Machine Learning Based System Identification with Binary Output Data Using Kernel Methods

Rachid Fateh[1], Hicham Oualla[2], Es-said Azougaghe[1], Anouar Darif[1], Ahmed Boumezzough[1], Said Safi[1], Mathieu Pouliquen[3], and Miloud Frikel[3]

[1]*Sultan Moulay Slimane University, Beni Mellal, Morocco,*
[2]*Akkodis, Paris, France,*
[3]*Normandie University, Caen, France*

https://doi.org/10.26636/jtit.2024.1.1430

Abstract — Within the realm of machine learning, kernel methods stand out as a prominent class of algorithms with widespread applications, including but not limited to classification, regression, and identification tasks. Our paper addresses the challenging problem of identifying the finite impulse response (FIR) of single-input single-output nonlinear systems under the influence of perturbations and binary-valued measurements. To overcome this challenge, we exploit two algorithms that leverage the framework of reproducing kernel Hilbert spaces (RKHS) to accurately identify the impulse response of the Proakis C channel. Additionally, we introduce the application of these kernel methods for estimating binary output data of nonlinear systems. We showcase the effectiveness of kernel adaptive filters in identifying nonlinear systems with binary output measurements, as demonstrated through the experimental results presented in this study.

Keywords — finite impulse response, kernel adaptive filtering, nonlinear systems identification, Proakis C channel

1. Introduction

Linear adaptive filters represent a distinct category of digital filters known for their capacity to adapt their parameters based on input data. These filters are widely employed in signal processing applications, including tasks such as noise reduction, echo cancellation, and equalization [1]–[4]. The fundamental concept behind linear adaptive filters revolves around employing an algorithm that continually updates the filter coefficients in response to variations in the input signal [5], [6].

The widely adopted algorithm for this purpose is the least mean squares (LMS) algorithm [7], which iteratively refines the filter coefficients to minimize the mean squared error between the filter's output and the desired output. The performance of linear adaptive filters hinges on various factors, such as the selection of the filter structure, the choice of the algorithm for coefficient updates, and the design of the input signal. In general, these filters demonstrate optimal performance when the input signal remains relatively stationary or changes gradually over time. Additionally, selecting a filter structure that aligns with the statistical properties of the signal further enhances its effectiveness.

Linear adaptive filters are widely used in various fields, such as communications, control systems, and biomedical signal processing, due to their versatility and effectiveness in handling complex signals. They are often used in conjunction with other signal processing techniques, such as Fourier analysis and wavelet transforms, to enable more sophisticated signal processing approaches [1]. For instance, in the field of biomedical signal processing, adaptive filtering has been utilized to remove noise and artifacts from electroencephalogram (EEG) signals [8]. In communication systems, adaptive filters have been applied to mitigate channel impairments and improve signal quality. In control systems, adaptive filtering has been used to identify and estimate system parameters, and to compensate for time-varying disturbances [9].

Machine learning (ML) algorithms, falling within the category of kernel methods, find extensive application across various tasks, such as classification, regression, channel identification, and more [10], [11]. These methods function by mapping data into a higher-dimensional space, allowing for more effective separation and analysis, without the need to explicitly calculate the coordinates in that space [12]–[15]. These methods are based on the concept that a decision boundary in the reproducing kernel Hilbert space (RKHS) [12] can be represented as a linear boundary in a lower-dimensional space, making it possible to capture complex, non-linear relationships between input features and the output variable [16]. Support vector machines (SVMs) are one of the most widely used kernel methods, especially for classification tasks [17]. Kernel ridge regression, on the other hand, is often used for regression tasks due to its ability to capture non-linear relationships between variables. Additionally, kernel principal component analysis (PCA) is used for data analysis, allowing for non-linear feature extraction and dimensionality reduction [18].

In addition to their performance, kernel methods are also highly interpretable, allowing users to understand how the algorithm is making predictions and adjust the model accordingly. For example, SVMs can be visualized by plotting

the decision boundary in the input feature space, which can provide insight into the characteristics of the data and the model [19].

Recently, kernel methods have been successfully applied in channel identification tasks, particularly in the context of blind channel identification, where the channel parameters are estimated without any prior knowledge of the channel. Kernel-based blind channel identification methods typically use a kernel function to map the received signal into a high-dimensional space, where the channel parameters can be estimated using linear regression techniques. The estimated parameters can then be used to equalize the received signal and improve the accuracy of the communication system [20]–[24].

At present, there are many adaptive kernel filtering algorithms that have been exploited for channel identification in wireless communication systems. Some of them are described below.

Kernel least mean squares (KLMS) is a kernel-based adaptive filter that can be used for channel identification in wireless communication systems [25]. In KLMS, a kernel function is used to map the input data into a higher-dimensional space, where the linear regression problem is easier to solve. The KLMS algorithm updates the filter coefficients based on the difference between the predicted output and the actual output. The Gaussian kernel is a popular choice for this purpose.

Kernel normalized least mean squares (KNLMS) is a variant of KLMS that includes a normalization factor in the update rule. This helps prevent the filter coefficients from becoming too large and unstable [26]. KNLMS can be used for channel identification in wireless communication systems, and it has been proven to be effective in reducing computational complexity of KLMS.

Kernel extended improved proportionate NLMS (KEIPNLMS) [27] is an algorithm that employs a radial basis function (RBF) kernel to perform an implicit mapping of the data using the kernel trick to estimate the impulse response parameters for single-input single-output (SISO) nonlinear system identification.

In this paper, we investigate a non-linear system identification problem in the presence of noise. Section 2 provides a detailed description of the problem. In Section 3, we introduce fundamental notations of kernel methods, followed by a discussion of four algorithms: LMS, NLMS, KLMS, and KNLMS. We then evaluate the effectiveness of kernel methods using binary-valued output by analyzing simulation results in Section 4. Our findings are summarized in Section 5 which concludes the paper.

2. System Descriptions

In this section, we introduce some notations and assumptions that will be used throughout the paper.

