	3.97	1.14
7	1.01	1.10	1.47	1.11
8	1.00	1.00	1.00	1.00

V. CONCLUSION

A robust and fastly convergent interval analysis-based method for root finding of nonmonotonic polynomials has been proposed in this paper. It has been developed for numerical time-domain analysis of switched nonlinear networks, which involves numerous calculation of the zeros of strongly nonlinear nonmonotonic waveforms. A special form of interval Newton's operator for interval contraction mapping has been adopted which guarantees high accuracy within few iterations. Its performances are sensibly better than those ones of other interval operators without involving additional computations. It has been demonstrated that the width of polynomial interval extensions and interval Newton's operators obtained using mean-value-theorem is not monotonic with respect to the forms order, but maximum order forms give always the smallest interval width. This directly affects computational efficiency. A simple method for the determination of the

TABLE IV
ROOT SEARCH BY MEANS OF KRAWCZYK'S AND m.v.t. NEWTON'S OPERATORS (EXAMPLE 2)

iteration n	I_n	$w(I_n)$	$w(I_n)/m(I_n)$
	Krawczyk's operator (speed up factor: 1)		
1	[0.30186385880138 0.50000000000000]	0.19813614119862	0.49418897989691
2	[0.30846447005464 0.50000000000000]	0.19153552994536	0.47382547295471
3	[0.32998503458193 0.50000000000000]	0.17001496541807	0.40968200228744
4	[0.38395337493109 0.50000000000000]	0.11604662506891	0.26256277391997
5	[0.45331747298518 0.50000000000000]	0.04668252701482	0.09793700071108
6	[0.47890580751849 0.48307675179930]	0.00417094428081	0.00867155904317
7	[0.48095522544547 0.48097714344648]	0.00002191800101	0.00004557077342
8	[0.48096618338739 0.48096618397227]	0.00000000058487	0.00000000121604
9	[0.48096618367983 0.48096618367983]	0	0
	Newton's operator: 8th-order m.v.t. form (speed up factor: 2.5)		
1	[0.46130915014280 0.50000000000000]	0.03869084985720	0.08049616473838
2	[0.48095748835321 0.48097564562623]	0.00001815727302	0.00003775163238
3	[0.48096618367578 0.48096618368425]	0.00000000000847	0.00000000001762
4	[0.48096618367983 0.48096618367983]	0	0

centered form of a polynomial by means of Tartaglia's triangle has been also indicated.

The theoretical properties discussed in Section III and the results presented in Section IV outline some important facts concerning the problem of roots finding by means of interval Newton's operator. Whenever using Interval Analysis two main elements must be accounted for: the width of intervals and the amount of arithmetic computations. It is a diffused belief, supported by theory and experiences indeed, that interval operators ensuring smaller interval width must be somehow more complicated, and then require additional computations, with respect to basic ones. This is often judged as a sufficient reason to renounce using enhanced versions of interval operators for contraction mapping. In despite of that, we have seen that maximum order m.v.t. form of Newton's operator joins both the objectives of small width and low amount of computations.

Finally, it is worth stressing one last point. Interval arithmetics is obviously more honerous than real arithmetics, since each interval computation requires a number of real computations. The main drawback of real type iterative methods for roots finding lies in the demand for monotonicity or for a suitable guess solution, whereas they allow faster computations in safe conditions. It is the author's opinion that further improvements can be attained in roots finding problems by cleverly matching preliminar interval analysis-based existence, inclusion and monotonicity tests with real analysis-based computations.

APPENDIX A

Let consider the polynomial

$$p(t) = p_0 + p_1 t + p_1 t + p_2 t^2 + \cdots + p_N t^N,$$

of which the coefficients of the centered form

$$p_c(t) = p(t) = c_0 + c_1(t-\tau) + c_2(t-\tau)^2 + \cdots + c_N(t-\tau)^N$$

are sought.

After the change of variable $t' = t - \tau$ we obtain

$$\pi(t') = p_0 + p_1(t'-\tau) + p_2(t'-\tau)^2 + \cdots + p_N(t'-\tau)^N$$

and

$$\pi_c(t') = \pi(t') = c_0 + c_1 t' + c_2 t'^2 + \cdots + c_N t'^N.$$

The coefficients of each Newton's binomial $(t'-\tau)^k$ in $\pi(t')$, for $k = 0, \cdots, N$, can be easily calculated by means of Tartaglia's triangle. The coefficient c_k in $\pi_c(t')$ equals the sum of the coefficients of t'^k in $\pi(t')$ coming from Newton's binomials. The coefficient of t'^s in $(t'-\tau)^k$, with $s \leq k$, is obtained as the product of $p_k(-\tau)^{k-s}$ by the sth element, from right to left, of the $(k+1)$th row, from the top to the bottom, of Tartaglia's triangle.

