

# MULTI-OUTPUT KERNEL ADAPTIVE FILTERING WITH REDUCED COMPLEXITY

Diego Cuevas and Ignacio Santamaría

Dept. of Communications Engineering  
University of Cantabria, Spain  
email: {diego.cuevas,i.santamaria}@unican.es

## ABSTRACT

In this paper, two new multi-output kernel adaptive filtering algorithms are developed that exploit the temporal and spatial correlations among the input-output multivariate time series. They are multi-output versions of the popular kernel least mean squares (KLMS) algorithm with two different sparsification criteria. The first one, denoted as MO-QKLMS, uses the coherence criterion in order to limit the dictionary size. The second one, denoted as MO-RFF-KLMS, uses random Fourier features (RFF) to approximate the kernel functions by linear inner products. Simulation results with synthetic and real data are presented to assess convergence speed, steady-state performance and complexities of the proposed algorithms.

**Index Terms**—Multi-input multi-output (MIMO) regression, kernel adaptive filtering, Quantized Kernel Least Mean Square (QKLMS), random Fourier features.

## 1. INTRODUCTION

Nowadays, many modern machine learning applications require solving several decision making or prediction problems and, in many cases, the key to obtain better results and cope with a lack of data consists of exploiting the existing dependencies between those problems [1–4], which is often broadly referred to as multitask learning [5–7].

This paper focuses on the development of new kernel adaptive filtering (KAF) algorithms for multitask learning and, more specifically, for multi-output online regression [8]. Kernel adaptive filters that perform adaptive filtering in a high-dimensional reproducing kernel Hilbert space (RKHS) have been successfully applied over the past decades to a variety of nonlinear signal processing problems [9–11]. These algorithms have been extensively studied for noise cancellation, channel estimation and nonlinear system identification in online manner due to its universal modeling capabilities and modest computational complexity.

However, most of the multi-output KAF algorithms proposed in the recent literature are based on Gaussian Processes or on kernelized versions of the recursive-least square algorithm [12–14], which entail a high computational complexity. In this paper, we attempt to fill this gap by studying multi-output versions of the popular KLMS that exploit the temporal (intra) and spatial (inter) correlations among the input-output multivariate time series. A straightforward multi-output KLMS filter is derived by concatenating the time-embedded input time series and performing a matrix-valued kernel

expansion to obtain the multivariate output. This is equivalent to a multi-output LMS operating in a RKHS. We explore two sparsification methods to curb the linear growth of the KLMS with the number of training data. The first one uses the coherence criterion to limit the dictionary size and leads to a multi-output version of the quantized KLMS [15] (MO-QKLMS). The second one uses random Fourier features (RFF) [16] to approximate the kernel functions by linear inner products and leads to a multi-output version of the RFF-KLMS [17] (MO-RFF-KLMS). Simulation results with synthetic and real data are presented to assess convergence speed, steady-state performance and complexities of the proposed algorithms.

## 2. MIMO REGRESSION

We consider a multiple-input multiple-output (MIMO) real nonlinear system whose input consists of  $M$  time series,  $x_n^m$ ,  $m = 1, \dots, M$ , and produces  $P$  output signals,  $d_n^p$ ,  $p = 1, \dots, P$ . We assume the nonlinear MIMO system is causal with memory (time-embedding)  $L$  taps. Let  $\mathbf{x}_n^m = [x_n^m, x_{n-1}^m, \dots, x_{n-L+1}^m]^\top$  represent the time-embedded vector corresponding to the  $m$ -th input time series at time  $n$ . The input of the nonlinear MIMO system is denoted as  $\mathbf{x}_n = [(\mathbf{x}_n^1)^\top, \dots, (\mathbf{x}_n^M)^\top]^\top$ , and the corresponding output is  $\mathbf{d}_n = [d_n^1, \dots, d_n^P]^\top$ . Our goal is to learn a multi-output regression model or function  $f$  from a training data set  $\mathcal{S} = \{\mathbf{x}_n, \mathbf{d}_n\}_{n=1}^N$

$$f : \mathbb{R}^{LM} \rightarrow \mathbb{R}^P \\ \mathbf{x}_n \rightarrow \mathbf{d}_n.$$