The Hammerstein system, a distinctive nonlinear model, is frequently employed in the realm of system identification. System identification aims to create precise mathematical models that mirror the behavior of real-world systems using observed input-output data. In this context, we focus on the Hammerstein system depicted in Fig. 1. This system comprises a nonlinear static function followed by a FIR filter with a known order. This structure is chosen for its ability to effectively represent both nonlinear and linear dynamics in a system, offering sufficient flexibility and interpretability during the identification process.

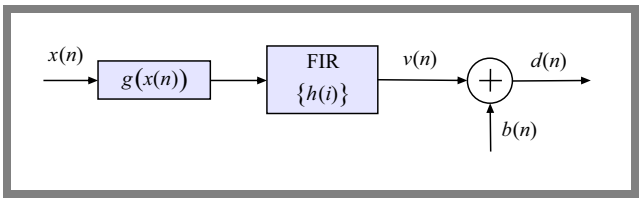

Fig. 1. Block diagram of the Hammerstein system.

As shown in Fig. 1, the desired system output can be obtained using the following expression:

$$\begin{cases} v(n) = \sum\limits_{i=0}^{L-1} h(i)\, g\big(x(n-i)\big) \\ d(n) = v(n) + b(n), \quad n = 0, 1, 2, \ldots, N \end{cases}, \tag{1}$$

where $x(k)$ is the input signal, $h(i), i = 0, 1, \ldots, L-1$ represents the channel impulse response, L refers to the FIR system order, $g(.)$ denotes the nonlinearity and $b(k)$ is the measurement noise.

The Hammerstein system was adopted under the following fundamental assumptions:

- the input sequence $x(n)$ is an independent and identically distributed (i.i.d.) bounded random process characterized by a zero mean,
- the additive noise, represented as $b(n)$, is proposed to be Gaussian and independent of both $x(n)$ and $d(n)$ (both are bounded),
- function $g(.)$ is both invertible and continuous for any finite value of x.

The hypotheses listed above are formulated to simplify the system analysis process and to achieve the best results in terms of mean square error. The primary objective of this paper is to present a comparison of the kernel methods that have been proposed in the literature for identifying the output d generated by Eq. (1).

3. Kernel Methods

Here, we introduce kernel methods, a category of techniques that empower us to extend traditional linear algorithms to handle non-linear data. The fundamental concept underlying kernel methods is the application of linear algorithms to a transformed representation of the data within a higher-dimensional space. This transformation facilitates the separation of data points into classes that were not linearly separable in the original space.

Kernel methods constitute a category of ML algorithms that leverage kernel functions to map input data into a higher-dimensional space. They find primary application in tackling classification and regression tasks.

Definition 1. A function $\kappa : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$ is a similarity measure if the following conditions are satisfied:

- $x, y \in \mathcal{X}\ \kappa(x, y) \geqslant 0$,
- $x, y \in \mathcal{X}\ \kappa(x, y) = \kappa(y, x)$,
- $\forall y \in \mathcal{X}, y \neq x\ \kappa(x, y) > \kappa(x, x)$,
- $\kappa(x, y) = \kappa(x, x) \Leftrightarrow x = y$.

3.1. Positive Definite Kernel

Theorem 1. Let $\mathcal{X}$ be a compact in $\mathbb{R}$ (compact = closed and bounded) and $K : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$ a symmetric function. We also assume that $\forall f \in \mathcal{L}_2(\mathcal{X})$:

$$\int_{\mathcal{X}} K(x, y) f(x) f(y) dx dy \geqslant 0 \quad \text{(Mercer condition)} . \tag{2}$$

Then there exists a Hilbert space $\mathcal{H}$ and $\Phi : \mathcal{X} \longrightarrow \mathcal{H}$ such that $\forall (x, y) \in \mathcal{X}^2$:

$$K(x, y) = \big\langle \Phi(x), \Phi(y) \big\rangle . \tag{3}$$

The function $K(x, y)$ is called positive definite kernel.

An equivalent condition for the function $K : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$ to be a definite positive kernel is the following:

- $\forall n \in \mathbb{N}$ and $\{x_i\}\ i = 1, \ldots, n \subset \mathcal{X}$ the Gramm matrix

$$K = [K_{i,j}]\ i = 1, \ldots, n = \big[K(x_i, x_j)\big] i = 1, \ldots, n \tag{4}$$

is positive definite, that is:

$$\forall c \in \mathbb{R}^n, \quad c \neq 0, \quad \text{we have} \quad c^\top K c > 0 . \tag{5}$$

Therefore, a valid kernel ensures the existence of $\mathcal{H}$ and can be expressed as a scalar product in Hilbert space $\mathcal{H}$. A good kernel also guarantees the convexity of the quadratic optimization problem under inequality constraints encountered for SVM.

3.2. Conditionally Positive Definite Kernel

A kernel is conditionally positive definite (CPD) if $\forall n \in \mathbb{N}$ and $\{x_i\}\ i = 1, \ldots, n \subset \mathcal{X}$ the Gramm matrix:

$$K = \big[K_{i,j}\big]\ i = 1, \ldots, n = \big[K(x_i, x_j)\big]\ i = 1, \ldots, n \tag{6}$$

is conditionally positive definite, i.e.

$$\forall c \in \mathbb{R}^n, \quad c \neq 0 \ \text{ such as } \sum_{i=1}^{n} c_i = 0, \ \text{we have } \ c^\top K c > 0. \tag{7}$$

This definition extends the class of kernel functions for which the SVM optimization problem is guaranteed to be convex. Given a positive conditionally defined symmetric kernel, there exists:

- a vector space $\mathcal{V}$,
- a transformation $\Phi : \mathcal{X} \longrightarrow \mathcal{V}$,
- a bilinear form $Q : \mathcal{V} \times \mathcal{V} \longmapsto \mathbb{R}$

such as:

$$K(x, y) = Q\big(\Phi(x), \Phi(y)\big) , \tag{8}$$

- if K is not defined positive, then Q is not a scalar product.

