Kernel-Based Digital Predistortion: An Approach with the EX-QKRLS Algorithm

Isaac Macario da Silva de Gouveia, José Antonio Apolinário Jr. and Cláudio Augusto Barreto Saunders Filho

***Abstract*— This paper investigates the effectiveness of kernel-based adaptive filtering for digital predistortion in power amplifiers. Specifically, the EX-QKRLS algorithm, which incorporates concepts from the Kalman filter, is employed to model and compensate for the nonlinearities introduced during signal amplification, without requiring memory polynomials or lookup tables. This method reduces overall signal distortion and improves transmission fidelity robustly and efficiently. Unlike conventional approaches, the proposed technique offers high adaptability, enabling rapid adaptation to changing system conditions, including temperature fluctuations and component aging.**

***Keywords*— Adaptive Filters, Digital Predistortion, Kalman Filter, Kernel, Power Amplifier, RF Weblab.**

I. INTRODUCTION

Power amplifiers (PA) play a crucial role in wireless communication systems. Due to their non-linear nature, the output signal suffers distortion when the amplifier operates close to saturation. In this scenario, the range is reduced, increasing the power consumption [1][2].

One technique to be used is digital predistortion (DPD), which consists of modifying the signal after estimating the PA response, so that, when amplified, the signal is transmitted as initially intended. DPD can be done using adaptive filters, since precise modeling of the nonlinearities to which the signal is exposed during amplification is necessary. However, estimating the amplifier's output can be computationally expensive, since the type of nonlinearity produced by the PA is unknown [3].

Currently, digital predistortion commonly relies on techniques that compensate for amplifier nonlinearities using series-based models, such as Volterra, memory polynomials, and Wiener-Hammerstein, or lookup tables (LUTs) [3] [4]. While effective, these approaches require prior knowledge of the distortion characteristics. Series-based models, for instance, demand the estimation of the polynomial order and memory depth, that is, how much influence past input samples have on the current output. LUT-based methods, on the other hand, require detailed mapping of the signal's amplitude and phase. Moreover, these models are often sensitive to temperature variations and long-term component aging, which reduces their robustness in real-world applications.

Variants of these models have been explored in recent studies, such as [5], [6], and [7]. As an alternative, this work proposes a kernel-based adaptive filtering approach for power amplifier linearization, using the EX-QKRLS (Extended Quantized Kernel Recursive Least Squares) algorithm. This method enables nonlinear modeling and predistortion without requiring prior estimation of the nonlinearity degree, making the process more automatic, adaptive, and flexible.

In this article, Section II presents a theoretical overview of the main concepts discussed. The algorithm is presented in Section III and Section IV shows the experiment's setup. The results are shown in Section VI and the final section presents conclusions and suggestions for future work.

II. FUNDAMENTALS OF DIGITAL PREDISTORTION WITH ADAPTIVE FILTERING

This section presents the fundamentals used in the implementation of DPD based on the algorithms explored in this work. The structures for modeling nonlinearities will be described, including memory polynomials, adaptive filtering algorithms such as the traditional recursive least squares algorithm (RLS) and its kernel-based version, and the Kalman filter.

A. Memory Polynomial

For the treatment of non-linear signals, techniques are created for modeling non-linearity that allow the use of adaptive filtering algorithms.

Since the nonlinearity introduced by the power amplifier can be modeled by polynomial expressions, the most widely implemented technique for this framework is the use of models based on the Volterra series [8]. This series can be interpreted as a Taylor series with memory [10] and is described by Equation (1).

$$\begin{aligned} y(k) = &\textstyle\sum_{l_1=0}^{\infty} w_1(l_1)u(k-l_1)+ \\ &+\textstyle\sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty} w_2(l_1,l_2)u(k-l_1)u(k-l_2)+ \\ &+\textstyle\sum_{l_1=0}^{\infty}\sum_{l_2=0}^{\infty}\cdots\sum_{l_i=0}^{\infty} w_i(l_1,l_2,...,l_i)u(k-l_1)...u(k-l_i), \end{aligned} \quad (1)$$

where $w_i(l_1, l_2, ..., l_i)$ are the coefficients of the nonlinear filter based on the Volterra model, $y(k)$ represents the noise-free system output, and $u(k-l_i)$ the input signal.

