Improved Interpolation and Approximation through Order Manipulation

Vahid Rahmati*
Ayandeh Samin Electronics
vahidrahmati9170@gmail.com

Abstract

A novel transform calling smoothing, which can improve interpolation and reduce approximation error, is introduced in this paper. This method can be applied to various formulas, including interpolation and approximation methods, which are denoted in the process of order manipulation. Subsequently, the paper shows how to achieve higher degree polynomial approximations through fewer interpolation points, which is impossible with ordinary methods of interpolation. In fact, this leads to an alternative solution to oscillatory behavior and Runge's phenomenon occurring in polynomial interpolations or methods of least squares approximation when the number of points is increased significantly to achieve higher degree polynomials with the aim of error reduction. Several ideas—in the form of theorems and their proofs—are therefore studied on the basis of smoothing process of the interpolation. Finally, a comprehensive comparison, with the intention of showing the advantage of the new trasnform over other methods in the form of MSE v. number of samples, is provided.

Keywords— Approximation, interpolation methods, extrapolation, order manipulation

I. Introduction

A method for smoothing interpolation points to significantly reduce error is introduced in this paper. With this approach, the interpolation formulas obtained can be polynomial or exponential with regard to the smoother function (SF), which is varied by the given points. It's as well shown that the interpolation error is directly proportional to high values and order of the points of the available function, and can be reduced by manipulating the order of the interpolation equation from the given points; then, it's proved that error diminution by assumption of equality of the order of the original function and interpolation form leads to a new class of interpolation formulas in which the parameters differ dynamically by observing the order of function from at-hand points. New formulas are employed to develop an approximation to MISO as well as MIMO systems. At long last, this paper collates new approach of interpolation methods with classical formulas, and shows that some improvements are achieved at the expense of complexity.

The main application of interpolation nowadays is in the field of multi-rate signal processing for purposes of sample rate conversion and reconstruction of continuous signals by stored, digitized values used for a specific signal processing application. For instance, symbol synchronization in receivers, speech coding with synthesis, and computer simulation of continuous time systems, apart from certain uses, are well-known applications of interpolation in multi-rate signal processing. Readers can be familiar to basic concepts of multi-rate signal processing by checking [1].

For more on applications of approximation and interpolation in classic signal processing, readers referred to [2]; Chapters 4. Sampling of Continuous-Time signals, and part 7.7 Optimum Approximation of FIR Filters. In fact, Filter design techniques involve interpolation both in classic and modern forms, and as an application to interpolation for array filters, readers can check [3].

In addition to the abovementioned applications, interpolation is intrinsically used in numerical analysis and approximation theory, such as the Newton-Cotes formula used in numerical integration. For other elementary theorems of numerical analysis, readers may consider [8].

Sampling theorem, discovered by Shannon and Nyquist as one of the most useful theorems in signal processing, has a new rival called Compressed Sensing/Sampling, abbreviated as CS, presented by Donoho. In fact, it deals with asymmetric sampling which leads to better compression. For more on history and developments of sampling theorems, please check [4]. Nowak and others have a useful paper on basics of CS [5]. Structured CS with dictionary based method is the latest development in this field. Supper resolution, as an important application of CS, with recourse to a mathematical

*Vahid Rahmati is with research department, Ayandeh Samin Electronics, Tehran, Iran.

framework such as the one presented in [6] by E. Candes and C. Fernandez-Granda, can be considered as an active subfield for future researches. Spectral and complex functions can also be integrated into CS frameworks as mentioned in [7].

Apart from the different kinds of engineering usages mentioned, one may need polynomial and exponential approximations; hence, accuracy is necessary. Accuracy, sometimes, results in complexity of algorithms. For example, Neville-Aitken algorithm suffers from this and in [9] a faster approach to polynomial interpolation is presented. In this paper, we don't try to optimize algorithms presented, but study a new theory that yields a new class of interpolation formulas.

The polynomial form of interpolation has some disadvantages; for instance, edge oscillation usually referred to as the oscillating theorem, occurs when the degree of the polynomial increases. Another well-known problem is the increase in interpolation error for highly exponential functions in accordance with Rolle's Theorem used to derive the error function. This theorem states: "at least one zero exists in the derivative of a function which is between two zeros of it." Certain considerations are required for exponential functions to have lessened approximation error, because the nth derivative of an exponential function is itself unlike the power of polynomials diminishing by differentiation. Readers can reach similar idea by comparing two polynomial interpolation methods of Lagrange and Newton discussed in [10].

Another issue in numerical analysis is that there are few chances of choosing another set of points in some applications; it means that points are dictated by a constraint, thus, Chebyshev Points—which are used to have lower error—are not available, and error reduction fails in this case too; however, by using these points, only the minimization of the product part of the error function is achieved. Piecewise-polynomial splines in subintervals primarily invented to solve Runge's phenomenon, which occurs for long-intervals in ∞-*norm*, and not error reduction; thence, issues of less number of points and $inf_{P_n \in \Pi} ||f - P_n||_\infty$ error minimization still prevail.

Another topic of interest is the minimax polynomial, where the theorem may be renewed for non-polynomial approximation formulas too; so any certain form of exponential interpolation can satisfy minimax existence theorem necessarily in $\mathbb{R}^{n+1}$ [11] for $\sum_{i=0}^{n} \eta_i x^i$ contained within the space of polynomials of $degree \leq n$ with point $\eta = (\eta_0, \ldots, \eta_n)$. The best approximation of non-linear type for L^∞ norm also exists by existence theorem extension, which is, however, not studied in this manuscript.

The Vallèe-Poussin theorem is useful for error bounding, while its infinite norm is just another error minimization problem. This is also true for polynomial-based interpolation formulas, yet it's used for characterizing minimax methods. The literature contains many interpolation methods together with rich theories; for example, Lagrange and Newton forms of polynomial interpolation, nonsingular Vandermonde matrix for finding polynomial coefficients, linear approximation, piecewise Splines (as a solution to remove interfering extremas caused by high-degree polynomials) and Hermite interpolating polynomials, which are for both the function and its derivatives. But often, since the main function points are available, it simplifies to polynomial interpolation. In fact, this paper suggests order manipulation as a step prior to interpolation rather than providing just another formula [11].

