Journal of Engineering Science and Technology
Vol. 19, No. 3 (2024) 894 - 910

COMPLEXITY REDUCTION OF MODEL OPERATIONS IN GENERALIZED MEMORY POLYNOMIAL FOR DIGITAL PREDISTORTION

HONG NING CHOO[1,*], SHAIFUL J. HASHIM[1,5], NURUL A. A. LATIFF[2], FAKHRUL Z. ROKHANI[1], ZUBAIDA YUSOFF[3], POORIA VARAHRAM[4]

[1]Department of Computer and Communication Systems Engineering, Faculty of Engineering, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[2]Faculty of Ocean Engineering Technology, University Malaysia Terengganu, 21300 Kuala Nerus, Terengganu, Malaysia
[3]Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia
[4]Benetel Limited, Dublin D08, Ireland
[5]Institute for Mathematical Research (INSPEM), University Putra Malaysia, 43400 Serdang, Selangor, Malaysia
*Corresponding Author: eddiechoohn@gmail.com

Abstract

Digital Predistortion (DPD) has been broadly implemented in Power Amplifier (PA) Linearization, to counter the PA non-linearity effects, which introduce additional operational costs to various PA applications such as base stations, mobile phones, and laptops. The core performance contributor of a DPD system is on its ability to accurately model the PA to acquire an inversed PA model that is used for compensating the input signals before feeding them to the PA. However, the improvement of PA modelling accuracy in DPD usually comes with a cost of increased computational requirements and additional challenges in implementations. In this paper, the popular Generalized Memory Polynomial (GMP) DPD algorithm, is optimized using the Binomial Reduction method to reduce the model operations complexity but maintaining linearization performance. The performance metrics include Normalized Mean Square Error (NMSE), where the pre-distorted PA output is measured against the ideal PA output to acquire magnitude of error in PA linearization. The NMSE measurement is applied on both original and treated algorithm, where they will both be compared. Close to 0 values indicates almost no differences among respective error magnitudes, concluding both algorithms have matching linearization performances. To measure model operations complexity, the number of multiplication operations required for each the original and treated algorithm is calculated, and then compared, where a lower number indicates fewer number of multiplication operations required, indicating lower model operations complexity. Model operations complexity is reduced 38%, with the treated GMP lagging 0.88 dB in NMSE. The difference in linearization performance is close to zero and acceptable, outweighed by the benefits observed in reduction of model operations complexity. The observed advantage would be impactful to almost all Memory Polynomial (MP) based DPD implementations in PAs, especially when PAs are increasing in importance in today's ever connecting world.

Keywords: Digital predistortion, Memory polynomial, PA linearization, Wireless communications.

1. Introduction

Power Amplifiers (PA) are prominent in wireless communications systems. Ideally, the PA is linear in terms of energy output with accordance to the level of input energy. In reality, there exists this saturation point, where the PA stops behaving linearly and starts exhibiting non-linearity, slanting away from the linear line. This results in near stagnant of output energy despite continuous increment in input energy. However, containing the PA to operate in the linear region before the saturation point results in low efficiency [1]. Adding to the complexity, today's increasing demand of bandwidth together with higher operating frequency introduces high Peak to Average Power Ratio (PAPR) to PAs. This drives the operation to happen near the PAs saturation point.

These conglomerated sources of inefficiencies and non-linearities produces signal output distortions, namely in amplitude and phase. Other undesired effects include Adjacent Channel Interference (ACI) and scatterings of the output signal called Memory Effects [2].

To neutralize the effects of PA non-linearity, PA linearization has been sought after by the academia and the industry [3]. Among the many PA linearization methods, Digital Predistortion (DPD) has emerged as the most celebrated method due to its well-balanced advantages against system cost [3-7]. The input signal is first pre-processed at the DPD system, where the DPD output is then taken in by the PA. The DPD system continues to learn and get updated, through iterations and feedbacks from the output or feedforward from the input [4, 8, 9].

The DPD block behaves as an inversed model of the PA, where it cancels out the non-linearity of the PA's function, resulting in a linearized PA output. Therefore, an accurate model of the PA in DPD would lead to in an effective PA linearization system, which will be able to nullify the amplitude and phase distortions, hence reducing ACI, as illustrated in Fig. 1.

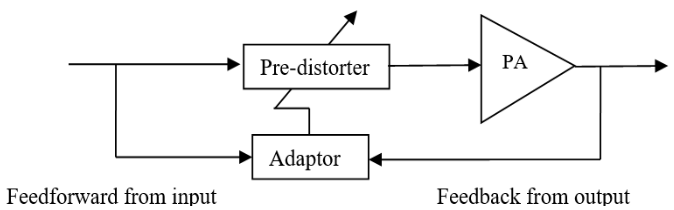

Fig. 1. DPD diagram with the feedback and feedforward path.

The accuracy of PA modelling in DPD systems usually comes at a cost. A more complex system is required to represent the non-linear modelling equation of the PA. This results an increase in model dimensions, number of model coefficients required and depth of memory in the polynomial. Conventionally, Volterra Series have been used to model non-linear PAs [10-12]. A famous improvement of Volterra Series is performed in [13, 14], where similar linearization performance is attained with reduced complexity, resulting in the Memory Polynomial (MP) algorithm.

In a research conducted by Morgan et al. [15], the MP is improved in terms of linearization accuracy. This is achieved with a cost of additional polynomial branches added together with higher model dimensions and increased number of model coefficients. The improved MP is the Generalized Memory Polynomial (GMP). Due to its improved performance, the GMP is largely adopted by the industry. GMP is popularly worked on by the academia to improve its performance [4, 8], becoming one of the most prominent MP-based DPD algorithms.

In a research conducted by Choo et al. [16], MP is reduced in terms of complexity using Binomial Reduction from [6] with savings in multiplication operations as shown in [17]. With successful optimization on MP, it is deduced in [6] that the same Binomial Reduction optimization method could be applied effectively on all DPD algorithms that evolved from MP. The objective is to achieve the reduction on model operations complexity without compromising linearization performance.