The biggest problem with this technique is its rapid growth in terms of coefficients, which, in some cases, can generate very high computational complexity. To get around this situation, the series is truncated in a certain order, or a specific term is suppressed. There are several types of reductions based on this series, but the most common for DPD systems being the memory polynomial (MP). [9], which is described in Equation (2).

$$y_{MP}(k) = \sum_{p=1}^{P-1}\sum_{m=0}^{M-1} w_{p,m}u(k-m)|u(k-m)|^p \quad (2)$$

The authors are with the Defense Engineering Program (PGED) and Electrical Engineering Program (PEE) of the Military Institute of Engineering (IME). This work was partially funded by the Brazilian Army.

For applications where memory effects are more significant or extend over a larger number of samples, it may be necessary to include more terms in the series. From there, cross terms are added to the MP; this alternative is called Cross-Term Memory (CT) and follows Equation (3).

$$y_{CT}(k) = \sum_{p=1}^{P-1}\sum_{m=0}^{M-1} w_{1p,m}u(k-m)|u(k-m)|^p + \\ + \sum_{p=1}^{P-1}\sum_{m=0}^{M-1}\sum_{l=1}^{L-1} w_{2p,m,l}u(k-m)|u(k-m-l)|^p \quad (3)$$

B. The RLS Algorithm

The recursive least squares algorithm is one of the main adaptive filtering algorithms because it presents a fast convergence to the optimal coefficient vector. This algorithm is based on the recursive update of the correlation matrix [10], according to

$$\mathbf{w}(k) = \left[\sum_{i=0}^{k} \lambda^{k-i}\mathbf{u}(i)\mathbf{u}^{\mathrm{H}}(i)\right]^{-1} \sum_{i=0}^{k} \lambda^{k-i}\mathbf{u}(i)d(i) = \mathbf{R}_D^{-1}(k)\mathbf{p}_D(k), \quad (4)$$

where $\mathbf{R}_D$ and $\mathbf{p}_D$ are called deterministic correlation matrix and deterministic cross-correlation vector, $d(i)$ is the desired value at the instant i, λ is the forgetting factor, usually set in the range $0 \ll \lambda \leq 1$, and $\mathbf{u}(i)$ represents the input vector at time i, composed of past samples of the input signal $u(k)$.

In contrast to the speed of convergence, the RLS algorithm presents a great computational effort, due to the calculation of the inverse of matrix $\mathbf{R}_D$ [11]. Furthermore, depending on its implementation and on the desired input data, it may be unstable [12].

C. Kernel-Based Adaptive Filtering

Kernel-based adaptive filtering (KAF) is a signal processing technique that combines adaptive filtering with kernel theory. Its main feature is the kernel trick, which allows operations to be performed on a high-dimensional vector space without explicitly mapping the data, thus enabling nonlinear problems to be treated linearly [13].

In this space, known as reproducing kernel Hilbert space (RKHS), the similarity between vectors is computed via a symmetric, non-negative kernel function [14]. The kernel function can take several forms, depending on the type of nonlinearity or the quality of estimation required. The most commonly used function, due to its universal approximation capability and numerical stability, is the Gaussian function [15]. The Gaussian model is often employed in nonlinear signal processing; its form, for complex values, is expressed by

$$\kappa(\mathbf{u}(k), \mathbf{u}(l)) = e^{-\frac{1}{2}\sum_{i=0}^{I}\frac{1}{\sigma_i^2}(u_i(k)-u_i^*(l))^2}, \quad (5)$$

where the coefficient σ is called the kernel bandwidth and represents the interaction range of the kernel.