The important concern of non-oscillatory Taylor series is convergence; moreover, the fact is that Taylor polynomials are not applicable to functions that are not differentiable infinite times. This leads to many continuous functions being disregarded. Choosing a set of interpolation points that converge to x_0 gives the Taylor series at that point, if the function is also differentiable.

The Neville-Aitken method is a soft, iterative algorithm used for evaluating interpolating polynomial of degree n using $n+1$ abscissas with the corresponding values of functions. In this case, the problem is how to optimize this algorithm for many values of x, which means running an algorithm with lower complexity if one needs many values of interpolating polynomial.

Choosing the best approximation concerns many interesting topics such as minimax approximation, Legendre and Chebyshev Polynomials, Lebesgue function, and many other subjects that are ignored here in this paper.

Studying order manipulation and error reduction in the matter of different types of functions is the main objective of this paper, while the comparison is carried out for both introduced and other interpolation formulas. The paper starts by presenting the order manipulation theorem, defining average error and order detection for signals, and then investigates smoothing transforms (STs) and improvements for two important smoothers. Several theorems with their proofs are added with the hope they would be useful. System approximation is also an important part of this paper. Ultimately, this paper collates new methods with classic types for several main function types.

II. Average error and Order definitions

In this section, Mean Square Error and formal order definitions are presented. The next part discusses new error factor by using an integral and it employs a special interpretation of order definition to build theories based on the aforementioned ideas.

MSE for N-point discrete function is defined as

$$MSE = \frac{1}{N}\sum_{m=0}^{N-1}\left(x\left(n\right) - \hat{x}\left(n\right)\right)^2, \tag{1}$$

where $x\left(n\right)$ is a discrete signal with its approximation $\hat{x}(n)$. In Part 3, this definition will be changed to get an integral error factor. The next thing that will be redefined later is the Big$-O$ notation. The usage of Big$-O$ notation in this paper is a little different from the usual applications, as in

$$f\left(x\right) = \sum_{n=0}^{\infty} a_n x^n = a_0 + a_1 x + a_2 x^2 + O\left(x^3\right), \tag{2}$$

which denotes canceled terms by indicating the next term's degree. The next usual application is showing the function's behavior in infinity, e.g. for a polynomial, and so we have

$$\sum_{n=0}^{m} a_n x^n = O\left(x^m\right), \quad as\ x \to \infty. \tag{3}$$

The specific definition of Big$-O$ notation is studied in *definition 1* of the next part. In this paper, the continuous variable x is replaced by t to resemble time domain variable compatible with signal processing standards.

III. Order manipulation and Average Error

III.1 Theory of Order Manipulation

In this part, the theory of order manipulation is studied as a separate, new idea, so as to apprehend the reasons producing error and showing order convergence as the preference intended for function and its approximation. In fact, this means $lim_{t\to\infty}\, ord(x\left(t\right))/ord\left(\hat{x}\left(t\right)\right)$ must converge to a constant. Indeed, accurate approximation based on order detection for the sample's vector (interpolation points), with a general form of interpolation, cannot diminish error for all forms of functions. In this case, we first introduce the familiar definition of Big$-O$ notation.

Definition 1: Assume that $lim_{x\to\alpha}\, f\left(x\right)/g\left(x\right) = 0$, so it can be written by means of subset's notation as $O(f\left(x\right)) \in O(g\left(x\right))$ and read as order of $f(x)$ is subset to order of $g(x)$.

With this definition mentioned above, it's easy to write

$$O(\sum_{i=0}^{N} a_i x^i) \in O\left(a^x\right) \in O\left(\Gamma\left(x\right)\right), \tag{4}$$

as limit converges to zero for consequent functions. Now, by assuming x as the main function, $\hat{x}$ as its approximation, and both f and y as arbitrary functions, the next idea can be formed. Now check the following

$$\begin{gathered}
\forall\, x, \hat{x},\ y \in O\left(f\right),\ O\left(x\right) \in O\left(\hat{x}\right) \in O\left(y\right) \\
\Leftrightarrow \\
\left\{O\left(x\right) \in O\left(\hat{x}\right)\right\}\ \wedge \left\{O\left(\hat{x}\right) \in O\left(y\right)\right\} \wedge \left\{O\left(x\right) \in O\left(y\right)\right\} \\
\Leftrightarrow \\
\left\{\bigwedge_{i=0}^{n}\left\{O\left(x_i\right) \in O\left(\hat{x}_i\right)\right\}\right\} \\
\wedge \\
\left\{\bigwedge_{i=0}^{n}\left\{O\left(\hat{x}_i\right) \in O\left(y\right)\right\}\right\} (4) \\
\wedge \\
\left\{\bigwedge_{i=0}^{n}\left\{O\left(x_i\right) \in O\left(y\right)\right\}\right\} \in O\left(f\right),
\end{gathered}$$

thus, approximation of x, which is $\overset{\wedge}{\mathrm{x}}$, can be accurate enough if we assume the order of main function equaling or subsetting to the order of interpolation formula denoted by f, and also subsetting to the order of this function itself. By this logic, the reverse case is not possible. To clarify this, consider $p \nRightarrow h\ \ \wedge h \Rightarrow p\ \ \equiv p \nLeftrightarrow h$, where p is the proposition for the existence of an accurate polynomial approximation while h is the proposition for the existence of a highly exponential function. In this case, and by previous discussion on order sub-setting, $h \Rightarrow p$ is possible by notation $O\left(x\right) \in O\left(y\right)$, but the reverse case is not; as a result, the deduction is: non-polynomial and polynomial interpolations do not yield equal errors in the sense of the infinity norm. The last three lines of (4) affirm that a class of functions called x_i, with their approximations $\hat{x}_i$, can be subset to f which is not inevitably a polynomial. Of course, it's clear that certain conditions are necessary for this to be valid; for instance, a class of sinusoidal functions of the form $x_i\left(t\right) = sin(\omega_i t)$ can have approximations like $\hat{x}_i\left(t\right) = A_i sin(B_i t) \in O\left(f\left(t\right)\right)$, and this is sufficient condition to parametrically reconstruct a wide-range of functions from $f(t)$, by assumption in the first line of (4).