3.3. Construction of Positive Definite Kernels

There are several approaches to obtain kernel functions.

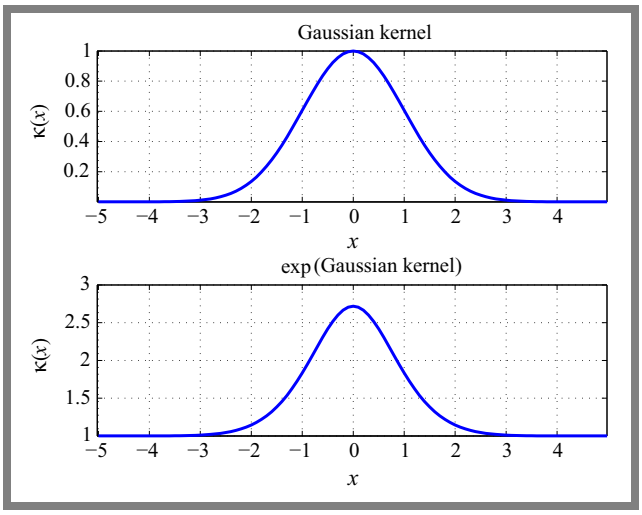

Fig. 2. The exponential of a kernel is a kernel.

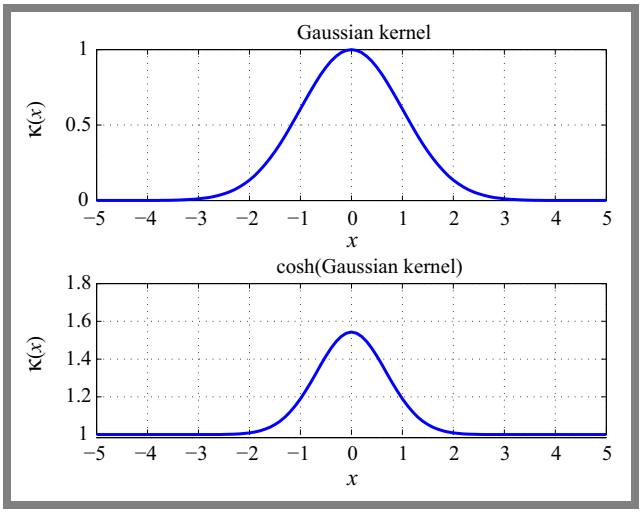

Fig. 3. The cosh of a kernel is a kernel.

- Direct construction (using the Φ projection):
  Direct definition of $\mathcal{H}$, $\Phi : \mathcal{X} \longmapsto \mathcal{H}$ and then construction of the kernel:

$$K : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R} \ \text{ by } \ K(x, y) = \big\langle \Phi(x), \Phi(y) \big\rangle \tag{9}$$

  Example 1. Let $\mathcal{X}$ be a compact in $\mathbb{R}$. We consider $\Phi : \mathcal{X} \longmapsto \mathbb{R}$, then $K : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$ defined by:

$$K(x, y) = \Phi(x) \cdot \Phi(y) \tag{10}$$

  is a positive definite kernel.
  Note that these conforming kernels cannot be interpreted as similarities.
  Particular cases:
  - $f : \mathbb{R} \longmapsto \mathbb{R}, \Phi(x) = x : K(x, y) = x \cdot y$,
  - $f : \mathbb{R} \longmapsto \mathbb{R}, \Phi(x) = e^x : K(x, y) = e^{x+y}$.

– Transformation of existing kernels:
  1) If $K : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$ is positive definite, then the definition of $\exp(K)$ is also positive definite.
  2) If $K : \mathcal{X} \times \mathcal{X} \longmapsto [-1, 1]$ is positive definite then $\cosh(K)$ is positive definite too.

- Combination of existing kernels:
  If $K_1, K_2 : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$ are positive definite and $\alpha_1, \alpha_2 > 0$ then the following kernels are also positive definite:
  - linear combination: $K(x,y) = \alpha_1 K_1(x,y) + \alpha_2 K_2(x,y)$,
  - simple product: $K(x,y) = \alpha_1 K_1(x,y) \cdot \alpha_2 K_2(x,y)$.

  Obviously, K is defined on $\mathcal{X} \times \mathcal{X}$ with values in $\mathbb{R}$.
  If $K_1 : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$ and $K_2 : \mathcal{X} \times \mathcal{X} \longmapsto \mathbb{R}$ are positive definite then are also positive definite:
  - direct sum : $K_1 \bigoplus K_2 = K_1 + K_2$,
  - tensor product : $K_1 \bigotimes K_2 = K_1.K_2$.

3.4. Examples of Kernels

In this subsection, we aim to illustrate several examples of kernels that are widely used across diverse applications. Kernels play a crucial role in various machine learning and statistical techniques, contributing to their flexibility and effectiveness.

1) Linear kernel:
   - **Definition:** $K(x_i, x_j) = x_i^\top x_j = \langle x_i, x_j \rangle$.
   - **Explanation:** A linear kernel computes the inner product between input vectors, providing a measure of similarity based on their alignment. It serves as a foundational choice, particularly in scenarios where the underlying relationships are expected to be linear.
2) Polynomial kernel:
   - **Definition:** $K(x_i, x_j) = \big(\langle x_i, x_j \rangle + c\big)^p$.
   - **Explanation:** A polynomial kernel introduces non-linearity by raising the dot product to a certain power p, with an optional constant term c. This kernel is effective in capturing higher-order relationships in the data.
3) Gaussian radial basis function (RBF) kernel:
   - **Definition:** $K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2)$.
   - **Explanation:** An RBF kernel measures similarity based on the Euclidean distance between vectors. It is widely employed for its ability to capture complex, non-linear relationships and is a key component in support vector machines (SVMs).
4) Sigmoid kernel:
   - **Definition:** $K(x_i, x_j) = \text{tgh}\big(a\langle x_i, x_i \rangle + c\big)$.
   - **Explanation:** A sigmoid kernel is useful for capturing relationships characterized by sigmoidal shapes. It is employed in neural networks and logistic regression, providing flexibility in modeling.