Although the kernel function can be computationally expensive for large datasets, kernel adaptive filtering (KAF) achieves promising results in modeling nonlinear signals. While Volterra-based models suffer from rapidly increasing complexity with the order and memory depth, kernel adaptive filters can provide similar accuracy with lower computational cost [14].

D. Discrete Kalman Filter

The Kalman filter is a widely used state estimation technique due to its ability to provide accurate estimates even in noisy and uncertain environments. Its operation is based on two main steps: prediction and updating [16].

In the prediction stage, the future state of the system is estimated based on the dynamic model and the current state. This process can be described by the state equation, represented in Equation (6).

$$\mathbf{x}(k+1) = \mathbf{F}(k+1, k)\mathbf{x}(k) + \mathbf{G}(k)\mathbf{r}(k), \quad (6)$$

where $\mathbf{x}(k)$ represents the state vector in time k, $\mathbf{F}(k+1, k)$ and $\mathbf{G}(k)$ are known transition matrices, and $\mathbf{r}(k)$ is the process noise, which represents the uncertainties associated with the system dynamics.

After prediction, the state estimate is corrected in the update stage. In this phase, the available measurements and their respective uncertainties are taken into account. This update is performed using the observation equation, Equation (7).

$$y(k) = \mathbf{u}^{\mathrm{H}}(k)\mathbf{x}(k) + \nu(k), \quad (7)$$

where $y(k)$ is the observed output of the system and $\nu(k)$ corresponds to the measurement noise. This term captures the external disturbances that affect the measurements and must be considered to ensure a robust estimate.

In this way, the Kalman filter continuously refines the prediction of the system states and corrects it based on observations, allowing efficient modeling of dynamic systems under noise. This approach is particularly useful in telecommunications, signal processing, and system control applications, where accuracy in state estimation is essential.

Although the classical Kalman filter was developed for systems with linear state and measurement equations, many practical problems involve nonlinear dynamics. In such cases, the extended Kalman filter (EKF) is applied, which linearizes the equations in a region close to the currently available state estimate [10]. The main drawback of the EKF is its computational complexity, which increases for nonlinear systems [10].

III. The EX-QKRLS Algorithm

The algorithm proposed in this work, EX-QKRLS, is a derivation of the EX-KRLS algorithm, presented in [17], with the addition of the quantization technique. Both concepts are discussed below.

A. The EX-KRLS Algorithm

The EX-RLS or extended RLS algorithm [18] is a variation of the RLS algorithm that makes it fully equivalent to a Kalman filter. When used in kernel-based filtering, we have the EX-KRLS algorithm. The EX-KRLS algorithm follows the state-space equations presented in the previous section, but uses the vector $\boldsymbol{\varphi}(k)$ as an input signal, instead of $\mathbf{u}(k)$, where $\boldsymbol{\varphi}(k)$ belongs to Hilbert space and obeys the relation $\kappa(\mathbf{u}, \mathbf{u}') = \boldsymbol{\varphi}^{\mathrm{T}}(\mathbf{u})\boldsymbol{\varphi}(\mathbf{u}')$.

This modification gives EX-KRLS a higher performance than KRLS, especially in dynamic tracking applications, where parameters or environmental conditions vary over time [14]. In such cases, the quick adaptation capacity is critical to ensuring that the model correctly follows these variations.

Despite its fast convergence, EX-KRLS can become unstable when processing a large number of samples, due to increasing computational complexity and numerical error propagation.

B. Sparcification

As explained in the previous section, the EX-KRLS algorithm is computationally demanding because, being an RLS algorithm, it requires matrix inversion operations, which significantly increase its computational cost. Therefore, its execution becomes slow, compromising the performance of DPD, especially in real-time applications.

However, not all samples contribute significantly to the signal estimate. In this scenario, Platt's novelty criterion [19] is introduced. This criterion, commonly used in radial basis functions, operates under the assumption that points close to each other in the input space convey similar information and can therefore be represented by a single point.