III.2 Average Error

A modified interpretation of MSE is used in this paper, which is denoted by E_A. To define this factor, consider $d\left(n\right) = x\left(n\right) - \hat{x}(n)$ with N bounded values; then MSE is

$$MSE := \frac{1}{N}\sum_{m=0}^{N-1} d^2\left(m\right). \tag{5}$$

Now, for every integer and non-integer point of difference function d, and by $dt = lim_{N\to\infty}\frac{t_2'-t_1'}{N}$, the outlined sum becomes an integral

$$\underset{N\to\infty}{lim}\frac{1}{N}\sum_{m=0}^{N-1} d^2(t_1' + mdt)\ = \frac{1}{t_2' - t_1'}\int_{t_1'}^{t_2'} d^2\left(t\right)dt := E_A. \tag{6}$$

If x is also analytical for every $t_1' + mdt$ value, the right side integral converges absolutely for the range $t \in (t_1', t_2')$, if and only if $\hat{x}$ has no singularities in this interval. Analytical assumption for x is rational, but $\hat{x}$ may be unbounded, as is completely related to interpolation form; therefore, convergence of (6) relies to a great extent on $\hat{x}$.

IV. Smoothing Transforms and Applications

Several theorems are studied under the title of smoothing[1] transforms. In this case, and by the first theorem, we show the way a polynomial interpolator transformed into another form—for instance, exponential—while it still interpolates given points. In the proof of Theorem 1, an attempt to show the possibility of reducing error by manipulating the order of the interpolator is presented. The next theorem also uses the first one to transform Newton and Lagrange forms into exponential types by selecting logarithmic smoothing function indicated by g. The third theorem extends the concept of uniqueness for exponential interpolation forms as proved in the literature for polynomial interpolations. Theorem 4 studies an application of Theorem 2 and (14) to estimate the n^{th} derivative of a function by at least two points ($N = 2$) out of regard for certain conditions. Finally, Theorem 5 shows modified existence theorem extension to improve certain algorithms that are derived from algebraic equations.

Theorem 1. *If $P_n(t) > 0$ is a polynomial interpolation in closed interval $\left[t_1', t_2'\right]$, with N bounded points of $x(t) > 0$ as follows*

$$\mathrm{P_n}(\mathrm{t}) = f(\mathrm{x}(\mathrm{t_0}), x(\mathrm{t_1}), \ldots, x(\mathrm{t_{N-1}}), t) := \mathrm{f}(\mathrm{x}(\mathrm{t_i}), t), \quad (7)$$

by the assumption of order manipulation that was discussed earlier, and for smoothing $x(t_i)$ points, this polynomial is transformed to P_n' as follows in (8) by a function g, which is selected in accordance with the order of interpolation points.

$$\mathrm{P}_\mathrm{n}'(\mathrm{t}) := \mathrm{g}^{-1}(\mathrm{f}(\mathrm{g}(\mathrm{x}(\mathrm{t_i})), t)). \quad (8)$$

Proof. To show that the defined equation moderates error to a considerable degree by certain g, recall the error function e for a polynomial interpolation P_n as shown in

$$e(t) = x(t) - P_n(t) = \frac{x^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (t - t_i). \quad (9)$$

According to this, the error is directly proportional to magnitudes of $\mathrm{x}^{(\mathrm{n+1})}(\xi)$, as stated in (9); now, if $g^{-1} = O(x)$ is chosen correctly, then it indicates property (10):

$$|O(x(t))| \approx \left|O\left(g^{-1}\left(f\left(g\left(x\left(t_i\right)\right), t\right)\right)\right)\right|, \quad t_{1'} \leq t \leq t_{2'} \quad (10)$$

Notification is required, because, by this process, $O\left(P_n'\right)$ is not necessarily equal to $O(x)$, where limit is not in ∞, and this is the reason for having a special definition for order as explained before. But, here, the deduction is that

$$log_t x(t) \propto \alpha log_t P_n'(t), \quad (11)$$

for constant α which is independent of order meaning P_n' has almost an equal order compared to x for $t \in [t_1', t_2']$. Interpolation is done for smoothed magnitudes of x rather than main points right away, and to recover preserved amplitudes, g^{-1} is applied to it. By presented order definition and detecting order of x, the best choice is $g = O(x^{-1})$.■

This theorem is employed by the next theorems to generate interpolation formulas as examples of the ST.

Theorem 2. *Let $x(t) > 0$ for $t \in [t_1', t_2']$ with $N \geq 2$ bounded points of equally spaced by $T = t_m - t_{m-1}$, then $I_N(t)$, $I_L(t)$, and $I_R(t)$ in (12–14) interpolate given points almost equalling to original function $x(t)$. First,*

$$I_N(t) := \prod_{i=0}^{N-1} a_i^{\varphi(i,t)}, \quad \varphi(m,t) = \frac{1}{T^m m!} \prod_{i=0}^{m-1} (t - t_i), \quad (12)$$

Second,

$$I_L(t) := \prod_{i=0}^{N-1} x^{\psi(i,t)}(t_i), \quad \psi(m,t) = \prod_{\substack{i=0 \\ i \neq m}}^{N-1} \frac{t - t_i}{t_m - t_i}, \quad (13)$$

And finally,

$$I_R(t) := \left(f\left(\sqrt[c]{x(t_i)}, t\right)\right)^c, \quad (14)$$

where a_i factors are calculated by Forward Division Algorithm (FDA). FDA is very similar to Newton's Forward Difference, so a modified algorithm for extraction of the a_i values is used as a modified Newton's Forward Difference (NFD) [11] followed in Fig. 1.