3.5. LMS Algorithm

The LMS algorithm is a specific type of an adaptive filter used in digital signal processing and machine learning applications. It is widely used to tackle tasks such as system estimation, channel radio equalization, noise cancellation, and adaptive beamforming.

The basic idea behind the LMS algorithm is to adjust the weights of a linear filter iteratively, based on the difference between the predicted output of the filter and the actual output. The algorithm uses a measure of the error between the predicted and actual outputs, called the "error signal", to update the filter weights in the direction that minimizes the error. The LMS weight update recursion is [7]:

$$\theta(n+1) = \theta(n) + \mu\, e(n)\, x(n)\ , \tag{11}$$

where μ is the step size or learning rate, and $e(n)$ is the error at time n given by:

$$e(n) = d(n) - \theta(n)^\top x(n)\ , \tag{12}$$

where $d(n)$ is the desired output at time n.

3.6. NLMS Algorithm

The NLMS algorithm is a variation of the LMS algorithm that improves its performance by normalizing the weight update step based on the power of the input signal. This makes the algorithm more robust to changes in the input signal power, and it can converge faster and more accurately than the standard LMS algorithm.
The basic idea behind the NLMS algorithm is similar to that of the LMS algorithm. It adjusts the weights of a linear filter iteratively based on the difference between the predicted output of the filter and the actual output. However, the weight update step in the NLMS algorithm is normalized based on the power of the input signal, which helps prevent the weight update from becoming too large or too small.
The formula for the weight update in the NLMS algorithm is [28]:

$$\theta(n+1) = \theta(n) + \frac{\mu}{\|x(n)\|^2} e(n)\, x(n)\ , \tag{13}$$

where $\theta(n)$ is the filter weights at iteration n, μ is the step-size parameter, $e(n)$ is the error signal at iteration n, $x(n)$ is the input signal at iteration n, and $\|x(n)\|^2$ is the power of the input signal.
The NLMS algorithm has several advantages over the LMS algorithm, including its faster convergence rate, better tracking of time-varying signals, and improved robustness to changes in the input signal power. However, it can be sensitive to noise and can still suffer from slow convergence or local minima if the step-size parameter is set too high.

3.7. KLMS Algorithm

Kernel least mean square (KLMS) is an online ML algorithm that is designed for non-linear regression problems. KLMS uses a kernel function to transform the input data into a high-dimensional feature space, where a linear relationship between inputs and outputs can be learned using a simple linear model. A description of the KLMS algorithm is given below [25]:

$$\begin{aligned} \theta(0) &= 0 \\ e(n) &= d(n) - \theta(n-1)^\top \Phi\big(x(n)\big) \\ \theta(n) &= \theta(n-1) + \mu\, e(n) \Phi\big(x(n)\big)\ . \end{aligned} \tag{14}$$

The KLMS algorithm works by transforming the input data into a high-dimensional feature space using a kernel function. The kernel function measures the similarity between two data points and maps them into a high-dimensional space, where they can be linearly separable. The output $\theta(n-1)^\top \Phi\big(x(n)\big)$ is then predicted using the inner product of the kernel function and the weight vector θ. The error $e(n)$ is computed as the difference between the predicted output $\theta(n-1)^\top \Phi\big(x(n)\big)$ and the actual output $d(n)$.
The KLMS algorithm updates the weight vector w using the error e and the kernel function κ to compute the update. The update rule follows the same principle as the LMS algorithm, where the weight vector is updated in the direction that reduces the error. The update rule includes the kernel function to account for the nonlinear relationship between inputs and outputs.

3.8. KNLMS Algorithm

The KNLMS algorithm is a variation of the NLMS algorithm that uses a kernel function to map the input signal to a higher-dimensional space, where it is easier to separate linearly. This makes the algorithm more powerful and versatile, with applications in nonlinear filtering and prediction.
The basic idea behind the KNLMS algorithm is to apply a nonlinear mapping to the input signal using a kernel function, such as a Gaussian or polynomial function. The mapped signal is then used as the input to the standard NLMS algorithm, which updates the filter weights based on the difference between the predicted output and the actual output. The weight update step is normalized based on the power of the mapped signal, similar to the NLMS algorithm.
The formula for the weight update in the KNLMS algorithm is [26]:

$$\theta(n+1) = \theta(n) + \frac{\mu}{\varepsilon + \left\|\Phi\big(x(n)\big)\right\|^2} e(n)\, \Phi\big(x(n)\big)\,, \quad (15)$$

where $\theta(n)$ is the filter weights at iteration n, μ is the step-size parameter, $e(n)$ is the error signal at iteration n, $\Phi\big(x(n)\big)$ is the mapped input signal at iteration n, ε refers to a small constant mobilized to avoid numerical problems, and $\|\Phi\big(x(n)\big)\|^2$ is the power of the mapped input signal.
The KNLMS algorithm has several advantages over the NLMS algorithm, including its ability to handle nonlinear signals and its improved performance in high-dimensional spaces. However, it can be computationally expensive and may require careful selection of the kernel function and its parameters to achieve good performance.

4. Simulation Results and Discussion

To validate the efficacy of the presented algorithms in the presence of Gaussian additive noise, simulations were performed, focusing on nonlinear system identification with binary-valued output, utilizing the Gaussian kernel.