We define the difference between two vectors as $dist = min(||\mathbf{u}(k) - \mathbf{u}(l)||)$. Knowing the distance between samples allows us to define a threshold that determines whether a sample is relevant to the system. In this work, this threshold is named δ. In the implemented algorithm, if the sample has a distance greater than δ, it is incorporated into the model; otherwise, the sample is discarded and the weights of the samples already stored are updated [20]. This process is known as quantization and is represented by the letter Q in the acronym EX-QKRLS[1].

If the parameter δ is set to a very low value, more samples will be considered relevant for the algorithm, resulting in behavior similar to EX-KRLS, which produces accurate results but is computationally heavy. Conversely, a larger δ would consider fewer samples, reducing the number of operations, but at the cost of estimation accuracy.

To balance this trade-off is to initialize δ as a line with a positive slope that passes through the origin. After a certain number of samples, δ becomes constant. This strategy favors early samples, establishing a baseline for modeling and, once this level is reached, it becomes more selective in its choices.

In this way, EX-QKRLS is described in Algorithm I, with inputs: input vector $\mathbf{u}$, forgetting factor λ, noise variance ratio q, transition matrix scale factor α, regularization parameter Π, kernel bandwidth σ_k, and maximum threshold δ_{max}. The kernel function is denoted by $\kappa(\cdot,\cdot)$ and the dictionary vector $\mathbf{D}$ stores samples selected as relevant according to Platt's criterion. Vector $\mathbf{a}$ contains the filter coefficients, while $\mathbf{Q}$ and ρ are auxiliary variables updated recursively to ensure stability and weighting of new information. The outputs of this algorithm are $\mathbf{y}(k)$, the estimated output, and $e(k)$, the error between the desired output $d(k)$ and the filter output.

[1]The derivation of the EX-QKRLS algorithm, as well as the codes used in this article — all implemented in MATLAB® — are available in the DPD repository, accessible at `https://github.com/isaacmacario2/DPD`.

Algorithm I: EX-QKRLS Algorithm

Initialization

$\mathbf{D}(1) = \mathbf{u}(1)$, $\mathbf{a}(1) = \frac{\alpha d(1)}{\Pi\lambda + \kappa(\mathbf{u}(1),\mathbf{u}(1))}$, $\delta = 0$, $m = \frac{\delta_{max}}{\text{length}(\mathbf{u})}$,

$\rho(1) = \frac{\Pi\lambda}{(|\alpha|^2\lambda + \Pi q)}$, $\mathbf{Q}(1) = \frac{|\alpha|^2}{[\Pi\lambda + \kappa(\mathbf{u}(1),\mathbf{u}(1))][|\alpha|^2 + \Pi\lambda q]}$

for each k
  for each l
    $\mathbf{h}(l) = \kappa(\mathbf{u}(k), \mathbf{D}(l))$
    $\mathbf{y}(k) = \mathbf{y}(k) + \mathbf{h}^{\mathrm{H}}(l) * \mathbf{a}(l)$
  end
  $[dist, idx] = \min(\|\mathbf{u}(k) - \mathbf{D}(l)\|)$
  if $dist \geq \delta$
    $\mathbf{D} = [\mathbf{D} \;\; \mathbf{u}(k)]$
    $\mathbf{z}(k) = \mathbf{Q}(k-1)\mathbf{h}(k)$
    $r = \lambda^k \rho(i-1) + \kappa(\mathbf{u}(k), \mathbf{u}(k)) - \mathbf{h}^{\mathrm{H}}(k)\mathbf{z}(k)$
    $e(k) = d(k) - \mathbf{h}^{\mathrm{H}}(k)\mathbf{a}(k-1)$
    $\mathbf{a}(k) = \alpha \begin{bmatrix} \mathbf{a}(k-1) - \mathbf{z}(k)r^{-1}(k)e(k) \\ r^{-1}(k)e(k) \end{bmatrix}$
    $\rho(k) = \frac{\rho(k-1)}{|\alpha|^2 + \lambda^k q \rho(k-1)}$
    $\mathbf{Q}(k) = \frac{|\alpha|^2}{r(k)(|\alpha|^2 + \lambda^k q \rho(k-1))} \begin{bmatrix} \mathbf{Q}(k-1)r(k) + \mathbf{z}(k)\mathbf{z}^{\mathrm{H}}(k) & -\mathbf{z}(k) \\ -\mathbf{z}^{\mathrm{H}}(k) & 1 \end{bmatrix}$
  else
    $e(k) = d(k) - \mathbf{h}^{\mathrm{H}}(k)\mathbf{a}(k-1)$
    $\mathbf{a}(idx) = \mathbf{a}(idx) + e$
  end
  if $\delta < \delta_{max}$
    $\delta = m * k$
  else
    $\delta = \delta_{max}$
  end
end