[1] There are certain cases in which the word "smoothing" used in mathematics and signal processing. In this paper, applying specific functions on data sets and then their inverses on the resulting trasnform/algorithm is the meaning of smoothing.

t	$x\,(\mathrm{t})$		
t_0	$x\,(t_0) = a_0$		
t_1	$x\,(t_1)$	$\frac{x(t_1)}{x(t_0)} = a_1$	
t_2	$x\,(t_2)$	$\frac{x(t_2)}{x(t_1)}$	$\frac{x(t_2)}{x(t_1)}\ \frac{x(t_0)}{x(t_1)} = a_2$
$\vdots$	$\vdots$	$\vdots$	$\ddots$
t_{N-1}	$x\,(t_{N-1})$	$\frac{x(t_{N-1})}{x(t_{N-2})}$	$\ldots$

Fig. 1. Forward Division Algorithm presented for calculating a_i factors of interpolation equation (12).

$f(x(t_i), t)$ is a polynomial interpolation similar to Equation (7), where c is smoothing parameter and for (14); c is calculated via the signal's order, as in (15)[2]

$$if\ (\{\max(x\,(t)) \ = x\,(t_{m'})\} \wedge \{t_{m'} > 0\ \}) \Rightarrow c = \frac{log\,(x\,(t_{m'}))}{log\,(t_{m'})}, \tag{15}$$

Proof. A formal proof based on mathematical induction for these interpolation formulas can be found by substituting $t = t_i$ for $0 \le i < N$ to check equality of $I_N(t_i)$, $I_L(t_i)$, and $I_R\,(t_i)$ to $x(t_i)$ in certain sampling points. Consequently, for the rest of the points, the signal is almost equal to $(12-14)$, but to use Theorem 1 for re-proving these identities indirectly, the base interpolation method is NFD in (12), Lagrange Interpolating Polynomial (LIP) in (13), and any polynomial approximation for (14) with parameter c, which is the polynomial's degree and is given by $n = c'(N-1)$, where $c' = \lceil c \rceil$. So, by applying theorem 1 on NFD, we get

$$I_N\,(t) \ = \exp\left(\frac{log\,(x\,(t_0))}{0!T^0} + \frac{\log\,(\mathrm{x}\,(\mathrm{t}_1)) \ - \log\,(x\,(t_0))}{1!T^1}\,(t-t_0) + \ldots\right)$$

$$= \exp\left(\log\left(x^{\frac{1}{0!T^0}}\,(t_0)\right) + \log\left(\left(\frac{x\,(t_1)}{x\,(t_0)}\right)^{\frac{(t-t_0)}{1!T^1}}\right) + \ldots\right)$$

$$= \exp\left(\log\left(x^{\frac{1}{0!T^0}}\,(t_0)\left(\frac{x\,(t_1)}{x\,(t_0)}\right)^{\frac{(t-t_0)}{1!T^1}}\ldots\right)\right)$$

$$= \prod_{i=0}^{N-1} a_i^{\varphi(i,t)}, \ \ \varphi\,(m,t) = \frac{1}{T^m m!}\prod_{i=0}^{m-1}(t-t_i)\,.$$

Furthermore, application of theorem 1 on LIP yields

$$I_L\,(t) = \exp(\log\,(x\,(t_0))\ \frac{(t-t_1)\cdots t - t_{N-1})}{(t_0-t_1)\cdots(t_0-t_{N-1})} +$$

$$\cdots + \log\,(x\,(t_{N-1}))\ \frac{(t-t_0)\cdots(t-t_{N-2})}{(t_{N-1}-t_0)\cdots(t_{N-1}-t_{N-2})})$$

$$= \exp(\log\left(x^{\frac{(t-t_1)\cdots(t-t_{N-1})}{(t_0-t_1)\cdots(t_0-t_{N-1})}}\,(t_0)\right) +$$

$$\ldots + \log\left(x^{\frac{(t-t_0)\cdots(t-t_{N-2})}{(t_{N-1}-t_0)\cdots(t_{N-1}-t_{N-2})}}\,(t_{N-1})\right))$$

$$= \prod_{i=0}^{N-1} x^{\psi(i,t)}\,(t_i)\,, \ \ \psi\,(m,t) = \prod_{\substack{i\,=\,0\\ i\,\neq\,m}}^{N-1}\frac{t-t_i}{t_m-t_i}$$

The same proof can be found for (14) too. Now consider

$$\lim_{t\to\infty}\frac{x^{\frac{1}{c}}\,(t)}{logx(t)} \ \to \infty, \tag{16}$$

which shows that a n^{th} root function smoother is not befitting a higher order type of signal, though it still keeps errors lessened to a better extent than pure polynomial-based interpolation as $O\left(t^{N-1}\right) \in O\left(t^{c(N-1)}\right) \in O\left(a^t\right)$. ■

This theorem is an application of Theorem 1 employing two main SFs in an algebraic form. Matrix form can be used with regard to this fact that g needs to be an one-to-one correspondence function and satisfy Property (10). We have to notice the assumption in Theorem 2 that states $x\,(t) > 0$, but if signals are both positive and negative, a constant b_0 can be added to make $x(t_i)$ positive and then subtracted after interpolation as shown in (17)

$$\mathrm{I}\,(\mathrm{t}) := \mathrm{g}^{-1}\left(\mathrm{f}\left(\sqrt[\mathrm{c}]{\mathrm{g}\,(\mathrm{x}\,(\mathrm{t_i}) + \mathrm{b}_0)}, t\right)\right) - \mathrm{b}_0, \tag{17}$$

by adding/subtracting b_0, this issue is solved in a simple manner; now we can study the next theorem which considers the concept of uniqueness.