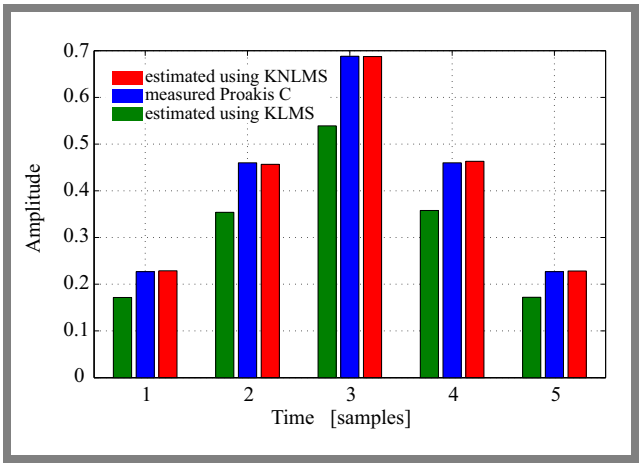

Fig. 4. Estimation of the Proakis C channel impulse response, for a data length of $N = 1000$ and $SNR = 20$ dB.

$$\kappa(x,y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right), \quad \forall (x,y) \in \mathcal{X}^2\,, \quad (16)$$

where $\sigma > 0$ represents the smoothing parameter.
The simulations involve passing a signal $x(n)$ from a normal distribution with mean 0 and variance 1 through a Hammerstein system. This system consists of a nonlinearity $\mathrm{tgh}(x)$ followed by a linear finite impulse response (FIR) channel. The linear channel uses an impulse response h of length $L = 5$ known as the Proakis C channel, with coefficients of $[0.227, 0.460, 0.688, 0.460, 0.227]$. Additionally, Gaussian white noise with a power of 20 dB is added to the channel output during each of the 1024 iterations. Finally, with the aid of a binary detector $I[.]$ that employs a predefined threshold $C \in \mathbb{R}$, the system's output $d(k)$ becomes measurable. The quantized output data $s(k)$ can be represented mathematically as follows:

$$s(k) = I_{[d(n)\geqslant C]} = \begin{cases} 1 & \text{if } d(n) \geqslant C \\ -1 & \text{otherwise}\,. \end{cases} \quad (17)$$

4.1. Proakis C Channel Identification

Figure 4 shows the impulse response parameters of the Proakis C channel estimated using both algorithms (KLMS and KNLMS). The estimates were obtained for $SNR = 15$ dB and $N = 1024$ input signal samples, using 50 Monte Carlo iterations. The results indicate that the kernel normalized least mean squares (KNLMS) algorithm accurately estimates the response parameters, while the KLMS algorithm produces estimated values that differ significantly from the measured values. To assess the frequency domain performance of both algorithms, we visualize the estimated amplitude and phase response of the Proakis C channel's impulse response for a sample size of $N = 1000$ and $SNR = 20$ dB. Figure 5 highlights the estimates of the amplitude and phase of the Proakis C channel, using the KLMS and KNLMS algorithms. Based on these results, the KNLMS algorithm proves to be more effective than the KLMS algorithm, as it allows to ob-

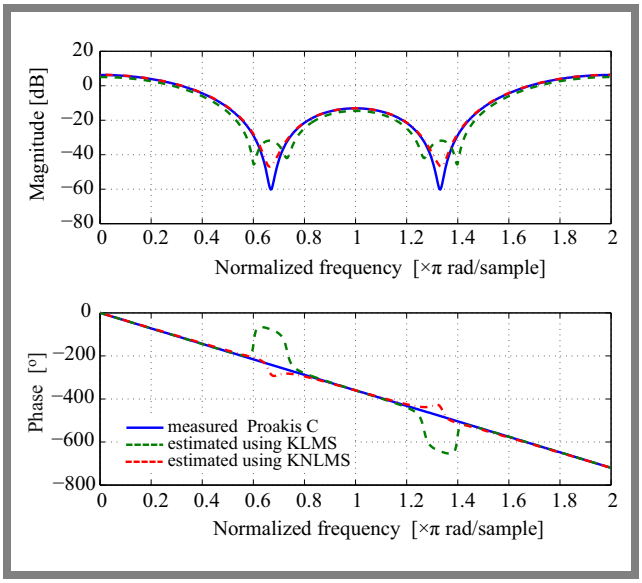

Fig. 5. Estimation of the Proakis C amplitude for a data length of $N = 3000$ and $SNR = 20$ dB.

tain the same shapes of the estimated amplitude and phase values as those measured.

4.2. Output Data Estimation

In Figs. 6 and 7, the estimation of the output $d(k)$ for nonlinear system identification without binary-valued output observations is demonstrated using KLMS and KNLMS. The lower graphs depict the complete signal form for a data length of $N = 1000$, while the upper graphs focus on data lengths between 590 and 640 to provide a more detailed view of the processed signals.

It should be noted that with the KNLMS algorithm, the estimated output $d(k)$ follows the true model in perfect agreement with the measured data (Fig. 7). In comparison, with the KLMS algorithm, we observe that the estimated output $d(k)$ follows the variations of the real output with some fluctuations. The performance of the KLMS algorithm degrades

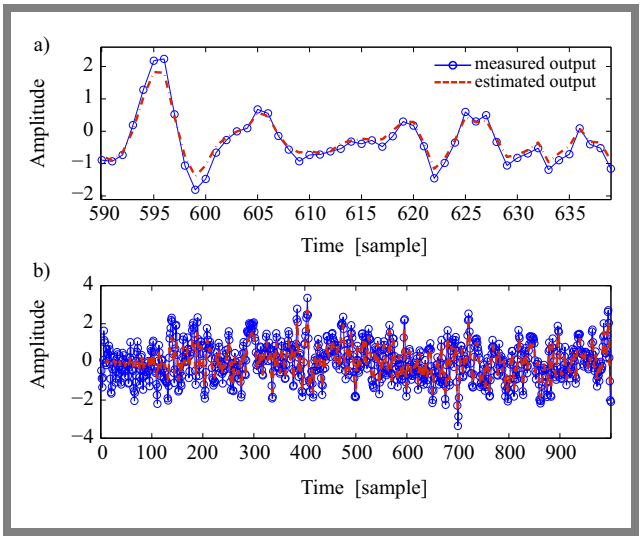

Fig. 6. Output $d(k)$ estimation using KLMS algorithm: a) zoomed-in between 590–640 samples, b) full 1000 samples.