In this paper, an attempt is made to optimize another DPD algorithm derived from MP, the GMP from [15]. The treated GMP is expected to achieve reductions in model operations complexity against the original GMP algorithm, without compromising PA linearization performance. The attempt is courageous because it goes against the common trend, which is improvement in linearization performance at a cost of increased model operations complexity and implementation complexity. This paper attempts differently by reducing model operations complexity instead, while retaining the linearization performance of the DPD algorithm.

Section II describes the model description of a DPD system, diving deeper into the conventional Volterra Series algorithm, MP, GM, and the improved method, the GMP with Binomial Reduction (GMP-BR). Section III describes the performance metrics deployed for model operations complexity versus performance comparison, where error reduction capabilities is expressed with Normalized Mean Square Errors (NMSE). To evaluate model operations complexity, calculation methods for model coefficients, together with number of multiplication operations are derived and presented. Section IV includes the Results and Discussion for GMP-BR against GMP, with reference to the performance metrics described in Section III. Section V concludes the paper.

2. Model Description

To help readers appreciate the unique attempt made through this paper in reducing complexity instead of adding complexity for performance, the DPD is first described with the help of a diagram to assist readers visualize the role of the pre-distorter in pre-distorting the PA. Next the conventional non-linearity modelling algorithm, the Volterra Series is briefly presented.

The next sub-section elaborates on one of the prominent attempts in simplifying Volterra Series, the infamous Memory Polynomial (MP). Chronologically, the MP is then conveniently improved in terms of linearization performance, by increasing model operations complexity, coined as the Generalized MP. Lastly, the GMP is treated, where the reduction steps are presented so that readers may replicate the optimization process.

2.1. Digital predistortion (DPD)

In essence, the DPD Block behaves as an inversely non-linear model of the PA, where the signal input is driven into the DPD block first, and then to the PA. Figure

2 shows the respective functions of the DPD block, the PA, and the function compositions, resulting in *y*, a linearized output signal.

From Fig. 2, it is obvious that the success of DPD depended on the modelling of the PA, including its non-linearities. The next subsection explores a conventionally used algorithm in DPD to model non-linear systems [3].

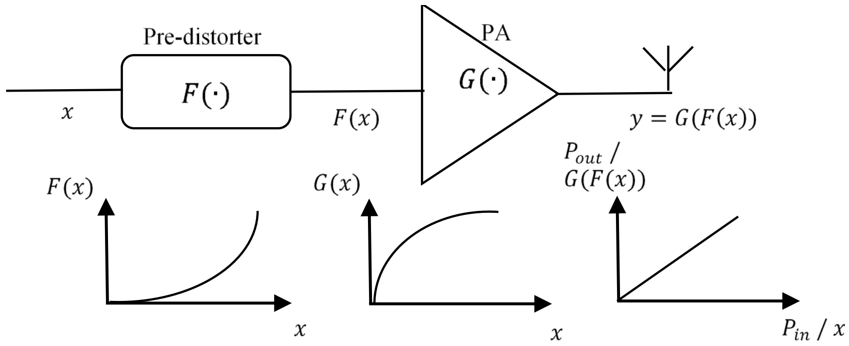

Fig. 2. Simplistic diagram of a DPD block and the PA, where the two inversely non-linear functions cancel out the non-linearities [18].

2.2. Volterra series

The Volterra Series is deployed conventionally in modelling of non-linear systems, which comes naturally useful in DPD to model the PA. The full series is shown below:

$$y(t) = \sum_k \int \cdots \int h_{2k+1}(\tau_{2k+1}) \prod_{i=1}^{k+1} z(t-\tau_i) \prod_{i=k+2}^{2k+1} z^*(t-\tau_i) d\tau_{2k+1} \quad (1)$$

where $h_{2k+1}(\tau_{2k+1}) = \frac{1}{2^{2k}} \binom{2k+1}{k} \tilde{h}_{2k+1}(\tau_{2k+1}) e^{-j2\pi\left(\sum_{i=1}^{k+1} \tau_i - \sum_{i=k+2}^{2k+1} \tau_i\right)}$

In discrete-time domain, Eq. (1) becomes the following:

$$y(n) = \sum_k \sum_{l_1} \cdots \sum_{l_{2k+1}} h_{2k+1}(l_1, l_2, \cdots, l_{2k+1}) \prod_{i=1}^{k+1} z(n-l_i) \prod_{i=k+2}^{2k+1} z^*(n-l_i) \quad (2)$$

In the Volterra Series, the No. of model coefficients, h increase exponentially, when the model dimensions: non-linearity order, k increases [13]. This results in a complex DPD model and gives incentives to researchers to optimize the Volterra Series for complexity reduction.

The next subsection presents a rather successful optimization performed on Volterra Series, where the diagonal kernels are harvested, yielding the Memory Polynomial (MP). In MP the No. of model coefficients no longer increases exponentially when the model sizes increase, resulting in a complexity reduced DPD algorithm comparing to Volterra Series.

2.3. Memory polynomial (MP)

The Memory Polynomial (MP) is shown as follows:

$$z(n) = \sum_{\substack{k=1 \\ k\ odd}}^{K} \sum_{q=0}^{Q} a_{kq} x(n-q) |x(n-q)|^{k-1} \quad (3)$$

where K is the non-linearity order, Q is the memory depth , $x(n)$ is PA input signal, and a_{kq} as the model coefficients.

To calculate the model coefficients, the Least Square Method is used by first replacing the input signal $x(n)$ with the output signal $y(n)$:

$$z(n) = \sum_{\substack{k=1 \\ k\ odd}}^{K} \sum_{q=0}^{Q} a_{kq} y(n-q)|y(n-q)|^{k-1} \tag{4}$$

Equation (4) in matrix form:

$$z = Y \cdot a \tag{5}$$

where

$$z = [z(0), z(1), \ldots, z(N-1)]^T \tag{6}$$

$$Y = \left[y_{10}, \ldots, y_{K0}, \ldots, y_{1Q}, \ldots y_{KQ}\right] \tag{7}$$

$$y_{KQ} = \left[y_{KQ}(0), y_{KQ}(1), \ldots y_{KQ}(N-1)\right]^T \tag{8}$$

$$a = \left[a_{10}, \ldots, a_{K0}, \ldots, a_{1Q}, \ldots a_{KQ}\right]^T \tag{9}$$

To obtain the model coefficients, Eq. (5) could be rewritten as:

$$a = (Y^{conj} \cdot Y)^{-1} Y^{conj} z \tag{10}$$

The MP is one of the most widely used DPD algorithms in the academia and in the industry [3]. The popularity caught the attention of many researchers, and one has added more complexity to the MP to increase accuracy, yielding Generalized MP (GMP) in the next subsection.