To compare execution speed, EX-KRLS and EX-QKRLS were applied to a random stationary signal with variance 0.1. The reference signal was generated using a third-degree polynomial nonlinearity with memory 2, corrupted by additive white Gaussian noise (variance 0.001). With $\sigma_k = 0.5$, 1000 samples, and 500 Monte Carlo runs, runtime was measured using MATLAB®'s functions *tic* and *toc*, varying parameter δ. Accuracy was assessed by the average difference in MSE values (in dB). MSE quantifies the mean squared deviation between the estimated and desired outputs and is commonly used to evaluate adaptive filters. Results are shown in Table I, and the learning curve in Figure 1.

TABLE I

ALGORITHM RUNTIME COMPARISON AND ACCURACY MSE DEVIATION AFTER CONVERGENCE

	$\delta = 0.4$	$\delta = 0.7$	$\delta = 1$
EX-KRLS	4.311 s	4.311 s	4.311 s
EX-QKRLS	3.434 s	1.951 s	1.451 s
MSE Difference	1.344 dB	2.482 dB	3.326 dB

IV. EXPERIMENTAL SETUP

In this section, the performance of the proposed model will be evaluated. The following results are obtained through implementations made in MATLAB®.

To perform tests with real measurements, RF WebLab [21] was used, a remote signal amplification platform developed through a partnership between Chalmers University (Sweden) and National Instruments (NI). The system provides access to high-performance equipment for testing linearization algorithms.

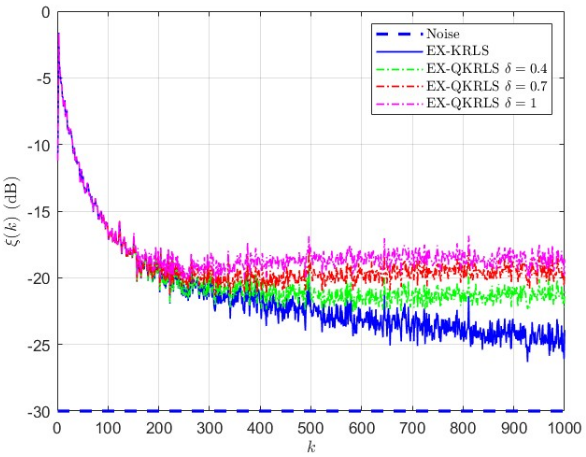

Fig. 1. Learning curves for algorithms EX-KRLS and EX-QKRLS.

The RF WebLab consists of a computer that manages the flow of user requests, a NI PXIe-5646R VST vector signal generator and analyzer, a preamplifier with an approximate gain of 40 dB, a Cree CGH40006-TB GaN power amplifier with a nominal gain of 13 dB for 2 GHz, and a 30 dB attenuator.

The complex baseband signal is sent with normalized amplitude. The system then adds a 2.4 GHz carrier and configures the RMS power before amplification. The output is returned in the baseband domain.

Predistortion begins with the generation of a bitstream, which is modulated using QAM and multiplexed via OFDM. The resulting signal is then amplitude-normalized to meet system requirements before transmission. After amplification, the signal goes through the synchronization process followed by predistortion using an indirect learning architecture [22]. The resulting signal is then relayed to the amplification system, as illustrated in Figure 2.