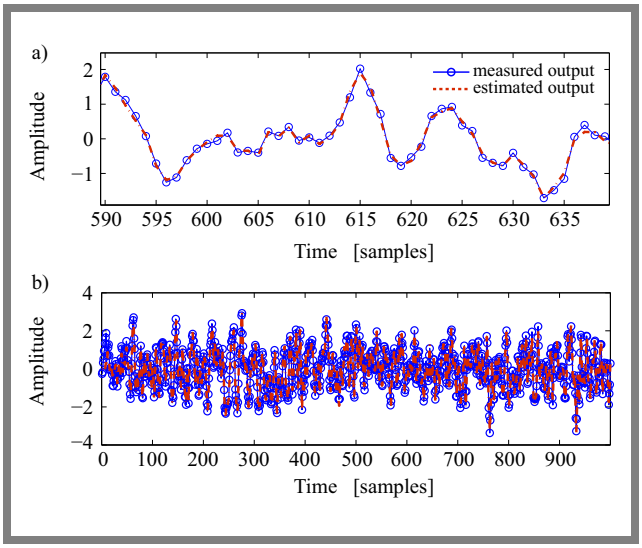

Fig. 7. Output $d(k)$ estimation using KNLMS algorithm: a) zoomed-in between 590–640 samples, b) full 1000 samples.

when estimating the output for small sample sizes $N < 100$, and we have a significant difference between the shape of the estimated and measured output (Fig. 6).

To illustrate the performance of adaptive kernel filter algorithms (KLMS and KNLMS) based on binary data output, the identification is applied to Hammerstein models with different complexities and binary output.

The estimation of the binary output $s(k)$ as a function of the number of samples is presented in Figs. 8 and 9, using KNLMS and KLMS for a SNR of 20 dB. It appears that the estimated binary output takes the same form as the measured output data. The estimation of the binary output is done with high accuracy using the KNLMS algorithm. In the case of the KLMS algorithm, we observe a difference in some samples (Fig. 8).

The KNLMS algorithm is often considered more effective than the KLMS algorithm for system identification because it can produce more accurate estimates of the output data.

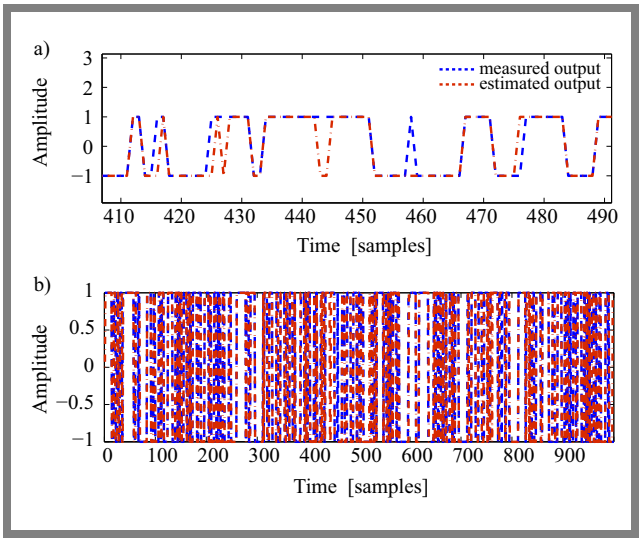

Fig. 8. Binary output $s(k)$ estimation using KLMS algorithm: a) zoomed-in between 410–590 samples, b) full 1000 samples.

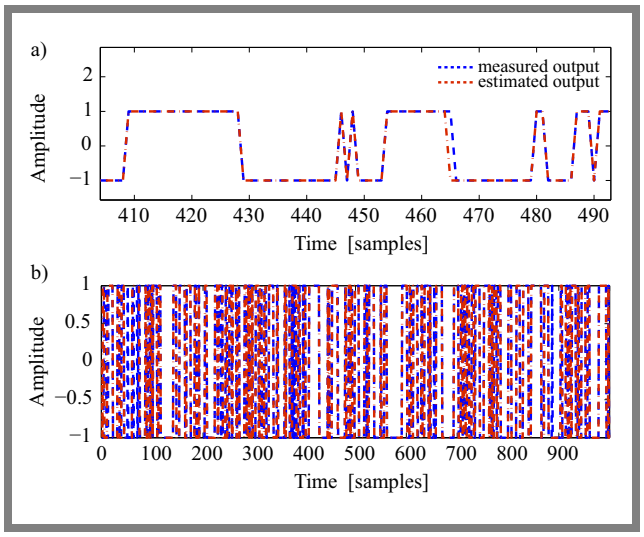

Fig. 9. Binary output $s(k)$ estimation using KNLMS algorithm: a) zoomed-in between 410–590 samples, b) full 1000 samples.

The main reason for this is that the KNLMS algorithm includes a normalization step that improves the stability and convergence of the algorithm. This is particularly important in noisy environments, where the impact of noise on the estimated impulse response can be reduced, leading to improved accuracy in the channel identification process.
Based on the results obtained, it can be concluded that kernel adaptive filters are effective in identifying nonlinear systems with binary output measurements. Nonlinear systems with binary output can offer various advantages. Firstly, binary output is easier to process and interpret compared to continuous output due to its simplicity, which results in lower computational and storage requirements. Secondly, binary output is more robust to noise and interference than continuous output. Binary signals are less affected by small variations or fluctuations in the input signal, thus enhancing the accuracy and reliability of the system.

5. Conclusion

In our research paper, we delved into the intricate task of identifying nonlinear systems characterized by binary output data, and we addressed this challenge through the implementation of adaptive kernel filtering algorithms. Our specific application honed in on the estimation of parameters associated with the Proakis channel, a scenario with inherent complexities. The Proakis channel is known for its practical relevance in communication systems, and accurate estimation of its parameters is crucial for optimizing system performance.
In the course of our investigations, we focused on the Hammerstein system identification problem, aiming to discern the most effective algorithm for binary output data estimation and channel impulse response parameter estimation. Notably, simulations unveiled compelling results, showcasing that, in this context, the KNLMS algorithm exhibited superior performance when compared to the KLMS algorithm.
The KNLMS algorithm's proficiency in handling the unique challenges posed by binary output data and accurately estimating the channel impulse response parameters in the Hammerstein system underscores its efficacy in scenarios where nonlinearities and binary responses are prevalent. Our findings contribute valuable insights to the field of system identification, especially in applications where binary output data and the Hammerstein system model are prominent, offering researchers and practitioners a promising tool for enhancing the accuracy and reliability of their models. These results pave the way for further exploration and refinement of adaptive kernel filtering algorithms in the realm of nonlinear system identification.
Moving forward, our primary emphasis will be on enhancing the kernel algorithm to enable the identification of measurable frequency-selective fading radio channels.