Existing kernel least-mean square (KLMS) algorithms are highly efficient learning machines for the identification of single-output nonlinear systems, but their MIMO counterparts have not yet been widely researched. Nevertheless, a suboptimal approach to the problem could be to use  $P$  independent KLMS filters, each predicting or identifying one component of the  $P$ -dimensional output vector  $\mathbf{d}_n$ . That is, we could apply standard KLMS filters to learn independent single-input single-output regression models. However, this approach does not exploit the existing correlation among the input time series and therefore is clearly suboptimal. In the following section, we propose two multi-output KLMS-like algorithms that fully exploit the inter-series correlation with reduced complexity.

## 3. MULTI-OUTPUT KLMS

Kernel methods are based on a nonlinear transformation of the data  $\mathbf{x}_i$  into a high-dimensional feature space. In this feature space, inner products can be calculated by using a positive definite kernel function satisfying Mercer's condition [18]:  $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$ . This simple idea, also known as the *kernel trick*,

This work was supported by the Ministerio de Ciencia, Innovación y Universidades and AEI/FEDER funds of the E.U., under grant PID2019-104958RB-C43 (ADELE).

allows us to perform inner-product based algorithms implicitly in feature space by replacing all inner products by kernel functions. Many kernel functions exist, though the most commonly used is the Gaussian kernel

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_k^2}\right). \quad (1)$$

Thanks to the Representer Theorem [19], the output to a new input at time instant  $n + 1$ ,  $\mathbf{x}_{n+1}$ , for a single-output KLMS can be expressed as a kernel expansion in terms of the training data  $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$

$$y_{n+1} = \sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_{n+1}). \quad (2)$$

A straightforward multi-output KLMS generalization to (2) is

$$\mathbf{y}_{n+1} = \sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_{n+1}) = \mathbf{A}_n^\top \mathbf{k}_n, \quad (3)$$

where  $\mathbf{A}_n^\top = [\alpha_1, \dots, \alpha_n]$  is a  $P \times n$  matrix with columns  $\alpha_i = [\alpha_{1i}, \dots, \alpha_{Pi}]^\top$ , and  $\mathbf{k}_n = [\kappa(\mathbf{x}_1, \mathbf{x}_{n+1}), \dots, \kappa(\mathbf{x}_n, \mathbf{x}_{n+1})]^\top$  represents the kernels vector.

It is well known that the functional representation of the KLMS algorithm grows linearly with the number of processed data or dictionary size, leading to a heavy computational burden and huge memory requirements. Therefore, various online sparsification criteria have been developed to curb the growth of the kernel expansion [11, 20]. In the following subsections, we explore two sparsification approaches that are particularly well-suited to multi-output scenarios.

### 3.1. MO-QKLMS

A single-output KLMS algorithm, named Quantized Kernel Least Mean Squares (QKLMS), which applies the coherence criterion as a way to achieve sparsification was proposed in [15]. Here we propose a multi-output generalization of the QKLMS termed MO-QKLMS.

The MO-QKLMS sparsification procedure uses coherence as a measure to characterize the dictionary, which is defined in a kernel context as

$$\epsilon = \max_{i \neq j} \frac{|\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle|}{\sqrt{\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_i) \rangle \langle \Phi(\mathbf{x}_j), \Phi(\mathbf{x}_j) \rangle}}. \quad (4)$$

Using the unit-norm Gaussian kernel defined in (1) to compute the inner products, (4) simplifies to

$$\epsilon = \max_{i \neq j} |\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle| = \max_{i \neq j} |\kappa(\mathbf{x}_i, \mathbf{x}_j)|. \quad (5)$$