2.4. Generalized memory polynomial (GMP)

MP is enhanced with improved model accuracy, through increasing the number of summation branches by taking into account the lagging and leading components of the respective sampling points [15]. This enhancement is coined as the Generalized Memory Polynomial (GMP):

$$y_{GMP}(n) = \sum_{k=0}^{K_a-1} \sum_{q=0}^{Q_a-1} a_{kq} x(n-q)|x(n-q)|^k + \sum_{k=1}^{K_b} \sum_{q=0}^{Q_b-1} \sum_{l=1}^{L_b} b_{kql} x(n-q)|x(n-q-l)|^k + \sum_{k=1}^{K_c} \sum_{q=0}^{Q_c-1} \sum_{l=1}^{L_c} c_{kql} x(n-q)|x(n-q+l)|^k \tag{11}$$

where K_a Q_a are the model dimensions where the signal and envelope are aligned; where K_b Q_b L_b are the model dimensions for signal and lagging envelope; K_c Q_c $L_c$are the model dimensions for signal and leading envelope [15].

With improved model accuracy, the GMP comes with a cost of increased model complexity when compared with MP. Despite that, the GMP received popular adoption by the academia [15]. To visualize the advantage of GMP against MP, Fig. 3 shows the comparison of complexity versus performance for GMP against MP and Volterra Series with reference to [19], when stress-tested with high non-linear memory effect.

The next subsection showcases the contribution of this paper, where GMP is being optimized by reducing model operations complexity, without compromising its linearization performance.

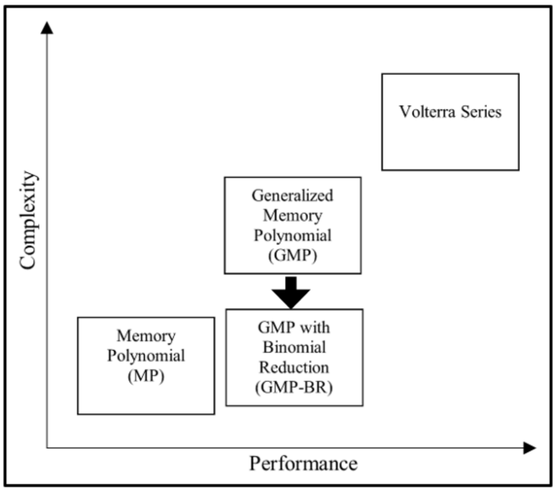

Fig. 3. Comparison between MP, GMP and Volterra Series. GMP-BR is higher in complexity compared to MP, but still lower compared to Volterra Series. While maintaining linearization performance, GMP-BR reduces complexity compared to GMP.

2.5. GMP with binomial reduction (GMP-BR)

To reduce the model complexity of GMP, the optimization method Binomial Reduction first experimented in [6] is applied. The confidence on the effectiveness of Binomial Reduction on GMP is ascertained, through the Binomial Reduction on MP that happened in [16], with reduced model complexity reported in [17] in terms of Multiplication Operations reduction.

Following the steps shown in [6, 20], the steps of Binomial Reduction for GMP is shown in the following equations.

First, Eq. (11) is rearranged, to have all k and q to start from 0.

$$y_{GMP}(n) = \sum_{k=0}^{K_a-1}\sum_{q=0}^{Q_a-1} a_{kq}x(n-q)|x(n-q)|^{k} + \sum_{k=0}^{K_b-1}\sum_{q=0}^{Q_b-1}\sum_{l=1}^{L_b} b_{kql}x(n-q)|x(n-q-l)|^{k+1} + \sum_{k=0}^{K_b-1}\sum_{q=0}^{Q_b-1}\sum_{l=1}^{L_b} b_{kql}x(n-q)|x(n-q-l)|^{k+1} \quad (12)$$

Expands the absolutes $|x(n-q)|$:

$$y_{GMP}(n) = \sum_{k=0}^{K_a-1}\sum_{q=0}^{Q_a-1} a_{kq}x(n-q)\sqrt{x(n-q)_{real}^{\ 2} + x(n-q)_{imag}^{\ 2}}^{\,k} + \sum_{k=0}^{K_b-1}\sum_{q=0}^{Q_b-1}\sum_{l=1}^{L_b} b_{kql}x(n-q)\sqrt{x(n-q-l)_{real}^{\ 2} + x(n-q-l)_{imag}^{\ 2}}^{\,k+1} + \sum_{k=0}^{K_c-1}\sum_{q=0}^{Q_c-1}\sum_{l=1}^{L_c} c_{kql}x(n-q)\sqrt{x(n-q+l)_{real}^{\ 2} + x(n-q+l)_{imag}^{\ 2}}^{\,k+1} \quad (13)$$

Rearranging Eq. (13):

$$y_{GMP}(n) = \sum_{k=0}^{K_a-1}\sum_{q=0}^{Q_a-1} a_{kq}x(n-q)\left[x(n-q)_{real}{}^2 + x(n-q)_{imag}{}^2\right]^{\frac{k}{2}} + \sum_{k=0}^{K_b-1}\sum_{q=0}^{Q_b-1}\sum_{l=1}^{L_b} b_{kql}x(n-q)\left[x(n-q-l)_{real}{}^2 + x(n-q-l)_{imag}{}^2\right]^{\frac{1}{2}}\left[x(n-q-l)_{real}{}^2 + x(n-q-l)_{imag}{}^2\right]^{\frac{k+1}{2}} + \sum_{k=0}^{K_c-1}\sum_{q=0}^{Q_c-1}\sum_{l=1}^{L_c} c_{kql}x(n-q)\left[x(n-q+l)_{real}{}^2 + x(n-q+l)_{imag}{}^2\right]^{\frac{1}{2}}\left[x(n-q+l)_{real}{}^2 + x(n-q+l)_{imag}{}^2\right]^{\frac{k+1}{2}} \quad (14)$$