Theorem 3. *There exists a unique exponential interpolation of the form (12) or (13) for a set of N bounded points of the function x.*[3]

Proof. Uniqueness of exponential interpolation is a direct consequence of the uniqueness of a polynomial interpolation, so if Newton and Lagrange interpolation forms are denoted by $N(t)$ and $L\,(t)$, then

$$\mathrm{N}\,(\mathrm{t}) = L\,(\mathrm{t}) \to smoothing \to \mathrm{I_N}\,(\mathrm{t}) = \mathrm{I_L}\,(\mathrm{t})\,. \tag{18}$$

In this case, smoothing is equivalent to applying a function like g on the points' magnitudes. ■

This theorem suggests that both of the forms of the polynomial interpolation can be chosen as a base equation for (8) resulting in an equality regardless of g as the SF.

One of the applications of (14) can be an estimation of the n^{th} derivative of x for $t = t_0$, which is studied in Theorem 4 in the wake of Theorem 2.

[2] Please note that by shifting t, it's possible to change the domain of validity. In this paper, a simple interpretation by max{} function is used.

[3] Uniqueness of a polynomial interpolation simply states that "only a unique polynomial of degree n for a set of $n+1$ points exists." Readers may question that "if it's so, why do we compare interpolating polynomials?" In fact, uniqueness theorem asserts existence of such polynomial, however, extraction of it by various algorithms produces different errors, so it's rational to compare methods.

Theorem 4. *Assume $x(t) = a^t$, then the n^{th} derivative is estimated by*

$$x^{(n)}(t) \cong \frac{d^n}{dt^n}\left(f\left(\sqrt[c']{x(t_i)}, t\right)^{c'}\right), \tag{19}$$

where $n \leq c'(N-1)$ applies.

Proof. Polynomial interpolation of a set of abscissas that tends to t_0 and their corresponding x values led to a Taylor series of differentiable x. In fact, it's easy to prove by induction that the n^{th} derivative is calculated on the strength of the preceding derivatives, and in (19), the n^{th}coefficient is calculated by virtue of all $x^{(i)}(t_0)$ values where $0 \leq i < n$. ■

A special case of (19), where $N = 2$ and $c = c'$ are chosen in (14) with recourse to binomial expansion of $\mathrm{I_R}$, is

$$I_R(t) = \sum_{k=0}^{c} \binom{c}{k} (x(t_0))^{\left(\frac{c-k}{c}\right)} \left(\frac{x^{\frac{1}{c}}(t_1) - x^{\frac{1}{c}}(t_1)}{t_1 - t_0}\right)^k t^k, \tag{20}$$

where t^k coefficients are almost equal to $x^{(k)}(t_0)/k!$, so $x^{(n)}(t_0)$ can be written as

$$x^{(n)}(t_0) \cong \frac{\Gamma(c+1)}{\Gamma(c-n+1)} (x(t_0))^{\left(\frac{c-n}{c}\right)} \left(\frac{x^{\frac{1}{c}}(t_1) - x^{\frac{1}{c}}(t_1)}{t_1 - t_0}\right)^n. \tag{21}$$

In contrast to the Existence Theorem and with an eye on the Existence Theorem Extension—which requires much work and is not analyzed in this document—a Modified Existence Theorem Extension is studied by Theorem 5 in combination with previous theorems and to make headway for the purpose of showing that an approximation of any kind (and not only polynomials) can have an existence theorem of its own. But to clarify distinction by dint of the existence of minimax polynomial, recall that if P_n is a minimax polynomial that certainly exists, and $f \in C^0[a,b]$, then $||f - P_n||_\infty$ is minimized for any $n \in \mathbb{N}$. Indeed, Equation (14) in Theorem 2 gives polynomial interpolation of any degree, but it does not verify that $\mathrm{I_R}$ is P_n; in support of this, we may say that (14) generates polynomials of any degree, but polynomials of high degree do not always reduce error.

Theorem 5. *Assume that $g(x(t))$ exists for $t \in [t_1', t_2']$ with $N \geq 2$ bounded values, then $I_g(t)$ in (22) usually interpolates given points with lowest possible error.*

$$I_g(t) = g^{-1}\left(f\left(\sqrt[c]{g(x(t_i) + a(t))}, t\right)\right) - a(t). \tag{22}$$

In (22)*, a is an arbitrary function chosen with regard to x , and g is a smoothing function chosen with respect to $x(t_i)$ and its magnitudes.* *Proof.* An ordinary proof based on mathematical induction for these interpolation formulas can be found by substituting $t = t_i$ for $0 \leq i < N$ to check equality. ■

The application of Theorem 5 is vital if we consider $x(\mathrm{t_i})$ as inputs to an algorithm, g , a as our modifications to it, and finally f as the algorithm itself that needs improving. Of course, choosing appropriate a is a must. In fact, in (17), $\mathrm{b_0}$ was a special case of a while choosing g also studied extensively in previous theorems. Indeed, (22) asserts that error produced in a chosen algorithm f can be reduced greatly if SF g and additive function a are selected according to the input data or seeds called $x(\mathrm{t_i})$.

Approximation formulas developed so far employ simple forms of SF g, instead, it's possible to derive a variety of other formulas based on different forms of SF. For example, we choose $g(t) = x^{(m)}(t)$ in order to have integral approximation formulas as

$$\hat{x}(t) = \underbrace{\int \dots \int}_{m} \mathrm{f}\left(x^{(m)}(t_i), t\right) dt^m, \tag{23}$$

where $f(\alpha_i, t)$ is any form of approximation. Another choice for SF is $g(t) = \log\left(x^{(m)}(t)\right)$ used in Lagrange interpolating polynomial and resulted in the following repeated integral

$$\hat{x}(t) = \underbrace{\int \dots \int}_{m} \prod_{i=0}^{N-1} \left(x^{(m)}(t_i)\right)^{\psi(i,t)} dt^m, \tag{24}$$

where $\psi(i,t)$ is defined in (13). If we apply n-fold integral formula, we get

$$\hat{x}(t) = \frac{1}{\Gamma(m+1)} \int (t-\alpha)^{m-1} \prod_{i=0}^{N-1} \left(x^{(m)}(t_i)\right)^{\psi(i,\alpha)} d\alpha. \tag{25}$$

IV.1 MISO/MIMO System Approximation

Current trends in system approximation are divided into several categories as: 1) Fuzzy systems 2) Neural networks 3) Adaptive fuzzy neural networks 4) SVD-based method 5) Krylov method 6) Moment matching 7) SVD-Krylov method, and some other methods. Each of these may suffer from certain problems that are outlined in summary by following.