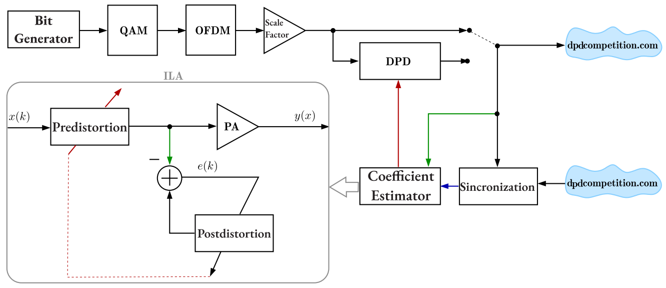

Fig. 2. DPD structure using RF WebLab features.

V. Experimental Results

For comparison purposes, tests were performed with the RLS with MP, RLS with CT, and EX-QKRLS algorithms. The experiment considered only a single predistortion loop, with the algorithm being implemented sample by sample.

For the test, a signal with 16-QAM modulation and an OFDM scheme was used, where the bandwidth is 40 MHz, with 1272 subcarriers spaced 30 kHz apart and a cyclic prefix of 144 samples. For this experiment, an oversampling factor of 3 was utilized. The RMS power defined for the input signal and delivered to the signal generator was -22 dBm.

First, DPD was performed with the RLS algorithm with MP and CT. For comparison purposes, all tests were performed with a forgetting factor of 0.99999 and, based on the study carried out in [23], a fifth-degree polynomial with a memory depth of 3 was used.

In the test with the EX-QKRLS algorithm, the parameters are as follows: $\sigma_k = 0.9$, $\delta = 0.3$, with a scale factor referring to the transition matrix of Equation (6) equal to 1, a regularization parameter equal to 0.01 and the ratio of the noise variances equal to 10^{-4}.

As a measure of performance, Figure 3 presents the signal spectra, highlighting the harmonic attenuation obtained with the application of pre-distortion for each algorithm tested. For reference, the spectra of the original signal and the amplified signal without DPD are also shown.

Figure 4 presents the constellation diagram visually showing the decrease in error vector magnitude (EVM) when using the EX-QKRLS algorithm. The EVM is mathematically defined as $\text{EVM} = \sqrt{\frac{\sum_{n=1}^{N} |s_n - \hat{s}_n|^2}{\sum_{n=1}^{N} |s_n|^2}}$, where s_n is the ideal symbol, $\hat{s}_n$ the received symbol, and N is the number of symbols. The EVM values of the tests of each algorithm can be seen in Table II.

Finally, Figure 5 depicts the PA normalized gain with and without the use of EX-QKRLS. This graph relates the input power to the resulting gain, considering a measured input saturation power of 28.45 dBm. The 1-dB compression point is clearly visible without DPD, while the predistorted case shows effective linearization of the PA response.

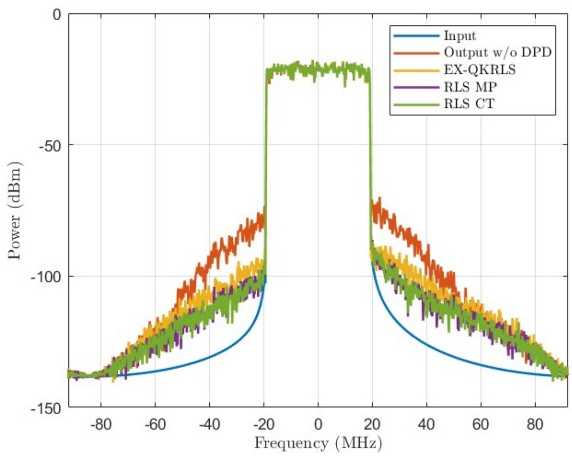

Fig. 3. Spectrum of input signal and amplified signals for each type of DPD.