When the coherence between the new datum  $\mathbf{x}_{n+1}$  and the dictionary elements at time  $n$ ,  $\mathcal{D}_n$ , is below a given threshold

$$\max_{i \in \mathcal{D}_n} |\kappa(\mathbf{x}_i, \mathbf{x}_{n+1})| \leq \epsilon_u, \quad (6)$$

then, the MO-QKLMS includes  $\mathbf{x}_{n+1}$  into the dictionary and the filter coefficients are updated as

$$\mathbf{A}_{n+1} = \begin{bmatrix} \mathbf{A}_n \\ \mu \mathbf{e}_{n+1}^\top \end{bmatrix}, \quad (7)$$

where  $\mathbf{e}_{n+1} = \mathbf{d}_{n+1} - \mathbf{y}_{n+1}$  is the multivariate error vector. When the coherence is above the threshold, the new datum is not included

in the dictionary and the coefficients corresponding to the dictionary element closest to  $\mathbf{x}_{n+1}$ , say  $\mathbf{x}_j$ , are updated as  $\mathbf{A}_{n+1,j} = \mathbf{A}_{n,j} + \mu \mathbf{e}_{n+1}^\top$ , where  $\mu$  is the step-size and  $\mathbf{A}_{n,j}$  denotes the  $j$ -th column of  $\mathbf{A}_n$ . The MO-QKLMS algorithm is summarized in Algorithm 1.

---

#### Algorithm 1: MO-QKLMS

---

**Input:** Sequential multi-output training data  $\{\mathbf{x}_n, \mathbf{d}_n\}_{n=1}^N$ , Gaussian kernel width  $\sigma_k$ , step size  $\mu$ , threshold  $\epsilon_u$

- 1 Initialize  $\mathcal{D}_1 = \{\mathbf{x}_1\}$ ,  $\mathbf{A}_1 = \mu \mathbf{d}_1^\top$
  - 2 **for**  $n = 1 : N - 1$  **do**
  - 3   Select input-output training pattern  $(\mathbf{x}_{n+1}, \mathbf{d}_{n+1})$
  - 4   Compute kernels vector
 
$$\mathbf{k}_n = [\kappa(\mathbf{x}_1, \mathbf{x}_{n+1}) \quad \dots \quad \kappa(\mathbf{x}_n, \mathbf{x}_{n+1})]^\top$$
  - 5   Compute prediction  $\mathbf{y}_{n+1} = \mathbf{A}_n^\top \mathbf{k}_n$
  - 6   Compute error  $\mathbf{e}_{n+1} = \mathbf{d}_{n+1} - \mathbf{y}_{n+1}$
  - 7   Compute coherence  $\epsilon = \max_{i \in \mathcal{D}_n} |\kappa(\mathbf{x}_i, \mathbf{x}_{n+1})|$
  - 8   **if**  $\epsilon \leq \epsilon_u$  **then**
  - 9     New datum is included in the dictionary
 
$$\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{\mathbf{x}_{n+1}\}$$
  - 10    Update coefficients  $\mathbf{A}_{n+1} = \begin{bmatrix} \mathbf{A}_n \\ \mu \mathbf{e}_{n+1}^\top \end{bmatrix}$
  - 11   **else**
  - 12     Dictionary is maintained  $\mathcal{D}_{n+1} = \mathcal{D}_n$
  - 13     Find the closest dictionary element
 
$$\min_{j \in \mathcal{D}_n} \|\mathbf{x}_{n+1} - \mathbf{x}_j\|$$
  - 14     Update closest dictionary element coefficient
 
$$\mathbf{A}_{n+1,j} = \mathbf{A}_{n,j} + \mu \mathbf{e}_{n+1}^\top$$
  - 15   **end if**
  - 16 **end for**
- 