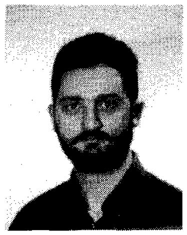
ACKNOWLEDGMENT

The author wishes to thank the anonimous reviewers for their deep work and for the discussions which have contributed to improve the paper.

REFERENCES

[1] R. E. Moore, *Interval Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1966.
[2] K.Nickel, Ed., "Interval mathematics," in *Proc. Int. Symp. on Interval Math.*, Karlsruhe (D), May 1975.
[3] R. E. Moore, "Methods and applications of interval analysis," in *SIAM Studies in Applied Mathematics.* Philadelphia: SIAM, 1979.
[4] G. Alefeld and J. Herzberger, *Introduction to Interval Computations.* New York: Academic, 1983.
[5] K. Nickel, Ed., "Interval mathematics," in *Proc. Int. Symp. on Interval Math.*, Freiburg i. Br. (D), Sept. 1985. Berlin: Springer-Verlag, 1986.
[6] R. E. Moore, Ed., *Reliability in Computing.* San Diego, CA: Academic, 1988.
[7] A. Neumaier, *Interval Methods for Systems of Equations.* Cambrige: Cambridge University Press, 1990.
[8] Ch. Ullrich, Ed., "Computer arithmetic and self-validating numerical methods," in *Proc. SCAN-89.* (Invited Papers). San Diego, CA: Academic, 1990. (sub. papers) J. C. Baltzer AG, Basel.
[9] E. Kaucher, S. M. Markov, and G. Mayer, Eds., "Computer arithmetic, scientific computation and mathematical modelling," in *Proc. SCAN-90. IMACS Annals on Computing and Applied Math.*, vol. 12, 1992, J. C. Baltzer AG, Basel, 1991.
[10] R. E. Moore, "A test for existence of solutions to nonlinear systems," *SIAM J. Numer. Anal.*, vol. 14, pp. 611–615, 1977.
[11] ——, "A computational test for convergence of iterative methods for nonlinear systems," *SIAM J. Numer. Anal.*, vol. 15, pp. 1194–1196, 1978.
[12] E. Hansen and S. Sengupta, "Bounding solutions of systems of equations using interval analysis," *Interval Library*, Freiburg, referenced in [7].

[13] R. E. Moore and L. Qi, "A successive interval test for nonlinear systems," *SIAM J. Numer. Anal.*, vol. 19, no. 4, pp. 845–850, Aug. 1982.
[14] F. A. Oliveira, "Bounding solutions of nonlinear equations with interval analysis," in *Proc. 13th World Congr. Comp. Appl. Math.*, Dublin, Ireland, July 1991, vol.1, pp. 246–247.
[15] H. Ratschek, "Centered forms," *SIAM J. Numer. Anal.*, vol. 17, no. 5, pp. 656–662, 1980.
[16] H. Ratschek and J.Rokne, "Optimality of centered form for polynomials," *J. Approx. Theory,* vol. 2, pp.151–159, 1981.
[17] G. Alefeld, "Uber das Divergenzverhalten des Intervall-Newton-Verfahrens," *Computing*, vol. 46, pp. 289–294, 1991.
[18] N. Femia and V. Tucci, "Time domain analysis of switched nonlinear networks with frozen variables," submitted to *IEEE Trans. Circuits Syst. —I.*



Nicola Femia was born in Salerno, Italy, in 1963. He received the doctor degree in engineering of industrial technologies (section electronics) with honors from the University of Salerno, in 1988.

Since 1990 he has been a researcher of Electrical Engineering with the University of Salerno, Faculty of Engineering, where he is a temporary teacher of electrotechnics. His main scientific interests are in switching networks and power electronics. The current field of activity concerns modeling, analysis and numerical simulation of power electronic circuits with internally controlled switching instants. He is the coauthor of technical papers on power electronics, HV materials engineering and electrical breakdown phenomena.