TABLE II
EVM Comparison Table

	EVM w/o DPD	EVM w/ DPD	Difference
RLS MP	8.74%	6.02%	2.72
RLS CT	8.74%	5.93%	2.81
EX-QKRLS	8.74%	6.17%	2.57

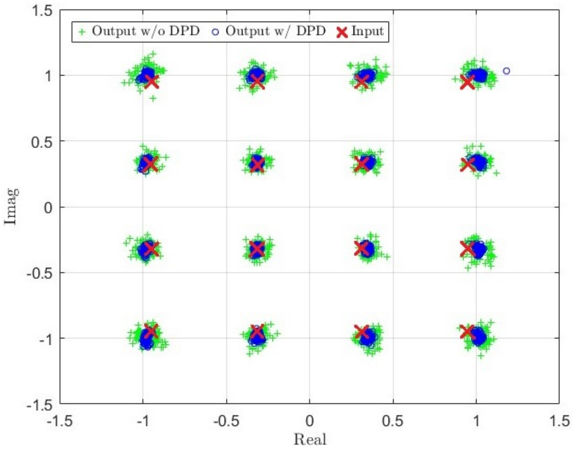

Fig. 4. Input signal constelation (in red), after amplification without DPD (green) and with predistortion using the EX-QKRLS Algorithm (blue).

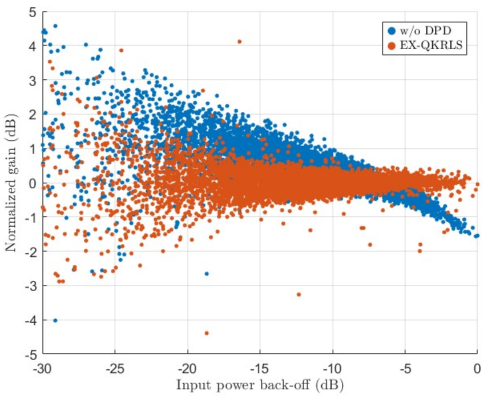

Fig. 5. Gains of amplified signals without DPD (blue) and with DPD (red).

VI. Conclusions

This work demonstrated that the EX-QKRLS algorithm presents similar performance to those of algorithms RLS MP and RLS CT in modeling the nonlinearity of the power amplifier, but without relying on a known polynomial structure. Such singularity makes the DPD technique a viable option for communication systems with an unknown PA behavior, ensuring efficient adaptation to different operating conditions.

Furthermore, the EX-QKRLS algorithm, by presenting this characteristic of adaptability to any type of nonlinearity, becomes an essential factor for systems such as software-defined radio and cognitive radio, where operability in different waveforms is required.

It is worth noting that kernel-based algorithms perform better in modeling the nonlinearity of power amplifiers that have little memory delay. This is the case for a signal from the RF WebLab [23], whose behavior is well captured by kernel-based models due to its low memory depth.

For future work, a new algorithm derived from EX-KRLS is under development, incorporating QR decomposition techniques using Householder reflectors. The goal is to improve the numerical stability of the EX-KRLS algorithm, making it more robust under ill-conditioned scenarios and more suitable for real-time digital predistortion implementations.

References

[1] A. S. Sedra, K. C. Smith and T. C. Carusone, *Circuitos Microeletrônicos (in Portuguese)*. 8th ed. Rio de Janeiro: LTC, 2023.

[2] A. A. M. Saleh, "Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers," in IEEE Transactions on Communications, vol. 29, no. 11, pp. 1715-1720, Nov. 1981.

[3] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt and J. Pastalan, "A Generalized Memory Polynomial Model for Digital Predistortion of RF Power Amplifiers", in IEEE Transactions on Signal Processing, vol. 54, no. 10, pp. 3852-3860, Oct. 2006.

[4] K. J. Muhonen, M. Kavehrad and R. Krishnamoorthy, "Look-up table techniques for adaptive digital predistortion: a development and comparison", in IEEE Transactions on Vehicular Technology, vol. 49, no. 5, pp. 1995-2002, Sep. 2000.

[5] X. Liu, J. Zhai, Z. Zhang, D. Ning, P. Chen and C. Yu, "High-Precision Digital Predistortion for Sub-6-GHz GaN Power Amplifiers Excited by a Signal With 400-MHz Modulation Bandwidth", in IEEE Microwave and Wireless Technology Letters, vol. 34, no. 5, pp. 552-555, May 2024.