### 3.2. MO-RFF-KLMS

Another alternative to limit the growth of the multi-output KLMS thus reducing its computational complexity, called random Fourier features KLMS (RFF-KLMS), was proposed in [17]. Here we propose a multi-output generalization of the RFF-KLMS named MO-RFF-KLMS. This KLMS-type algorithm is based on finding good finite-dimensional approximations of the kernel functions. That is, it obtains a mapping  $\Psi : \mathbb{R}^{LM} \rightarrow \mathbb{R}^D$ , with  $D > LM$ , such that [16]

$$\langle \Psi(\mathbf{x}_i), \Psi(\mathbf{x}_j) \rangle \approx \kappa(\mathbf{x}_i, \mathbf{x}_j). \quad (8)$$

The feature vectors  $\Psi(\mathbf{x})$  are obtained using random Fourier features (RFF) maps. The underlying idea is based on Bochner's theorem, which guarantees that the Fourier transform of an appropriately scaled, shift-invariant kernel is a probability density function [16], [21]

$$\begin{aligned} \kappa(\mathbf{x}_i - \mathbf{x}_j) &= \int_{\mathbb{R}^{LM}} p(\boldsymbol{\omega}) e^{j\boldsymbol{\omega}^T(\mathbf{x}_i - \mathbf{x}_j)} d\boldsymbol{\omega} \\ &= \mathbb{E}_{\boldsymbol{\omega}} [\Psi_{\boldsymbol{\omega}}(\mathbf{x}_i) \Psi_{\boldsymbol{\omega}}(\mathbf{x}_j)^*]. \end{aligned} \quad (9)$$

Therefore,  $\Psi_{\boldsymbol{\omega}}(\mathbf{x}_i) \Psi_{\boldsymbol{\omega}}(\mathbf{x}_j)^*$  is an unbiased estimate of  $\kappa(\mathbf{x}_i, \mathbf{x}_j)$  when  $\boldsymbol{\omega}$  is drawn from  $p(\boldsymbol{\omega})$ . To reduce the variance of this estimate, a sample average of  $D$  randomly chosen  $\Psi_{\boldsymbol{\omega}}(\cdot)$  is used. Hence, the  $D$ -dimensional inner product  $\frac{1}{D} \sum_{k=1}^D \Psi_{\boldsymbol{\omega}_k}(\mathbf{x}_i) \cdot \Psi_{\boldsymbol{\omega}_k}(\mathbf{x}_j)$  is a low variance approximation to the kernel evaluation  $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ . This approximation improves exponentially fast in  $D$  [16].

In general, the features  $\Psi_\omega(\mathbf{x})$  are complex. However, they can be expressed using real valued cosine bases by exploiting the symmetry property of the kernel  $\kappa$ . Therefore, the mapping  $\Psi_\omega(\mathbf{x}) = \sqrt{2} \cos(\omega^\top \mathbf{x} + b)$ , where  $\omega$  is drawn from  $p(\omega)$  and  $b$  is drawn uniformly from  $[0, 2\pi]$ , also satisfies (9).

The vector  $\Psi(\mathbf{x}) = [\Psi_{\omega_1}(\mathbf{x}), \Psi_{\omega_2}(\mathbf{x}), \dots, \Psi_{\omega_D}(\mathbf{x})]^\top$  is thus a  $D$ -dimensional RFF of the input vector  $\mathbf{x}$ . This mapping satisfies the approximation in (8). For approximating the Gaussian kernel of width  $\sigma_k$ , we therefore draw  $\omega_i, i = 1, \dots, D$ , from the normal  $\mathcal{N}(0, \mathbf{I}_d/\sigma_k^2)$  distribution.

