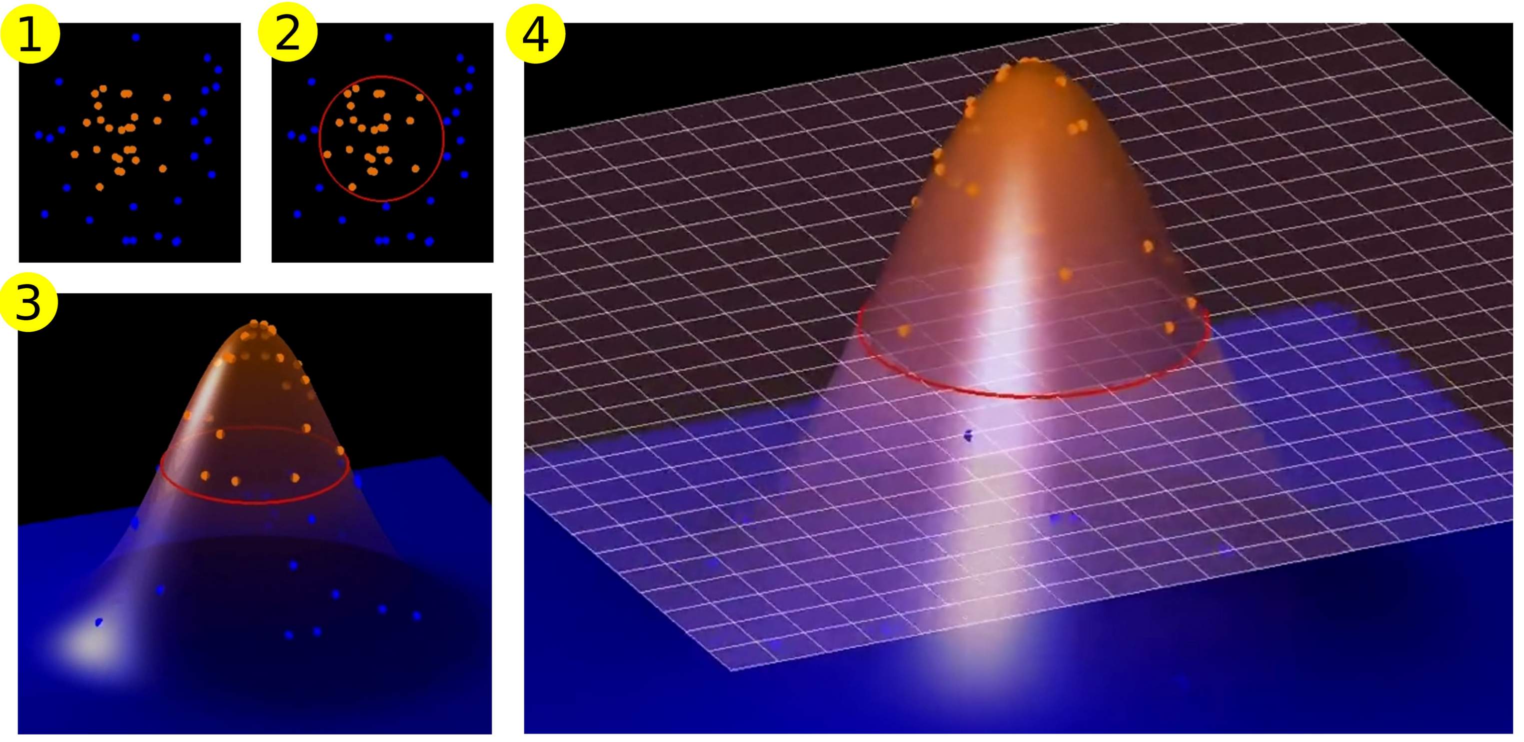
Improving Multikernel Adaptive Filtering with Selective Bias