Using the binomial theorem below:

$$Since\ (a+b)^h = \sum_{i=0}^{h}\binom{h}{i}a^i b^{h-i} = \sum_{i=0}^{h}\binom{h}{i}b^i a^{h-i} \quad (15)$$

Let $h = \frac{k}{2}$ for the main branch, $h = \frac{k+1}{2}$ for the leading and lagging branches, and $a = x(n-q)_{real}{}^2$ and $b = x(n-q)_{imag}{}^2$,

The basic functions of GMP are restructured as:

$$\left[x(n-q)_{real}{}^2 + x(n-q)_{imag}{}^2\right]^h = \sum_{i=0}^{h}\binom{h}{i}\left[x(n-q)_{real}{}^2\right]^i\left[x(n-q)_{imag}{}^2\right]^{h-i} \quad (16)$$

$$\left[x(n-q)_{real}{}^2 + x(n-q)_{imag}{}^2\right]^h = \sum_{i=0}^{h}\binom{h}{i}\left[x(n-q)_{imag}{}^2\right]^i\left[x(n-q)_{real}{}^2\right]^{h-i} \quad (17)$$

Reorganizing Eq. (16):

$$\sum_{i=0}^{h}\binom{h}{i}\left[x(n-q)_{real}{}^2\right]^i\left[x(n-q)_{imag}{}^2\right]^{h-i} = \sum_{i=0}^{h}\binom{h}{i}\left[x(n-q)_{real}{}^{2i}\right]\left[\frac{x(n-q)_{imag}{}^{2h}}{x(n-q)_{imag}{}^{2i}}\right] = x(n-q)_{imag}{}^{2h}\sum_{i=0}^{h}\binom{h}{i}\left[\frac{x(n-q)_{real}}{x(n-q)_{imag}}\right]^{2i} \quad (18)$$

Reorganizing Eq. (17):

$$\sum_{i=0}^{h}\binom{h}{i}\left[x(n-q)_{imag}{}^2\right]^i\left[x(n-q)_{real}{}^2\right]^{h-i} = \sum_{I=0}^{h}\binom{h}{i}\left[x(n-q)_{imag}{}^{2i}\right]\left[\frac{x(n-q)_{real}{}^{2h}}{x(n-q)_{real}{}^{2i}}\right] = x(n-q)_{real}{}^{2h}\sum_{i=0}^{h}\binom{h}{i}\left[\frac{x(n-q)_{imag}}{x(n-q)_{real}}\right]^{2i} \quad (19)$$

Extracting the binomial basis function of Eq. (18) , and represented it as

$$\sum_{i=0}^{h}\binom{h}{i}\left[\frac{x(n-q)_{real}}{x(n-q)_{imag}}\right]^{2i} = \sum_{i=0}^{h}\binom{h}{i}x^{2i} \quad (20)$$

Similarly, the binomial basis function of Eq. (19) is extracted and represented as

$$\sum_{i=0}^{h}\binom{h}{i}\left[\frac{x(n-q)_{imag}}{x(n-q)_{real}}\right]^{2i} = \sum_{i=0}^{h}\binom{h}{i}x^{2i} \quad (21)$$

Let

$$y = \sum_{i=0}^{h} \binom{h}{i} x^{2i} \approx x^{j} \quad (22)$$

Equation (22) is then expanded and explored in Figs. 4-6.

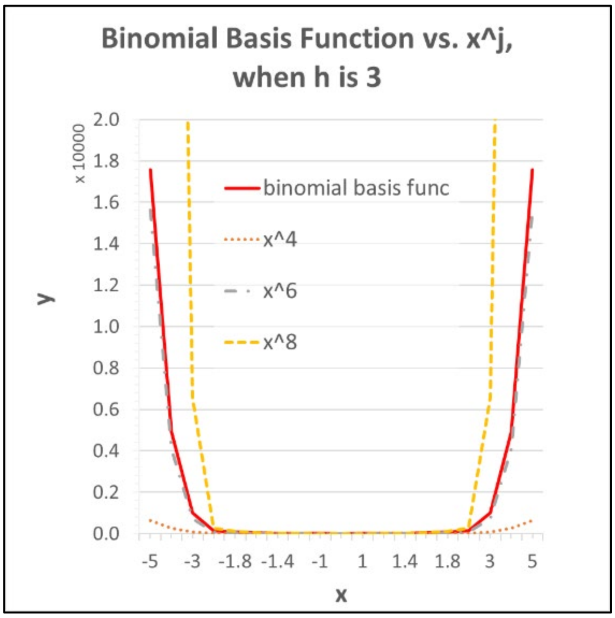

Fig. 4. Plot of Eq. (22), by letting h = 3. Notice that x^6 is almost matching with the binomial basis function in Eq. (22).

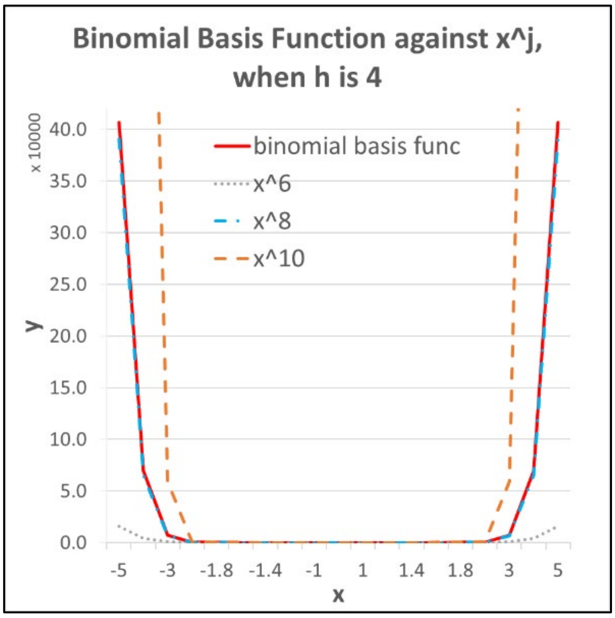

Fig. 5. Plot of Eq. (22), by letting h = 4. Notice that x^8 is almost matching with the binomial basis function in Eq. (22).