Since the RFF space is finite dimensional, it now becomes possible to directly work with the filter weights  $\Omega$  in this space. Therefore, the kernel expansion (3) in this case yields

$$\mathbf{y}_{n+1} = \Omega_n^\top \Psi(\mathbf{x}_{n+1}). \quad (10)$$

Notice that the update equations in the RFF space become exactly the same as the classical multi-output LMS algorithm

$$\Omega_{n+1} = \Omega_n + \mu \Psi(\mathbf{x}_{n+1}) \mathbf{e}_{n+1}^\top, \quad (11)$$

where  $\mathbf{e}_{n+1} = \mathbf{d}_{n+1} - \mathbf{y}_{n+1}$  is the multivariate error vector and  $\mu$  is the step size. The MO-RFF-KLMS algorithm is summarized in Algorithm 2.

---

**Algorithm 2: MO-RFF-KLMS**

---

**Input:** Sequential multi-output training data  $\{\mathbf{x}_n, \mathbf{d}_n\}_{n=1}^N$ , Gaussian kernel width  $\sigma_k$ , step size  $\mu$ , RFF dimension  $D$

- 1 Draw i.i.d.  $\{\omega_i\}_{i=1}^D$  from  $\mathcal{N}(0, \mathbf{I}_d/\sigma_k^2)$ , where  $d = LM$  is the input space dimension
  - 2 Draw i.i.d.  $\{b_i\}_{i=1}^D$  from  $U(0, 2\pi)$
  - 3 Initialize  $\Omega_1 = 0 \in \mathbb{R}^{D \times P}$
  - 4 **for**  $n = 1 : N - 1$  **do**
  - 5     Select input-output training pattern  $(\mathbf{x}_{n+1}, \mathbf{d}_{n+1})$
  - 6     Compute RFF vector
 
$$\Psi(\mathbf{x}_{n+1}) = [\psi_{\omega_1}(\mathbf{x}_{n+1}), \dots, \psi_{\omega_D}(\mathbf{x}_{n+1})]^\top,$$
 where each  $\psi_{\omega_i}(\mathbf{x}_{n+1}) = \cos(\omega_i^\top \mathbf{x}_{n+1} + b_i)$
  - 7     Compute prediction  $\mathbf{y}_{n+1} = \Omega_n^\top \Psi(\mathbf{x}_{n+1})$
  - 8     Compute error  $\mathbf{e}_{n+1} = \mathbf{d}_{n+1} - \mathbf{y}_{n+1}$
  - 9     Update  $\Omega_{n+1} = \Omega_n + \mu \Psi(\mathbf{x}_{n+1}) \mathbf{e}_{n+1}^\top$
  - 10 **end for**
- 

## 4. SIMULATION RESULTS

In this section, we provide two simulation examples to assess the performance of MO-QKLMS and MO-RFF-KLMS in comparison to their single-output versions in stationary and non-stationary environments. The first set of experiments is conducted using a synthetic dataset and the second one using real data. The optimal kernel width  $\sigma_k$  for both datasets is estimated from the first 200 samples using the parameter estimation script included in the toolbox KAFBOX [22]. The curves for the synthetic dataset are averaged over 500 independent Monte Carlo runs and the ones for the real dataset over 50 runs. The mean squared error (MSE) shown in figures corresponds to the MSE averaged over the  $P$  output time series. All the experiments are conducted using MATLAB that runs on a laptop with an Intel(R) Core(TM) i7-7500U 2.70GHz CPU and 8GB RAM.

### 4.1. Nonlinear MIMO System Identification

In the first experiment, we consider a MIMO 3x3 nonlinear system. Firstly, 3 Gaussian white time-series of 4000 samples each are generated as input signals. Secondly, these signals are filtered through a MIMO linear system that introduces intra- and inter-correlation among the time-series. To achieve this, the MIMO filter is  $\mathbf{H} = \mathbf{C} \otimes \mathbf{h}$  where  $\mathbf{h} = [1, 0.6]$  determines the temporal (intra-) correlation and the covariance matrix  $\mathbf{C}$ , whose elements are  $c_{ij} = 1$  if  $i = j$  and  $c_{ij} = \rho$  if  $i \neq j$ , determines the spatial (inter-) correlation. The time-embedding is  $L = 2$ . Finally, we apply a memoryless quadratic nonlinearity with coefficient  $\gamma = 0.3$  (fixed) and add white Gaussian noise with variance  $\sigma^2 = 0.01$  that models the measurement noise.