An important problem restricting the applicability of ordinary fuzzy controllers is the rule-explosion problem; it means, the number of rules in the database increases exponentially with the number of input variables to the controller itself. A hierarchical fuzzy controller as a universal approximator is a prime solution to this problem [12]. We can suggest formulas developed in this part or any other system approximator based on theorem 6 as a universal approximator if and only if interpolation form is selected well. So, in this case, readers firstly construct a system approximation, and secondly prove that it as a universal approximator by an arbitrarily small error bound [12]. Many system approximations are provided in the literature such as recursive and non-recursive mathematical approximators, neural approximators, fuzzy approximators and more, but among them, fuzzy controllers as universal approximators have an important property of uniform convergent. This property shows that fuzzy based MIMO systems with defined approximation accuracy can always be acquired by separating the input space into finer regions [13]. It's done specifically for MIMO systems satisfying uniform approximation bounds. The fuzzy system approximators working well, however, by adding the adaptively of learning inherited from neural networks as a new feature to these systems, they can be more effective. In [14], authors showed that fuzzy systems with high dimensions can be implemented with fewer number of rules compared to the Takagi-Sugeno fuzzy systems. These new systems are called adaptive fuzzy neural networks.

Van Dooren and others in [15] showed that it's possible to derive the gradients of the H2-norm of the approximation error via tangential interpolation. Krylov method, mainly for solving linear systems of equations, is accordingly studied in details by Grimme in [16]. Other model reduction methods such as SVD and moment matching are considered by Antoulas and others in [17]. In [18], authors compared SVD-based, Krylov-based, and SVD-Krylov-based methods extensively for approximation of large-scale systems.

In this part, we would like to develop a simple approach towards MIMO system approximation. Indeed, several pure, mathematical applications of mentioned theorems studied so as to ensure the development of a framework to express a MISO system. A simple key idea, for a modification on MISO system approximation, will be applied on this framework to develop MIMO system approximations. In fact, a MISO system S_i, consisted of N inputs with M values each, can be shown as a multi-variable function

$$Y_i = S_i\left(X_{11}\ldots X_{1M},\ \ldots,\ X_{N1}\ldots X_{NM}\right),\ 1\leq i\leq M, \tag{26}$$

where X_{ij} and Y_i denote input and the only output's i^{th} values respectively. We should stress, however, that in general,

$$\begin{pmatrix} Y_1 \\ \vdots \\ Y_M \end{pmatrix} = \begin{pmatrix} X_{11} & \cdots & X_{1N} \\ \vdots & \ddots & \vdots \\ X_{M1} & \cdots & X_{MN} \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_N \end{pmatrix} \tag{27}$$

is an equivalent matrix formulation to the MISO systems. Recall that the square matrices of this form (, where $M = N$), are not easily invertible resulting in singular, near singular solutions and many researcher have tried to acquire a_i by the use of different algorithms. So for large values of $M = N$, we cannot rely on a_i coefficients extracted by badly conditioned matrix

$$\begin{pmatrix} a_1 \\ \vdots \\ a_M \end{pmatrix} = \begin{pmatrix} X_{11} & \cdots & X_{1N} \\ \vdots & \ddots & \vdots \\ X_{M1} & \cdots & X_{MN} \end{pmatrix}^{-1} \begin{pmatrix} Y_1 \\ \vdots \\ Y_N \end{pmatrix}. \tag{28}$$

This fact must not be neglected that an alternative solution to this system representation is vital. One of the aims of this part is to indirectly solve this issue by interpolation and employment of a supplementary mapping function that is called T, instead of finding a_i coefficients. At this stage, and specifically for MIMO system, (26) can be generalized as

$$Y_{ji} = S_{ji}\left(X_{11},\ldots,X_{1M}\ldots X_{N1},\ldots,X_{NM}\right),\ 1\leq i\leq M, 1\leq j\leq P. \tag{29}$$

where P is the number outputs of the specified system. Now, with next theorem, we enable many interpolation formulas to be used as system approximations, that is, trying to convert multi-variable functions to non-unique single variable functions. Bijectivity condition, for the supplementary function that maps a set of input points $(X_{1i},\ldots,X_{Ni})$ to $T(X)$ as

$$(X_{1i},\ldots,X_{Ni}) \to T(X), \tag{30}$$

is the main purpose of the following theorem.

Theorem 6. *The necessary and sufficient condition for the existence of the invertible mapping function T, to map a multi-variable function to a single-variable function, is bijection as shown in*

$$i\neq j\ \Leftrightarrow T\left(X_{1j},\ldots,X_{Nj}\right)\neq T\left(X_{1i},\ldots X_{Ni}\right). \tag{31}$$

Proof. Assume that T^{-1}, the inverse of the mapping function T, defined as

$$X_{ij}{:=}T^{-1}\left(T\left(X_{j1},\ldots,X_{jN}\right),i\right).$$

We can summarize this notation as

$$X_{ij} = T^{-1}\left(T\left(X_{ji}\right),i\right),\ 1\leq j\leq N.$$

Now, assume that some i, j, and k, where $i\neq j$, exist such that the equality

$$T\left(X_{ik}\right) = T\left(X_{jk}\right),$$

holds. Clearly, the system maps inputs to wrong outputs as

$$T^{-1}\left(T\left(X_{ik}\right),i\right) = T^{-1}\left(T\left(X_{jk}\right),j\right),$$

where bijectivity condition is violated, so, in this case, approximation of the system with T doesn't exist. This ends the proof of the theorem by contradiction. □

In order to use this theorem efficiently, choose (25) as main interpolation method with a bijective function T. Then, for a MISO system, we have

$$S\left(X_1,\ldots X_N\right) = \prod_{i=1}^{N} Y_i^{\psi(i,T(X))},$$

$$\psi(m, T(X)) = \prod_{\substack{i=1 \\ i \neq m}}^{N} \frac{T(X) - T(X_{1i})}{T(X_{1m}) - T(X_{1i})}. \tag{32}$$