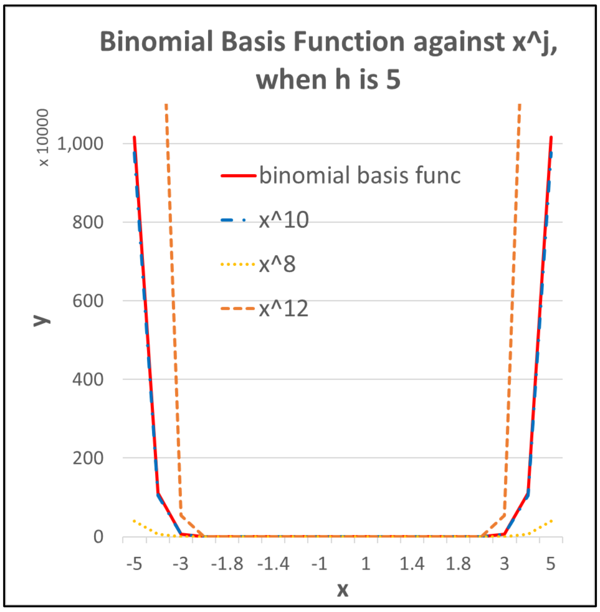

Fig. 6. Plot of Eq. (22), by letting h = 5. Notice that x^{10} is almost matching with the binomial basis function in Eq. (22).

Figures 4-6 are summarized in Table 1.

Table 1. Summary of Figs. 3-5.

h=3 (Fig. 4)	h=4 (Fig. 5)	h=5 (Fig. 6)
$y = \sum_{i=0}^{3} \binom{3}{i} x^{2i} \approx x^6$	$y = \sum_{i=0}^{4} \binom{4}{i} x^{2i} \approx x^8$	$y = \sum_{i=0}^{5} \binom{5}{i} x^{2i} \approx x^{10}$

Let

$$y = \sum_{i=0}^{h} \binom{h}{i} x^{2i} \approx x^{2h} \tag{23}$$

Substituting Eq. (23) into Eq. (20) and Eq. (21), then to Eq. (18) and Eq. (19), and finally in Eq. (14), results in GMP with Binomial Reduction below:

$$y_{GMP-BR}(n) = \sum_{k=0}^{K_a-1} \sum_{q=0}^{Q_a-1} a_{kq} x(n-q) x(n-q)_{real}{}^{k} + \sum_{k=0}^{K_b-1} \sum_{q=0}^{Q_b-1} \sum_{l=1}^{L_b} b_{kql} x(n-q) x(n-q-l)_{real}{}^{k+1} + \sum_{k=0}^{K_c-1} \sum_{q=0}^{Q_c-1} \sum_{l=1}^{L_c} c_{kql} x(n-q) x(n-q+l)_{real}{}^{k+1} \tag{24}$$

$$\boldsymbol{y_{GMP-BR}(n) = \sum_{k=0}^{K_a-1} \sum_{q=0}^{Q_a-1} a_{kq} x(n-q) x(n-q)_{imag}{}^{k} + \sum_{k=0}^{K_b-1} \sum_{q=0}^{Q_b-1} \sum_{l=1}^{L_b} b_{kql} x(n-q) x(n-q-l)_{imag}{}^{k+1} + \sum_{k=0}^{K_c-1} \sum_{q=0}^{Q_c-1} \sum_{l=1}^{L_c} c_{kql} x(n-q) x(n-q+l)_{imag}{}^{k+1}} \tag{25}$$

In the next sections, Eq. (25) will be used and addressed as GMP-BR [20] predicts relatively better linearization performance in Eq. (25) compared to Eq. (24).

The next section describes the performance metrics used to determine the improvement magnitude available after the treatment, with reference to the original untreated GMP algorithm.

3. Performance Metrics

To measure performance, the common methodologies are presented in this section. The derivations of the measurement equations are done with respect to the target algorithm in his paper: the original untreated GMP, and then on the treated GMP: GMP-BR. The derivation steps are presented clearly, for the readers to recreate and evaluate.

This section is divided into two main subsections: metrics to measure PA linearization performance, and metrics to measure model complexity for GMP and GMP-BR.

3.1. Metrics to measure PA linearization performance

The measurement of PA linearization performance has been done popularly using NMSE, where the error magnitude of the pre-distorted PA output signal is measured against the ideal output. The pre-distorted PA output signal which has a lower error magnitude against the ideal output has a relatively better linearization performance.

The following subsection shows the general NMSE measurement equation, then dives deeper into deriving the respective NMSE measurement equations for the original GMP and the treated GMP (GMP-BR).

3.1.1. Normalized mean square error (NMSE)

NMSE is calculated for the pre-distorted signal output with reference to the ideal PA output, as shown in [21]:

$$NMSE(dB) = 10\, log \frac{\sum_{n=1}^{N} |y_{ideal}(n) - y_{pd}(n)|^{2}}{\sum_{n=1}^{N} |y_{ideal}(n)|^{2}} \tag{26}$$

where y_{ideal} is the ideal PA Output, and y_{pd} is the pre-distorted PA output.

The calculated NMSE values is then compared among GMP-BR and GMP, where a lower magnitude of error indicates better linearization performance.

The NMSE calculation for GMP as used in [21] is shown below:

$$NMSE_{GMP}(dB) = 10\, log \frac{\sum_{n=1}^{N} |y_{ideal}(n) - y_{pd(GMP)}(n)|^{2}}{\sum_{n=1}^{N} |y_{ideal}(n)|^{2}} \tag{27}$$

Similarly, the NMSE calculation for GMP-BR is shown below:

$$NMSE_{GMP-BR}(dB) = 10\, log \frac{\sum_{n=1}^{N} |y_{ideal}(n) - y_{pd(GMP-BR)}(n)|^{2}}{\sum_{n=1}^{N} |y_{ideal}(n)|^{2}} \tag{28}$$

The NMSE measurements for GMP is presented. The derivation of NMSE calculation for GMP-BR is now complete. These equations would be vital in the results section later, where they will be used to evaluate the linearization performance of GMP and GMP-BR. The next section presents the metrics to measure model complexity for GMP and GMP-BR respectively.