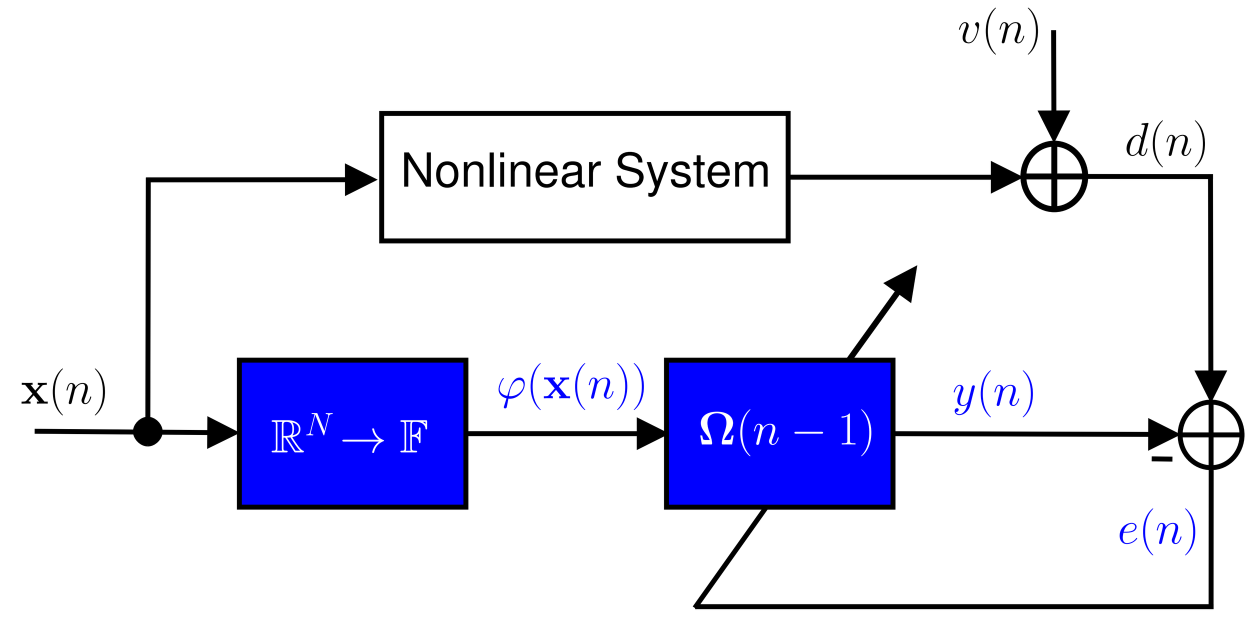
Magno T. M. Silva, Renato Candido
{magno, renatocan}@lps.usp.br
Universidade de São Paulo, Brazil

Jerónimo Arenas-García, Luis A. Azpicueta-Ruiz
{jarenas, azpicueta}@tsc.uc3m.es
Universidad Carlos III de Madrid, Spain

Introduction and Problem Formulation

- Kernel adaptive filters (KAFs) are important tools to **solve nonlinear problems**
- The input vector $\mathbf{x}(n) \in \mathbb{R}^N$ is **projected into a high dimension feature space** $\mathbb{F}$ as $\varphi(\mathbf{x}(n))$, where a standard linear adaptive filter is employed
- **Kernel trick**: $\varphi(\mathbf{x})^T\varphi(\mathbf{x}') = \kappa(\mathbf{x},\mathbf{x}')$, where $\kappa(\cdot,\ \cdot)$ is a Mercer's kernel