The values of the parameters used in the different adaptive filters under comparison are listed in Table 1, where  $\mu$  is the step size,  $m$  is the final dictionary size,  $\epsilon_u$  is the coherence criterion threshold and  $D$  is the RFF dimension. In order to make a fair comparison between the different KAF algorithms, SO-QKLMS and MO-QKLMS parameter  $\epsilon_u$  is chosen so that  $m = 1000$ .

Algorithm	Parameters
SO-KLMS	$\mu = 0.4, m = 1000$
MO-KLMS	$\mu = 0.4, m = 1000$
SO-QKLMS	$\mu = 0.4, \epsilon_u = 0.1$
MO-QKLMS	$\mu = 0.4, \epsilon_u = 1.22$
SO-RFF-KLMS	$\mu = 0.4, D = 1000$
MO-RFF-KLMS	$\mu = 0.4, D = 1000$

**Table 1.** Algorithms' parameters used in nonlinear MIMO system identification example.

Figure 1 shows the convergence curves obtained for the nonlinear MIMO system with spatial correlation coefficient  $\rho = 0.3$ . As expected, MO-QKLMS and MO-RFF-KLMS provide much lower steady-state MSE than their single-output counterparts. Moreover, the two proposed algorithms outperform a multi-output version of the standard KLMS without sparsification (denoted as MO-KLMS) and its corresponding single-output version. Further, in this example MO-QKLMS is slightly better than MO-RFF-KLMS. In this case, the optimal value chosen via cross-validation for the kernel width is  $\sigma_k = 2.23$ .

Figure 2 shows the convergence curves obtained for the nonlinear MIMO system when we increase the spatial correlation coefficient to  $\rho = 0.6$ . For this system, the optimal value chosen via cross-validation for the kernel width is  $\sigma_k = 1.67$ . Again, MO-QKLMS and MO-RFF-KLMS show a better performance in terms of steady-state MSE than that of their single-output counterparts and the MO-KLMS algorithm.

Meanwhile, it should be pointed out that the gap between the multi-output algorithms and their single-output versions increases as the spatial correlation coefficient  $\rho$  is increased from 0.3 to 0.6. This result supports the intuitive idea that multi-output algorithms perform better than their single-output versions because they take advantage of the spatial correlation among time series.

### 4.2. Lorenz Chaotic Time-Series Prediction

In the second experiment, we consider the Lorenz chaotic system determined by the differential equations [23]:

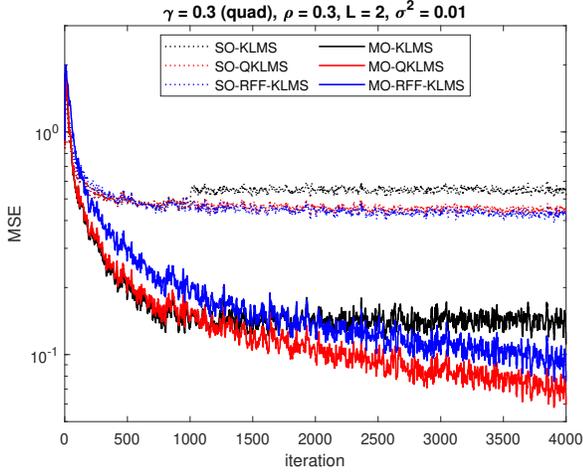


Fig. 1. Learning curves for  $\gamma = 0.3$  and  $\rho = 0.3$ .