3.2. Metrics to measure model complexity

To measure model complexity, the popular strategy used in the other DPD literatures is to compare algorithm model sizes through No. of model coefficients, which is used in [21]. On top of model sizes comparisons, this paper proposes multiplication operations as a benchmark for model operations complexity measurement, instead of the FLOP calculations, as different hardware compilers synthesize algorithms into different FLOP counts. This results in a non-standardized benchmarking which is questionable if presented. Using multiplication operations as a raw mathematical comparison, adds credibility into the model operations complexity comparison.

The following subsections shows the calculation of No. of model coefficients, and calculation of No. of multiplication operations, for both GMP and the treated GMP (GMP-BR).

3.2.1. Calculation of No. of model coefficients in GMP and GMP-BR

To evaluate model size and model complexity, one of the effective ways is to calculate the No. of model coefficients of the DPD algorithm, which is dependent on the model dimensions that indicates non-linearity order and memory depth. As presented in [21], the calculation of the model coefficients of GMP is shown below:

$$No.of\ Model\ Coefficients\ in\ GMP = K_a Q_a + \left((K_b - 1)(Q_b)(L_b)\right) + \left((K_c - 1)(Q_c)(L_c)\right) = No.of\ Model\ Coefficients\ in\ GMP - BR \quad (29)$$

The calculation of model coefficients of GMP-BR would be identical of GMP as shown in Eq. (29).

The model coefficients calculation equation will be used later in the results section, to indicate the model size of the DPD algorithms. The next subsection presents the calculation of No. of multiplications, to compare model operations complexity.

3.2.2. Calculation of No. of multiplication operations in GMP

To represent the model complexity of GMP in Eq. (13), the multiplication operations calculations are as follows:

$$\begin{aligned} &No.of\ Multiplication\ Operations\ in\ GMP \\ &= \sum_{k=1}^{K_a}\sum_{q=1}^{Q_a}\left(1+3(k)\right) + \sum_{k=1}^{K_b}\sum_{q=1}^{Q_b}\sum_{l=1}^{L_b}\left(1+3(k)\right) + \sum_{k=1}^{K_c}\sum_{q=1}^{Q_c}\sum_{l=1}^{L_c}\left(1+3(k)\right) \\ &= \frac{1}{2}K_a(3K_a+5)Q_a + \frac{1}{2}K_b(3K_b+5)Q_bL_b + \frac{1}{2}K_c(3K_c+5)Q_cL_c \end{aligned} \quad (30)$$

Since

$$\sum_{k=1}^{K_a}\sum_{q=1}^{Q_a}\left(1+3(k)\right) = \frac{1}{2}K_a(3K_a+5)Q_a$$

and

$$\sum_{k=1}^{K_b}\sum_{q=1}^{Q_b}\sum_{l=1}^{L_b}\big(1+3(k)\big)=\frac{1}{2}K_b(3K_b+5)Q_bL_b$$

and

$$\sum_{k=1}^{K_c}\sum_{q=1}^{Q_c}\sum_{l=1}^{L_c}\big(1+3(k)\big)=\frac{1}{2}K_c(3K_c+5)Q_cL_c$$

The equation above will be used to evaluate the model operations complexity of GMP. The No. of multiplication operations equation derived for GMP is a direct function of the model operations complexity in GMP, which will be compared with GMP-BR in the results section. The calculation of No. of multiplication operations in GMP-BR is presented in the next subsection.

3.2.3. Calculation of No. of multiplication operations in GMP-BR

Similarly, the multiplication operations calculations for GMP-BR in Eq. (25) are shown below:

$$\begin{aligned} &No.\,of\ Multiplication\ Operations\ in\ GMP-BR \\ &=\sum_{k=1}^{K_a}\sum_{q=1}^{Q_a}\big(1+1(k)\big)+\sum_{k=1}^{K_b}\sum_{q=1}^{Q_b}\sum_{l=1}^{L_b}\big(1+1(k+1)\big)+\sum_{k=1}^{K_c}\sum_{q=1}^{Q_c}\sum_{l=1}^{L_c}\big(1+1(k+1)\big) \\ &=\frac{1}{2}K_a(K_a+3)Q_a+\frac{1}{2}K_b(K_b+5)Q_bL_b+\frac{1}{2}K_c(K_c+5)Q_cL_c \end{aligned} \quad (31)$$

Since

$$\sum_{k=1}^{K_a}\sum_{q=1}^{Q_a}\big(1+1(k)\big)=\frac{1}{2}K_a(K_a+3)Q_a$$

and

$$\sum_{k=1}^{K_b}\sum_{q=1}^{Q_b}\sum_{l=1}^{L_b}\big(1+1(k+1)\big)=\frac{1}{2}K_b(K_b+5)Q_bL_b$$

and

$$\sum_{k=1}^{K_c}\sum_{q=1}^{Q_c}\sum_{l=1}^{L_c}\big(1+1(k+1)\big)=\frac{1}{2}K_c(K_c+5)Q_cL_c$$

Now, both equations to calculate No. of multiplications operations are derived for GMP and GMP-BR. Both equations will be used later in the results section, where the model with the lower No. of multiplication operations will be highlighted. The model with a lower No. of multiplication operations will have lower model operations complexity. The next section presents the results obtained, which will include No. of model coefficients, No. of multiplication operations, and NMSE for both GMP and GMP-BR. Discussions and comparisons will be conducted between the original GMP, and the binomially reduced GMP (GMP-BR).

4. Results and Discussion

Table 2 shows the model dimensions fed into both GMP and GMP-BR, resulting in both models have identical model coefficients of 38. As mentioned in the previous section, No. of model coefficients are direct indicators of model sizes, as it is a direct function of the model dimensions of the algorithm. When both GMP and GMP-BR has the same No. of model coefficients of 38, this indicates that both simulation models have identical model size, which gives a common ground to conduct the comparison on model operations complexity.