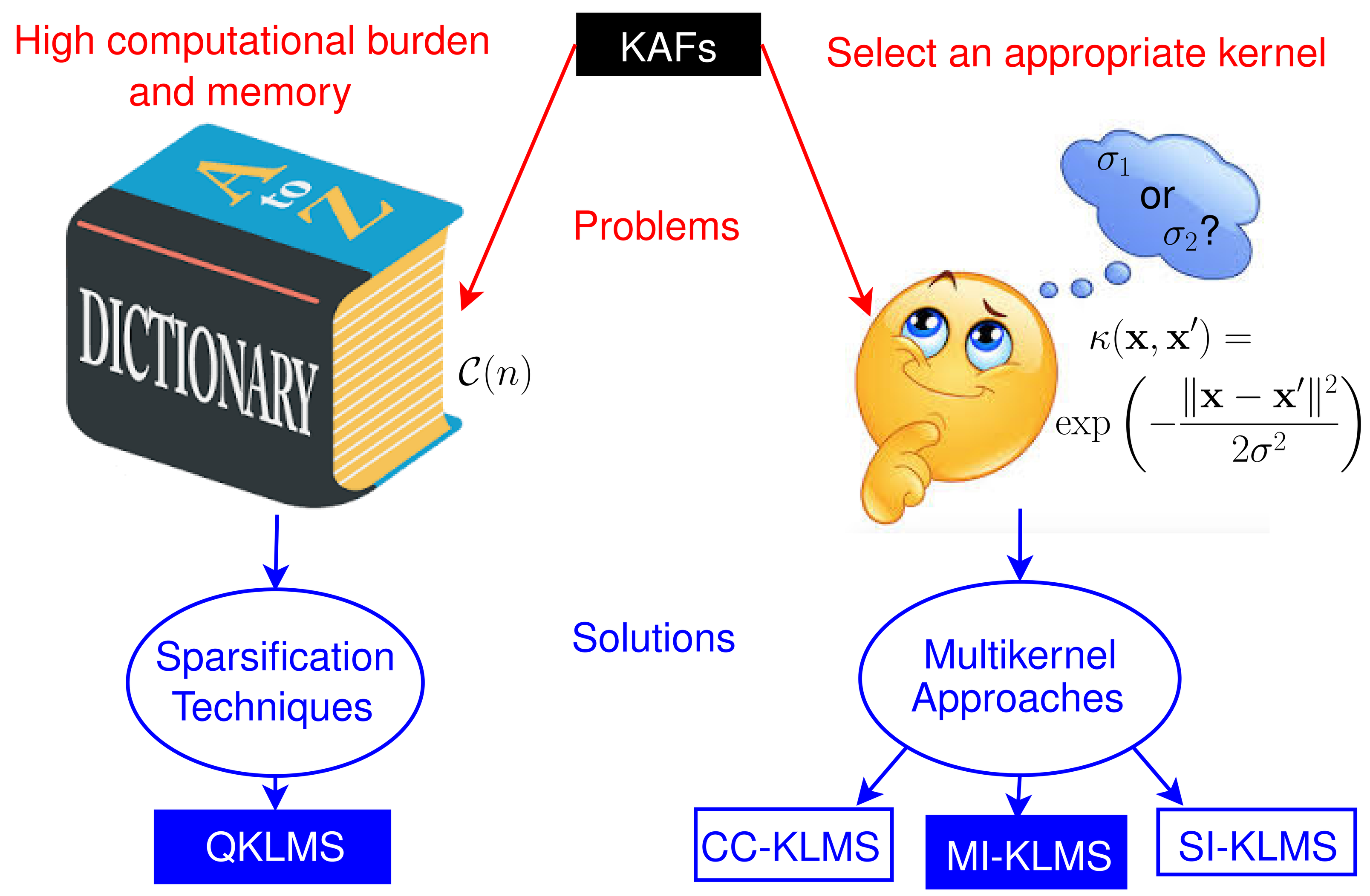


KLMS applied for nonlinear system identification, where $v(n)$ is a measurement noise

- The filter output of the **kernel least-mean-squares (KLMS)** algorithm is computed as

$$y(n) = \varphi(\mathbf{x}(n))^T\boldsymbol{\Omega}(n-1) = \sum_{i=1}^{n-1} \mu e(i)\kappa(\mathbf{x}(n),\mathbf{x}(i))$$

where $e(i) = d(i) - y(i)$ and μ is a step size



- **QKLMS: Quantized KLMS**, which is similar to the sparsified KLMS with novelty criterion

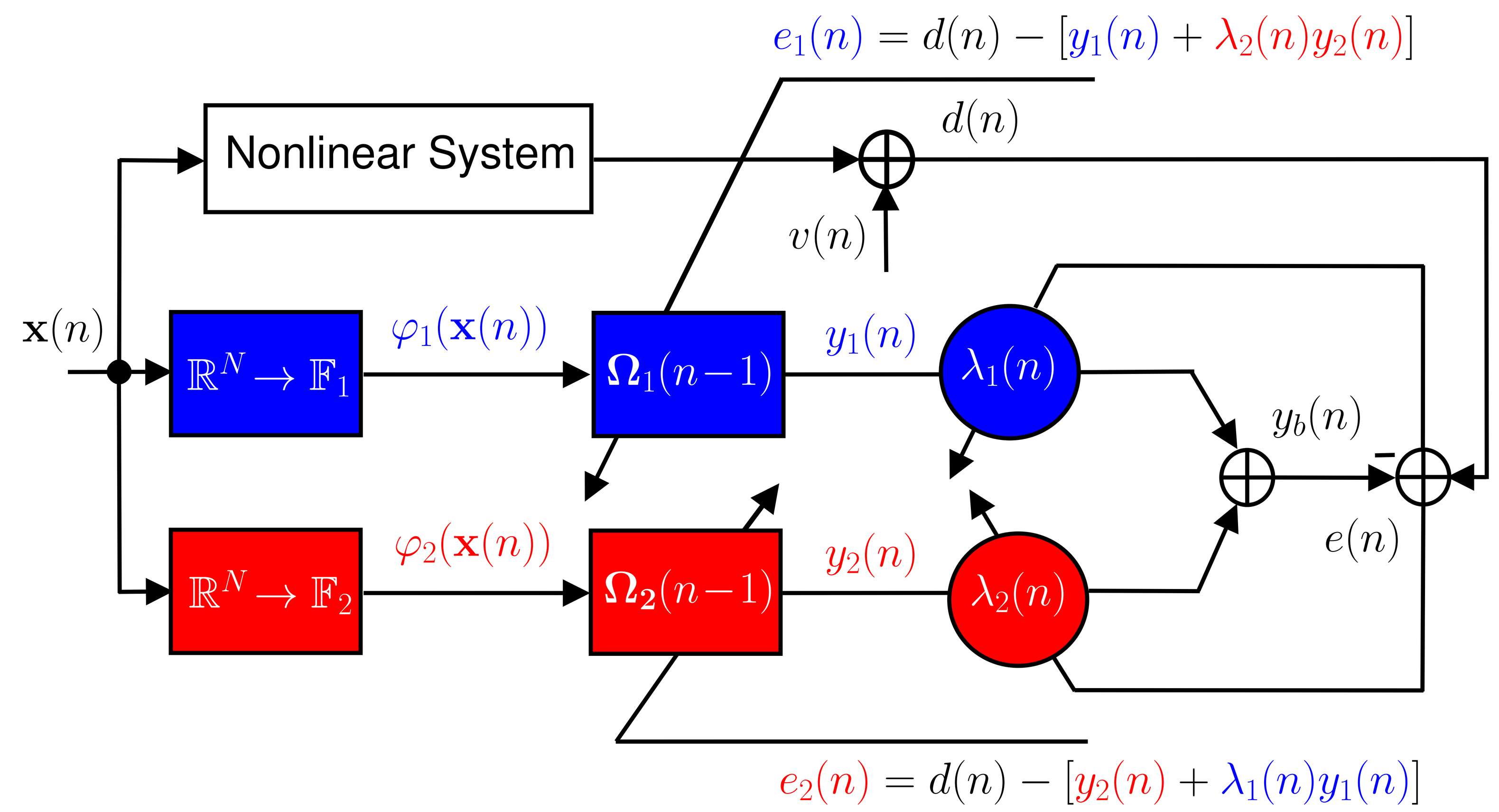
  dis$(\mathbf{x}(n),\mathcal{C}(n)) = \min_{1\le j\le N_c(n)} \|\mathbf{x}(n)-\mathbf{x}(c_j)\|$