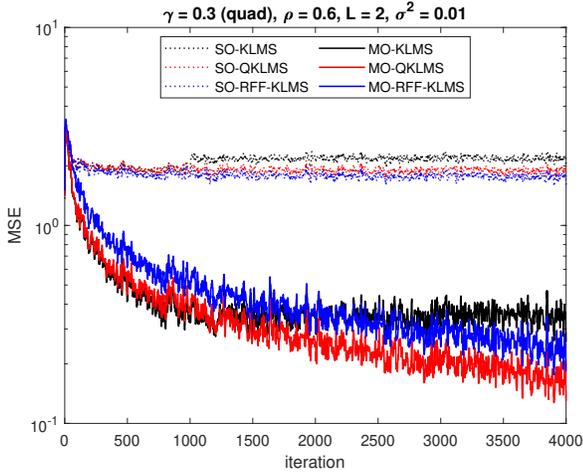


Fig. 2. Learning curves for  $\gamma = 0.3$  and  $\rho = 0.6$ .

$$\begin{cases} dx^1(t)/dt = \sigma(x^2(t) - x^1(t)) \\ dx^2(t)/dt = -x^2(t) - x^1(t)x^3(t) + Rx^1(t) \\ dx^3(t)/dt = x^1(t)x^2(t) - bx^3(t) \end{cases} \quad (12)$$

with  $\sigma = 10$ ,  $R = 28$  and  $b = 8/3$ . A prediction horizon of  $h = 1$  and an embedding of  $L = 3$  are used. The values of the parameters used in the different algorithms are listed in Table 2. In this case, the optimal value chosen for kernel width is  $\sigma_k = 1.94$ .

Algorithm	Parameters
SO-QKLMS	$\mu = 0.9, \epsilon_u = 0.07$
MO-QKLMS	$\mu = 0.9, \epsilon_u = 0.22$
SO-RFF-KLMS	$\mu = 0.4, D = 1000$
MO-RFF-KLMS	$\mu = 0.4, D = 1000$

Table 2. Algorithms' parameters for Lorenz time-series example.

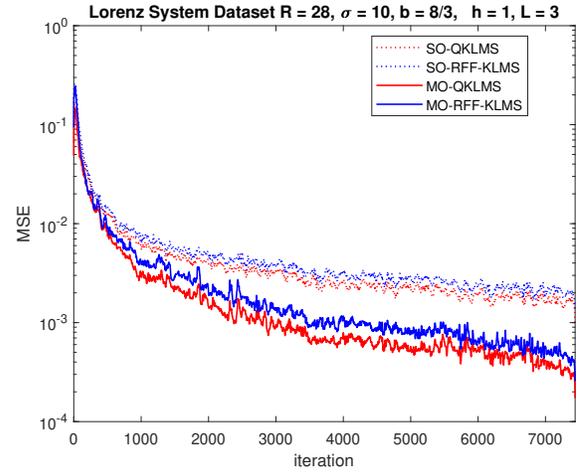


Fig. 3. Learning curves for Lorenz time-series example.

Figure 3 shows again that MO-QKLMS and MO-RFF-KLMS have lower steady-state MSE than that of their single-output versions, with MO-QKLMS narrowly outperforming MO-RFF-KLMS.

Table 3 shows a comparison between the two proposed algorithms in terms of training time, computational complexity (measured in FLOPS) and storage requirements for the Lorenz chaotic time-series dataset. It can be observed that MO-RFF-KLMS outperforms MO-QKLMS in terms of training time and computational complexity, thus making MO-RFF-KLMS our preferred option for real-time applications or systems with limited computing capabilities.

Algorithm	Training time	FLOPS	Storage
MO-QKLMS	0.35 s	79006	96 kB
MO-RFF-KLMS	0.15 s	50025	104 kB

Table 3. Training time and complexity for Lorenz time-series example.

## 5. CONCLUSIONS

Two multi-output KLMS-like algorithms are proposed in this paper: MO-QKLMS, which uses coherence as sparsification criterion to limit the growth of the dictionary size, and MO-RFF-KLMS, which uses random Fourier features to limit the complexity of the algorithm. Both algorithms outperform their single-output counterparts in terms of steady-state MSE and convergence speed. Both algorithms provide similar performance in terms of steady-state MSE, but the MO-RFF-KLMS proves to be a better choice in terms of computational complexity and training time. Future work can harness the proposed algorithms to solve multitask learning problems over an underlying graph as in [24, 25].