It is observed that GMP reported to have 368 multiplication operations, while GMP-BR requires only 226 multiplication operations. The difference of number of required multiplication operations between GMP and GMP-BR is at 142, which is a reduction of 38% against the required multiplication operations in the original untreated GMP algorithm. A reduction in the number of multiplications operations yields a reduction in model operations complexity, which contributes to savings in DPD operational cost and improvement in system efficiency.

Both simulation produces almost similar NMSE values of -20.29 dB for GMP, and -19.41 dB for GMP-BR. The NMSE between GMP and GMP-BR is at a negligible difference of 0.88 dB, indicating that they both have almost matching linearization performance. This also signifies the success of the Binomial Reduction in GMP, where linearization performance is retained, with reduction in model operations complexity.

Table 2. NMSE and model operations complexity comparison between GMP and GMP-BR.

Method	Model Dimensions	No. of Model Coefficients	No. of Multiplication Operations	NMSE (dB)
GMP Eq. (13) [15]	$K_a(1)$ $K_b(2)$ $K_c(3)$ $Q_a(2)$ $Q_b(4)$ $Q_c(6)$ $L_b(3)$ $L_c(5)$	38	368	-20.29
GMP-BR Eq. (25)	$K_a(1)$ $K_b(1)$ $K_c(4)$ $Q_a(2)$ $Q_b(1)$ $Q_c(2)$ $L_b(2)$ $L_c(6)$	38	226	-19.41

Figure 7 shows the number of Multiplication Operations for GMP and GMP-BR, with respect to the number of Model Coefficients. With increasing model sizes, GMP-BR is shown to have lower model operations complexity compared to GMP.

The binomially reduced algorithms could be compared with other optimization methods as well, to yield more meaningful comparison in terms of performance versus resources trade-off.

Comparisons could be done on the effectiveness of various optimization methodologies, in axis of linearization performance against resources required. The common trend was improving performance at a cost of resources increment. The ideal methodology would be the ones that increases performance while reducing required resources. Figure 8 helps visualizes the comparison proposal.

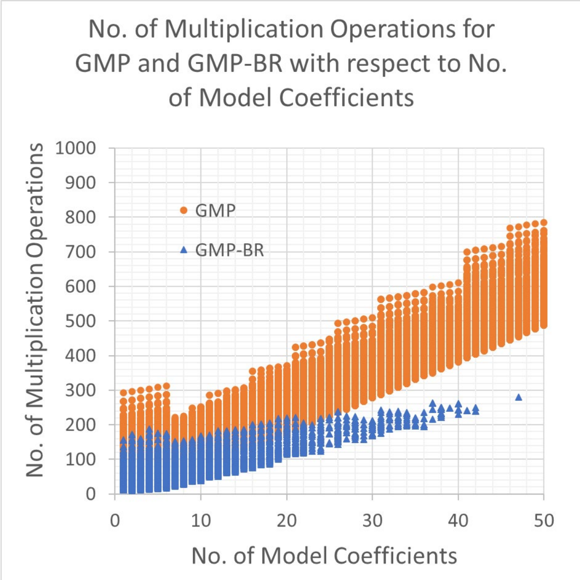

Fig. 7. Number of multiplication operations for GMP and GMP-BR, with respect to the number of model coefficients.

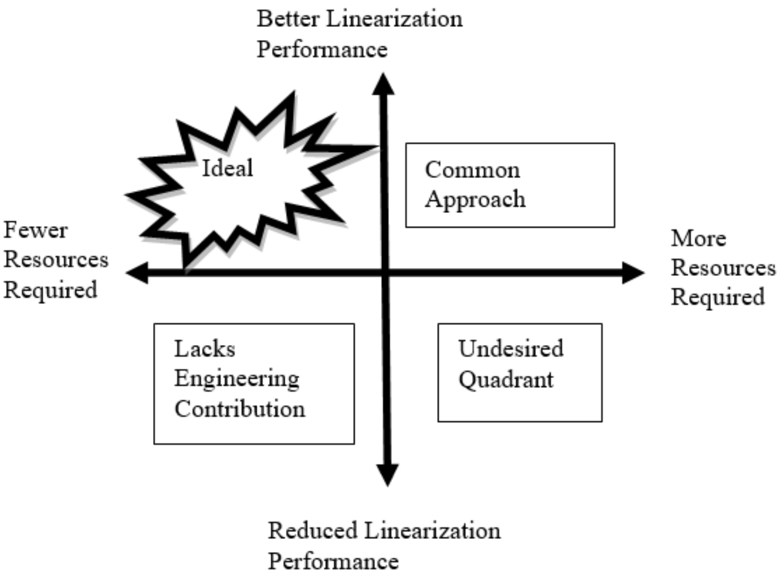

Fig. 8. Linearization performance against resources required quadrants for DPD algorithm optimization comparison. The ideal optimization methodology reduces resources required but improves linearization performance.

5. Conclusions

Binomial Reduction on GMP (GMP-BR) is proven to be effective, where model operations complexity is reduced while retaining linearization performance. The number of multiplication operations required is reduced 38% from 368 in GMP to

226 in GMP-BR. The linearization performance evaluation is as presented in the NMSE comparison, where GMP-BR is at -19.41 dB and GMP is at -20.29 dB, at a minute difference of 0.88 dB.

The results indicate that the binomially reduced GMP (GMP-BR) requires fewer multiplication operations compared to the original untreated method, the GMP. The savings could go as high as 38%, resulting in reduced number of multiplication operations, better energy efficiency in DPD systems and better resource utilization. All these savings are made possible, without compromising linearization performance.

For future work, the Binomial Reduction could be applied at other MP-based algorithms. Some of the prominent MP based algorithms include the Complexity Reduced GMP (CR-GMP) and Augmented CR-GMP (ACR-GMP) in [21]. Leveraging from the success scenarios in this paper, the complexity reduction has a high success rate, resulting in a more efficient DPD system. The challenges of implementing Binomial Reduction on other MP-based algorithms, would be on having an accurate estimation on the range of reductions in model operations complexity. The range of reductions depended upon the algorithm design of the target algorithm. Estimation of the model operations complexity reduction could probably be another fork in the research roadmap.