Clearly, for MIMO system, the multiple outputs should be used as

$$S_j(X_1, \ldots X_N) = \prod_{i=1}^{N} Y_{ij}^{\psi(i,T(X))},$$

$$\psi(m, T(X)) = \prod_{\substack{i=1 \\ i \neq m}}^{N} \frac{T(X) - T(X_{1i})}{T(X_{1m}) - T(X_{1i})}. \tag{33}$$

V. Comparison of Interpolation Formulas

In this part, an examination of the interpolation methods is carried out to detect the pertinence of each formula based on signal types. In fact, NFD v.LIP was always a hot topic of research. For example, a more classical comparison can be found in [19] that was done by Werner in 1984. In [19], the errors are rounded in a specific way, so the readers may notice that for certain exponential functions these rounded errors can make much distinction in final comparisons. For specific reasons such as complexity, Aitken's and Neville's algorithms were not part of many interpolation comparison benchmarks. To resolve this privation, readers may use [20] that compares the Lagrange representation, the Barycentric formula, Aitken's algorithm, and Neville's algorithm. Krogh in [9] compared algorithms such as polynomial interpolation with derivatives in certain points, simple interpolation, and piecewise polynomials having a continuous first derivative.

The comparison in this part considers 8 interpolation methods/formulas. These formulas are I(t) which is (12) or (13), N(t) and L(t) as both forms of polynomial interpolations, N$RS(t)$ and L$RS(t)$ as both root smoothed forms shown in (14), C$Spl(t)$ and C$Her(t)$ as cubic Spline and Hermite piece-wise interpolation and finally L$in(t)$ is the linear type. Therefore, to clarify the following comparison, all parameters are fixed except N, interpolation methods, and also test signal for $t \in [1,\ 20]$.

V.1 Sinusoidal

Signal form: $x(t) = \sin\left(\frac{\pi t}{10}\right),\ \ 1 \leq t \leq 20.$

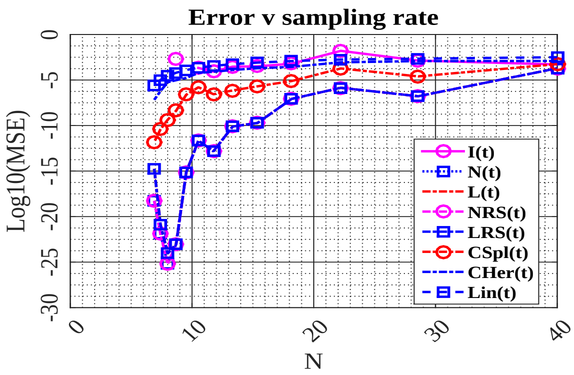

Fig. 2. Comparison of $log_{10}MSE$ for 8 interpolation methods for sinusoidal test signal.

V.2 Logarithmic

Signal form: $x(t) = \log_e t\ , 1 \leq t \leq 20$

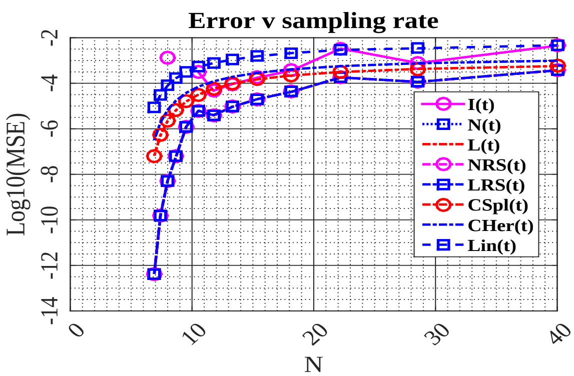

Fig. 3. Comparison of $log_{10}MSE$ for 8 interpolation methods for logarithmic test signal.

V.3 Polynomial

Signal form: $x(t) = 10t^6 - t^5 + 2t^4 - 3t^3 + 2t^2 - t + 1, \quad 1 \leq t \leq 20.$

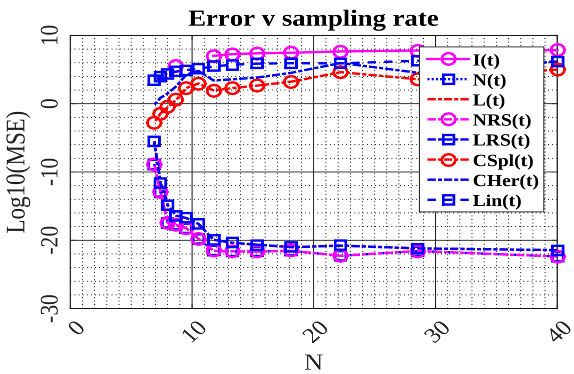

Fig. 4. Comparison of $log_{10}MSE$ of methods for polynomial test signal.

V.4 Exponential

Signal form: $x(t) = e^t, \quad 1 \leq t \leq 20$.

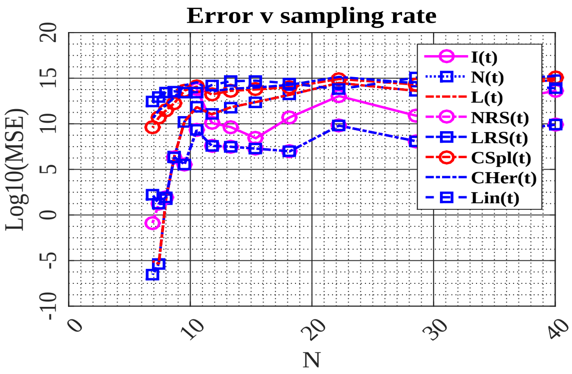

Fig. 5. Comparison of $log_{10}MSE$ for 8 interpolation methods for exponential test signal.