## 6. REFERENCES

- [1] M. A. Osborne, S. J. Roberts, A. Rogers, S. D. Ramchurn, and N. R. Jennings, "Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes," in *2008 International Conference on Information Processing in Sensor Networks (IPSN 2008)*, pp. 109–120, 2008.
- [2] P. Goovaerts, *Geostatistics for Natural Resource Evaluation*, vol. 42. 1997.
- [3] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 283–298, 2008.
- [4] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda, "Maximal margin labeling for multi-topic text categorization," in *Advances in Neural Information Processing Systems*, vol. 17, 2004.
- [5] M. Álvarez, L. Rosasco, and N. Lawrence, "Kernels for vector-valued functions: A review," *Foundations and Trends® in Machine Learning*, vol. 4, pp. 195–266, 2012.
- [6] Y. Zhang and Q. Yang, "A survey on multi-task learning," 2018.
- [7] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, no. 21, pp. 615–637, 2005.
- [8] D. Xu, Y. Shi, I. W. Tsang, Y. S. Ong, C. Gong, and X. Shen, "Survey on multi-output learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2409–2429, 2020.
- [9] W. Liu, J. C. Principe, and S. Haykin, *Kernel Adaptive Filtering*. Hoboken, NJ: John Wiley, 2010.
- [10] F. Pérez-Cruz, S. Van Vaerenbergh, J. J. Murillo-Fuentes, M. Lázaro-Gredilla, and I. Santamaria, "Gaussian processes for nonlinear signal processing: An overview of recent advances," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 40–50, 2013.
- [11] S. Van Vaerenbergh, *Adaptive Kernel Learning for Signal Processing*, ch. 9. Wiley, 2018.
- [12] M. Álvarez and N. Lawrence, "Computationally efficient convolved multiple output Gaussian processes," *Journal of Machine Learning Research*, vol. 12, pp. 1459–1500, 2011.
- [13] L. Yang, K. Wang, and L. Mihaylova, "Online sparse multi-output Gaussian process regression and learning," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 2, pp. 258–272, 2019.
- [14] J. Zhao and S. Sun, "Variational dependent multi-output Gaussian process dynamical systems," *Journal of Machine Learning Research*, vol. 17, no. 121, pp. 1–36, 2016.
- [15] B. Chen, S. Zhao, P. Zhu, and J. C. Principe, "Quantized kernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012.
- [16] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, vol. 3, pp. 1–8, 2007.
- [17] A. Singh, N. Ahuja, and P. Moulin, "Online learning with kernels: Overcoming the growing sum problem," in *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, 2012.
- [18] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995.
- [19] B. Schölkopf, A. J. Smola, and F. Bach, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2002.
- [20] C. Richard, J. C. M. 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.
- [21] M. Reed and B. Simon, *Fourier Analysis, Self-Adjointness*. No. II in Methods of Modern Mathematical Physics, Acad. Press, 2007.
- [22] S. Van Vaerenbergh and I. Santamaría, "A comparative study of kernel adaptive filtering algorithms," in *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pp. 181–186, Aug. 2013. Software available at <https://github.com/steven2358/kafbox/>.
- [23] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.
- [24] R. Nassif, S. Vlaski, C. Richard, and A. H. Sayed, "Learning over multitask graphs – Part I: Stability analysis," *IEEE Open Journal on Signal Processing*, vol. 1, pp. 28–45, 2020.
- [25] R. Nassif, S. Vlaski, C. Richard, and A. H. Sayed, "Learning over multitask graphs – Part II: Performance analysis," *IEEE Open Journal on Signal Processing*, vol. 1, pp. 46–63, 2020.