References

[1] S. Haykin, *Adaptive Filter Theory*, 4th ed., Hoboken: Prentice Hall, 920 p., 2002 (ISBN: 9780130901262).
[2] M.M. Sondhi, "The History of Echo Cancellation", *IEEE Signal Processing Magazine*, vol. 23, no. 5, pp. 95–102, 2006 (https//doi.org/10.1109/MSP.2006.1708416).
[3] A.H. Sayed, *Fundamentals of Adaptive Filtering*, Hoboken: John Wiley & Sons, 1168 p., 2003 (ISBN: 9780471461265).
[4] P.S.R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementations*, 3rd ed., New York: Springer, 652 p., 2008 (https://doi.org/10.1007/978-1-4614-4106-9).
[5] R. Fateh, A. Darif, and S. Safi, "Kernel and Linear Adaptive Methods for the BRAN Channels Identification", in: *International Conference on Advanced Intelligent Systems for Sustainable Development*, Tangier, Morocco, pp. 579–591, 2020 (https://doi.org/10.1007/978-3-030-90639-9_47).
[6] R. Fateh, A. Darif, and S. Safi, "Identification of the Linear Dynamic Parts of Wiener Model Using Kernel and Linear Adaptive", in: *International Conference on Advanced Intelligent Systems for Sustainable Development*, Tangier, Morocco, pp. 387–400, 2020 (https://doi.org/10.1007/978-3-030-90639-9_31).
[7] E. Ferrara, "Fast Implementations of LMS Adaptive Filters", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 474–475, 1980 (https://doi.org/10.1109/tassp.1980.1163432).
[8] S. Sanei and J.A. Chambers, *EEG Signal Processing*, Hoboken: John Wiley & Sons, 289 p., 2013 (https://doi.org/10.1002/9780470511923).
[9] M. Krstic, I. Kanellakopoulos, and P.V. Kokotovic, *Nonlinear and Adaptive Control Design*, Hoboken: John Wiley & Sons, 592 p., 1995 (ISBN: 9780471127321).
[10] R. Fateh, A. Darif, and S. Safi, "Channel Identification of Non-linear Systems with Binary-Valued Output Observations Based on Positive Definite Kernels", *E3S Web of Conferences*, vol. 297, art. no. 01020, 2021 (https://doi.org/10.1051/e3sconf/202129701020).
[11] R. Fateh, A. Darif, and S. Safi, "Hyperbolic Functions Impact Evaluation on Channel Identification Based on Recursive Kernel Algorithm", *2022 8th International Conference on Optimization and Applications (ICOA)*, Genoa, Italy, 2022 (https://doi.org/10.1109/ICOA55659.2022.9934118).
[12] J.W. Xu, A.R. Paiva, I. Park, and J.C. Principe, "A Reproducing Kernel Hilbert Space Framework for Information-theoretic Learning", *IEEE Transactions on Signal Processing*, vol. 56, no. 12, pp. 5891–5902, 2008 (https://doi.org/10.1109/TSP.2008.2005085).

[13] W. Liu, J.C. Principe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, Hoboken: John Wiley & Sons, 240 p., 2010 (ISBN: 9780470447536).

[14] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 462 p., 2004 (https://doi.org/10.1017/CBO9780511809682).

[15] N. Aronszajn, "Theory of Reproducing Kernels", *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950 (https://doi.org/10.2307/1990404).

[16] B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Cambridge: MIT Press, 2002 (https://doi.org/10.7551/mitpress/4175.001.0001).

[17] C. Cortes and V. Vapnik, "Support-vector Networks", *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995 (https://doi.org/10.1007/bf00994018).

[18] B. Scholkopf, A. Smola, and K.R. Muller, "Kernel Principal Component Analysis", in: *International Conference on Artificial Neural Networks (ICANN)*, Lausanne, Switzerland, pp. 583–588, 1997 (https://doi.org/10.1007/bfb0020217).

[19] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support Vector Machines", *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 4, pp. 18–28, 1998 (https://doi.org/10.1109/5254.708428).

[20] M. Zidane and R. Dinis, "A New Combination of Adaptive Channel Estimation Methods and TORC Equalizer in MC-CDMA Systems", *International Journal of Communication Systems*, vol. 33, no. 11, art. no. 4429, 2020 (https://doi.org/10.1002/dac.4429).

[21] M. Zidane, S. Safi, and M. Sabri, "Measured and Estimated Data of Non-linear BRAN Channels Using HOS in 4G Wireless Communications", *Data in Brief*, vol. 17, pp. 1136–1148, 2018 (https://doi.org/10.1016/j.dib.2018.02.005).

[22] R. Fateh, A. Darif, and S. Safi, "Performance Evaluation of MC-CDMA Systems with Single User Detection Technique using Kernel and Linear Adaptive Method", *Journal of Telecommunications and Information Technology*, no. 4, pp. 1–11, 2021 (https://doi.org/10.26636/jtit.2021.151621).

[23] M. Zidane, S. Safi, and M. Sabri, "Compensation of Fading Channels Using Partial Combining Equalizer in MC-CDMA Systems", *Journal of Telecommunications and Information Technology*, no. 1, pp. 5–11, 2017 (http://dlibra.itl.waw.pl/dlibra-webapp/Content/1962/ISSN_1509-4553_1_2017_5.pdf).