The Binomial Reduction of GMP paves new possibilities in the direction of DPD algorithm research. Instead of conveniently increasing model operations complexity to reach higher accuracy, this paper showcases that the alternative is possible, where linearization performance could be maintained, while attempting to reduce complexity in model operations. The findings in this paper on 38% complexity reduction with negligible 0.88 dB delta in NMSE, gives more confidence for other researchers in DPD algorithms to attempt model complexity reductions, instead of pursuing the conventional accuracy improvement through DPD algorithms augmentations.

Acknowledgements

This work was supported in part by University Putra Malaysia Grant FRGS/2/2013/TK02/UPM/02/3 and GP-IPS/2022/9735000.

References

1. Masterson, C. (2022). Digital predistortion for RF communications: From equations to implementation. *Analog Dialogue*, 56(2), 1-5.
2. Hadi, N.S.R.; Yusoff, Z.; Sadeque, M.G.; Hashim, S.J.; and Chaudhary, M.A. (2020). High gain over an octave bandwidth class-F RF power amplifier design using 10W GaN HEM. *Bulletin of Electrical Engineering and Informatics*, 9(5) 1899-1906.
3. Katz, A.; Wood, J.; and Chokola, D. (2016). The evolution of PA linearization: From classic feedforward and feedback through analog and digital predistortion. *IEEE Microwave Magazine*, 17(2), 32-40.
4. Wood, J. (2017). Digital pre-distortion of RF power amplifiers. *Proceedings of the* 2017 *IEEE Topical* Conference *on RF/Microwave Power Amplifiers for Radio and Wireless Applications* (*PAWR*2017), Phoenix, AZ, USA, 1-3.

5. Choo, H.N.; Varahram, P.; and Ali, B.M. (2013). DSP implementation of digital pre-distortion in wireless systems. *Proceedings of the* 2013 *IEEE* 11*th* *Malaysia International Conference on Communications* (*MICC*2013), Kuala Lumpur, Malaysia, 207-212.
6. Choo, H.N. (2017). *Enhanced memory polynomial with reduced* complexity *in digital pre-distortion for wireless power amplifier*. MSc dissertation, Department of Electrical Engineering, University Putra Malaysia.
7. Gilabert, P.L.; Braithwaite, R.N.; and Montoro, G. (2020). Beyond the Moore Penrose inverse: Strategies for the estimation of digital predistortion linearization parameters. *IEEE Microwave Magazine*, 21(12), 34-46.
8. Gilabert, P.L.; Montoro, G.; Vegas, D.; Ruiz, N.; and Garcia, J.A. (2019). Digital predistorters go multidimensional: DPD for concurrent multiband envelope tracking and outphasing power amplifiers. *IEEE Microwave Magazine*, 20(5), 50-61.
9. Ramsey, A.F. (2020). *An implementation of the redirected learning architecture for digital pre-distortion*. MSc dissertation, Department of Electrical Engineering, Iowa State University.
10. Guan, L.; and Zhu, A. (2010). Low-cost FPGA implementation of volterra series-based digital predistorter for RF power amplifiers. *IEEE Transactions on Microwave Theory and Techniques*, 58(4), 866-872.
11. Cheng, C.M.; Peng, Z.K.; Zhang, W.M.; and Meng, G. (2017). Volterra-series-based nonlinear system modeling and its engineering applications: A state-of-the-art review. *Mechanical Systems and Signal Processing*, 87, 340-364.
12. Braithwaite, R.N. (2017). Digital predistortion of an RF power amplifier using a reduced volterra series model with a memory polynomial estimator. *IEEE Transactions on Microwave Theory and Techniques*, 65(10), 3613-3623.
13. Ding, L. (2004). *Digital predistortion of power amplifiers for wireless applications*. PhD Thesis, Department of Electrical and Computer Engineering, Georgia Institute of Technology.
14. Ding, L.; Zhou, G.T.; Morgan, D.R.; Ma, Z.; Kenney, J.S.; Kim, J.; and Giardina, C.R. (2004). A robust digital baseband predistorter constructed using memory polynomials. *IEEE Transactions on Communications*, 52(1), 159-165.
15. Morgan, D.R.; Ma, Z.; Kim, J.; Zierdt, M.G.; and Pastalan, J. (2006). A generalized memory polynomial model for digital predistortion of RF power amplifiers. *IEEE Transactions on Signal Processing*, 54(10), 3852-3860.
16. Choo, H.N.; Latiff, N.A.A.; Varahram, P.; and Ali, B.M. (2016). Error reduction of memory polynomial with binomial reduction in digital pre-distortion for wireless communication systems. *Proceedings of the* 2016 *International Conference on Information and Communication Technology Convergence* (*ICTC*2016), Jeju, Korea (South), 505-510.
17. Choo, H.N.; Latiff, N.A.A.; Varahram, P.; and Ali, B.M. (2018). Operations optimization of memory polynomial with binomial reduction in digital pre-distortion for wireless communication systems. *Proceedings of the* 2018 *IEEE Symposium on Computer Applications & Industrial Electronics* (*ISCAIE*2018), Penang, Malaysia , 291-296.

18. Gilabert Pinal, P.L. (2008). *Multi look-up table digital predistortion for RF power amplifier linearization*. PhD Thesis, Department of Signal Theory and Communications, Universitat Politècnica de Catalunya.
19. Ghannouchi, F.M.; Hammi, O.; and Helaoui, M. (2015). *Memory polynomial based models*. In Ghannouchi, F.M.; Hammi, O.; and Helaoui, M. (Eds), *Behavioral modeling and predistortion of wideband wireless transmitters*. John Wiley & Sons, Inc, 88-114.
20. Choo, H.N.; Latiff, N.A.A; Varahram, P.; and Ali, B.M. (2017). Memory polynomial with binomial reduction in digital pre-distortion for wireless communication systems. *Pertanika Journal of Science & Technology*, 25(S), 49-58.
21. Liu, Y.J.; Zhou, J.; Chen, W.H.; and Zhou, B.H. (2014). A robust augmented complexity-reduced generalized memory polynomial for wideband RF power amplifiers. *IEEE Transactions on Industrial Electronics*, 61(5), 2389-2401.