The four previous comparisons showed the response of different interpolation equations with regard to given points that were generated by polynomial, sinusoidal, logarithmic, and exponential functions. As depicted in Fig. 2–5, root smoothed based formula shows better performance and it adjusts itself better to new signals. In fact, an increase in sampling rate cannot always dictate lessen error even for a large variety of signals, and this has a connection to Chebyshev Points. But another consideration can be the time of calculation, which is shown in Fig. 6.

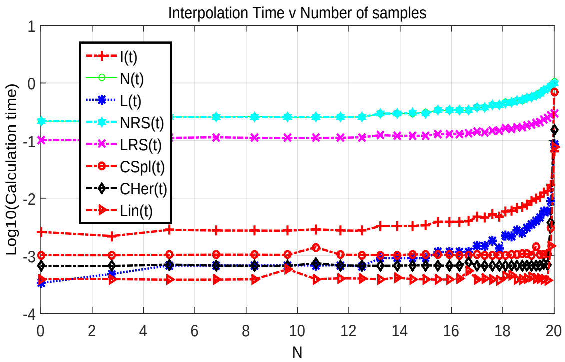

Fig. 6. Comparison of computation time of interpolation formulas versus N.

Calculation times for N$RS(t)$, L$RS(t)$, and I(t) are higher than other methods because of the complexity of the SF gin these alorithms.

VI. Conclusion

This paper studied a novel approach improving approximation and interpolation of classical formulas by means of order manipulation. The order manipulation, as a prior step to interpolation, is used to reduce error in MSE sense. The paper studied several theorems with their proofs to support the new method of order manipulation in order to use them for MIMO system approximations. The general idea considered by this paper called ST which led to generating various interpolation formulas by use of smoothers or SFs. They were also compared to other methods of interpolation. This was done in comparison part with 8 interpolation formulas and resulted in high performance of new methods with respect to their smoothing functions.

References

[1] Fliege, N.J., Multirate Digital Signal Processing, John Wiley and Sons, 1994.

[2] Oppenheim, Alan V., Ronald W. Schafer, and John R. Buck. Discrete-time signal processing. Vol. 2. Englewood Cliffs: Prentice-hall, 1989.

[3] Gunturk, Bahadir K., John Glotzbach, Yucel Altunbasak, Schafer, Ronald W., and Mersereau, Russel M., "Demosaicking: Color Filter Array Interpolation," IEEE SIGNAL PROC MAG., Vol. 22, Number 1, January 2005.

[4] Unser M., "Sampling—50 years after Shannon," Proc. IEEE, Vol. 88, No. 4, pp. 569–587, Apr. 2000.

[5] Nowak R., Baraniuk R. G., Candes E., and Vetterli M., "Compressive sampling," IEEE SIGNAL PROC MAG, Vol. 25, No. 2, pp. 12–13, 2008.

[6] E. Candes and C. Fernandez-Granda, "Towards a mathematical theory of super-resolution," Commun. Pure Appl. Math., to be published.

[7] L. Hu, Z. Shi, J. Zhou, and Q. Fu, "Compressed sensing of complex sinusoids: An approach based on dictionary refinement," IEEE T SIGNAL PROCES, vol.7, pp. 3809–3822, 2012.

[8] Atkinson K. An Introduction to Numerical Analysis, Second edition, John Wiley & Sons, New York, NY, 1989.

[9] Krogh, Fred T, "Efficient algorithms for polynomial interpolation and numerical differentiation." MATH COMPUT 24, no. 109 (1970): 185-190.

[10] Maeland E. "On the comparison of interpolation methods," IEEE T MED IMAGING, Vol. 7, No. 3, pp. 213–217, Sept. 1988.

[11] Davis P.J. Interpolation and Approximation, Second edition, Dover, New York, NY, 1975.

[12] Wang, Li-Xin, "Universal approximation by hierarchical fuzzy systems." FUZZY SET SYST 93, no. 2 (1998): 223-230.

[13] Zeng, Xiao-Jun, and G. Singh Madan, "Approximation theory of fuzzy systems-MIMO case." IEEE T FUZZY SYST. 3, no. 2 (1995): 219-235.

[14] Chak, Chu Kwong, Gang Feng, and Jian Ma, "An adaptive fuzzy neural network for MIMO system model approximation in high-dimensional spaces." IEEE T SYST MAN CY B, no. 3 (1998): 436-446.

[15] Van Dooren, Paul, Kyle A. Gallivan, and P-A. Absil, "H2-optimal model reduction of MIMO systems." APPL MATH LETT 21, no. 12 (2008): 1267-1273.

[16] Grimme, Eric James. "Krylov projection methods for model reduction," PhD diss., University of Illinois at Urbana-Champaign, 1997.

[17] Antoulas, Athanasios C., Danny C Cybernetics. Sorensen, and Serkan Gugercin, "A survey of model reduction methods for large-scale systems." COMMUN CONTEMP MATH 280 (2001): 193-220.

[18] Antoulas, Athanasios C., and Dan C. Sorensen, "Approximation of large-scale dynamical systems: An overview." Int. J. Appl. Math. Comput. Sci 11, no. 5 (2001): 1093-1122.

[19] Werner, Wilhelm. "Polynomial interpolation: Lagrange versus Newton." MATH COMPUT (1984): 205–217.

[20] Winrich, Lonny B, "Note on a comparison of evaluation schemes for the interpolating polynomial." COMPJ 12, no. 2 (1969): 154-155.

[21] Ben-Or, Michael, and Prasoon Tiwari, "A deterministic algorithm for sparse multivariate polynomial interpolation." In Proceedings of the twentieth annual ACM symposium on Theory of computing, pp. 301-309. ACM, 1988.

[22] Grigoriev, Dima Yu, Marek Karpinski, and Michael F. Singer, "Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields." SIAM J COMPUT 19, no. 6 (1990): 1059-1063.

[23] Dutt, A., M. Gu, and V. Rokhlin, "Fast algorithms for polynomial interpolation, integration, and differentiation." SIAM J NUMER ANAL 33, no. 5 (1996): 1689-1711.