[24] S. Safi, M. Frikel, A. Zeroual, and M. M'Saad, "Higher Order Cumulants for Identification and Equalization of Multicarrier Spreading Spectrum Systems", *Journal of Telecommunications and Information Technology*, no. 1, pp. 74–84, 2011 (https://doi.org/10.26636/jtit.2022.161122).

[25] W. Liu, P.P. Pokharel, and J.C. Principe, "The Kernel Least-mean-square Algorithm", *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 543–554, 2008 (https://doi.org/10.1109/tsp.2007.907881).

[26] C. Richard, J. Bermudez, and P. Honeine, "Online Prediction of Time Series Data with Kernels", *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2009 (https://doi.org/10.1109/TSP.2008.2009895).

[27] R. Fateh, A. Darif, and S. Safi, "An Extended Version of the Proportional Adaptive Algorithm Based on Kernel Methods for Channel Identification with Binary Measurements", *Journal of Telecommunications and Information Technology*, no. 3, pp. 47–58, 2022 (https://doi.org/10.26636/jtit.2022.161122).

[28] S. Ciochina, C. Paleologu, and J. Benesty, "An Optimized NLMS Algorithm for System Identification", *Signal Processing*, vol. 118, pp. 115–121, 2016 (https://doi.org/10.1016/j.sigpro.2015.06.016).

[29] L. Ljung, *System Identification: Theory for the User*, 2nd ed., Hoboken: Prentice Hall, 640 p., 1999 (ISBN: 9780136566953).

[30] F. Ding, *System Identification-Performances Analysis for Identification Methods*, Beijing: Science Press, 2014.

[31] X. Zhang, F. Ding, F.E. Alsaadi, and T. Hayat, "Recursive Parameter Identification of the Dynamical Models for Bilinear State Space Systems", *Nonlinear Dynamics*, vol. 89, pp. 2415–2429, 2017 (https://doi.org/10.1007/s11071-017-3594-y).

[32] L. Xu, "Application of the Newton Iteration Algorithm to the Parameter Estimation for Dynamical Systems", *Journal of Computational and Applied Mathematics*, vol. 288, pp. 33–43, 2015 (https://doi.org/10.1016/j.cam.2015.03.057).

[33] Q. Song, "Recursive Identification of Systems with Binary-valued Outputs and with ARMA Noises", *Automatica*, vol. 93, pp.106–113, 2018 (https://doi.org/10.1016/j.automatica.2018.03.059).

[34] J. Guo, X. Wang, W. Xue, and Y. Zhao, "System Identification with Binary-valued Observations under Data Tampering Attacks", *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3825–3832, 2020 (https://doi.org/10.1109/TAC.2020.3029325).

[35] L. Li, F. Wang, H. Zhang, and X. Ren, "A Novel Recursive Learning Estimation Algorithm of Wiener Systems with Quantized Observations", *ISA Transactions*, vol. 112, pp. 23–34, 2021 (https://doi.org/10.1016/j.isatra.2020.11.032).

[36] L. Zhang, Y. Zhao, and L. Guo, "Identification and Adaptation with Binary-valued Observations under Non-persistent Excitation Condition", *Automatica*, vol. 138, art. no. 110158, 2022 (https://doi.org/10.1016/j.automatica.2022.110158).

[37] R. Fateh and A. Darif, "Mean Square Convergence of Reproducing Kernel for Channel Identification: Application to Bran D Channel Impulse Response", in: *International Conference on Business Intelligence*, Beni-Mellal, Morocco, 2021 (https://doi.org/10.1007/978-3-030-76508-8_20).

[38] W. Liu and J.C. Principe, "Kernel Affine Projection Algorithms", *EURASIP Journal on Advances in Signal Processing*, vol. 2008, art. no. 784292, 2008 (https://doi.org/10.1155/2008/784292).

Rachid Fateh, Ph.D.
Laboratory of Innovation in Mathematics, Applications, and Information Technologies, Polydisciplinary Faculty
https://orcid.org/0000-0002-0574-2105
E-mail: fateh.smi@gmail.com
Sultan Moulay Slimane University, Beni Mellal, Morocco
https://www.usms.ac.ma

Hicham Oualla, Ph.D.
E-mail: hicham.oualla@akkodis.com
Akkodis, Paris, France
http://www.akkodis.com

Es-said Azougaghe, Ph.D.
Laboratory of Information Processing and Decision Support
https://orcid.org/0000-0002-0233-1132
E-mail: essaidazougaghe@gmail.com
Sultan Moulay Slimane University, Beni Mellal, Morocco
https://www.usms.ac.ma

Anouar Darif, Ph.D.
Laboratory of Innovation in Mathematics, Applications, and Information Technologies, Polydisciplinary Faculty
https://orcid.org/0000-0001-8026-9189
E-mail: anouar.darif@gmail.com
Sultan Moulay Slimane University, Beni Mellal, Morocco
https://www.usms.ac.ma

Ahmed Boumezzough, Ph.D.
Laboratory of Research in Physics and Engineering Sciences, Polydisciplinary Faculty
E-mail: ahmed.boumezzough@usms.ma
Sultan Moulay Slimane University, Beni Mellal, Morocco
https://www.usms.ac.ma

Said Safi, Ph.D.
Laboratory of Innovation in Mathematics, Applications, and Information Technologies, Polydisciplinary Faculty
https://orcid.org/0000-0003-3390-9037
E-mail: safi.said@gmail.com
Sultan Moulay Slimane University, Beni Mellal, Morocco
https://www.usms.ac.ma

Mathieu Pouliquen, Ph.D.
Laboratory of Engineering Systems, UNICAEN, ENSICAEN
E-mail: mathieu.pouliquen@unicaen.fr
Normandie University, Caen, France
https://www.unicaen.fr

Miloud Frikel, Ph.D.
Laboratory of Engineering Systems, UNICAEN, ENSICAEN
https://orcid.org/0000-0003-2178-1814
E-mail: miloud.frikel@ensicaen.fr
Normandie University, Caen, France
https://www.unicaen.fr