[6] C. -F. Cheang, P. -I. Mak and R. P. Martins, "A Hardware-Efficient Feedback Polynomial Topology for DPD Linearization of Power Amplifiers: Theory and FPGA Validation", in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 9, pp. 2889-2902, Sep. 2018.

[7] S. Zheng et al., "Performing Volterra Series Based Indirect Learning Digital Pre-Distorter in Self-Homodyne Detection", in Journal of Lightwave Technology, vol. 42, no. 5, pp. 1453-1459, Mar. 2024.

[8] M. Schetzen. The Volterra and Wiener Theories of Nonlinear Systems. New York: Wiley, 1980.

[9] Gan, Li, and Emad Abd-Elrady. "Digital Predistortion of Memory Polynomial Systems using Direct and Indirect Learning Architectures". Proceedings of the Eleventh IASTED International Conference on Signal and Image Processing (SIP) (F. Cruz-Roldán and N. B. Smith, eds.), No. 654-802. Calgary, AB: ACTA Press, 2009.

[10] P. S. R. Diniz, *Adaptive Filtering: Algorithms and practical implementation*. 4. ed. New York: Springer, 2013.

[11] S. Haykin, *Adaptive Filter Theory*, 1st ed., New Jersey: Prentice Hall, 1986.

[12] J. A. Apolinário Jr., *QRD-RLS Adaptive Filtering*. New York: Springer, 2009.

[13] P. S. R. Diniz, M. L. R. Campos, W. A. Martins, M. V. S. Lima, and J. A. Apolinário Jr., *Online Learning and Adaptive Filters*. New York: Wiley, 2023.

[14] W.Liu, J. C. Príncipe and S. Haykin, *Kernel Adaptive Filtering: A comprehensive introduction*. 1st ed. New Jersey: Wiley, 2010.

[15] P. Bouboulis and S. Theodoridis, "Extension of Wirtinger's Calculus to Reproducing Kernel Hilbert Spaces and the Complex Kernel RLS," in IEEE Transactions on Signal Processing, vol. 59, no. 3, pp. 964-978, Mar. 2011.

[16] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", in J. Fluids Eng., vol. 82, no 1, pp. 35–45, 1960.

[17] W. Liu, I. Park, Y. Wang and J. C. Principe, "Extended Kernel Recursive Least Squares Algorithm", in IEEE Transactions on Signal Processing, vol. 57, no. 10, pp. 3801-3814, Oct. 2009.

[18] S. Haykin, A. H. Sayed, J. R. Zeidler, P. Yee and P. C. Wei, "Adaptive tracking of linear time-variant systems by extended RLS algorithms", in IEEE Transactions on Signal Processing, vol. 45, no. 5, pp. 1118-1128, May 1997.

[19] J. Platt, "A resource-allocating network for function interpolation". in Neural Computation - NECO, v. 3, p. 213–225, Jun. 1991.

[20] B. Chen, S. Zhao, P. Zhu and J. C. Príncipe, "Quantized Kernel Recursive Least Squares Algorithm", in IEEE Transactions on Neural Networks and Learning Systems, vol. 24, no. 9, pp. 1484-1491, Sep. 2013.

[21] RF WebLab. Available at: <https://dpdcompetition.com/rfweblab/>. Accessed on May 3, 2025.

[22] H. Paaso and Mammela, "A. Comparison of direct learning and indirect learning predistortion architectures," in 2008 IEEE International Symposium on Wireless Communication Systems. [S.l.: s.n.], 2008. p. 309–313.

[23] M. H. Moghaddam, S. R. Aghdam, N. Mazzali, and T. Eriksson, "Statistical Modeling and Analysis of Power Amplifier Nonlinearities em Communication Systems," in IEEE Transactions on Communications, vol. 70, no. 2, pp. 822-835, Feb. 2022.