  If dis$(\mathbf{x}(n),\ \mathcal{C}(n)) \le \varepsilon$, keep the dictionary unchanged and update a_{j^*} as $a_{j^*}(n) = a_{j^*}(n-1) + \mu e(n)$,
  where
  $j^* = \arg\min_{1\le j\le N_c(n)} \|\mathbf{x}(n)-\mathbf{x}(c_j)\|$
  Otherwise, include $\mathbf{x}(n)$ and $a_{N_c(n)+1}(n) = \mu e(n)$ to the dictionary
- **CC-KLMS: convex combination of two KLMS filters**, in which the global output is a convex combination of the outputs of two KLMS filters running in parallel
- **MI-KLMS: multiple-input multikernel LMS**, in which L KLMS filters in parallel are adapted using a single error signal
- **SI-KLMS: single-input multikernel LMS**, where the kernel function is a convex combination of kernels

- MI-KLMS generally outperforms SI-KLMS and may outperform the convex combination of two KLMS filters
- **If the parameters of one kernel component are poorly adjusted**, the convex combination is able to select the best component filter and may outperform SI-KLMS and MI-KLMS

We propose **a scheme to improve the selection of kernel filters in MI-KLMS**, by multiplying the output of each kernel filter by an adaptive biasing factor in $[0, 1]$

Proposed Scheme



Robust MI-KLMS (R-MI-KLMS) with $L = 2$ KAFs applied to nonlinear system identification

- An important difference between the MI-KLMS scheme and our proposal is related to the **error employed to adapt each kernel**
- R-MI-KLMS permits to **weight the output of each kernel activating or deactivating the output** of unnecessary kernels in the global filter output
- We reinterpret the output of each branch of R-MI-KLMS as a convex combination with a virtual kernel whose output is always zero, i.e.,

$$y_b(n) = \sum_{\ell=1}^{L} \lambda_\ell(n) y_\ell(n) = \sum_{\ell=1}^{L} \lambda_\ell(n) y_\ell(n) + [1-\lambda_\ell(n)]\cdot 0$$

- Instead of adapting directly $\lambda_\ell(n)$, we adapt an auxiliary parameter α_ℓ:

$$\alpha_\ell(n) = \alpha_\ell(n-1) + \frac{\mu_{\alpha_\ell}}{p_\ell(n)} e(n) y_\ell(n) \lambda_\ell(n)[1-\lambda_\ell(n)]$$

where μ_{α_ℓ} is a step size and $p_\ell(n) = \beta p_\ell(n-1) + (1-\beta) y_\ell^2(n)$, with $0 \ll \beta < 1$

- $\lambda_\ell(n)$ and $\alpha_\ell(n)$ are related through a sigmoidal function

$$\lambda_\ell(n) = \text{sgm}[\alpha_\ell(n-1)] = \frac{1}{1+e^{-\alpha(n-1)}}$$

- $\alpha_\ell(n)$ has to be restricted to a range of $[-\alpha^+, \alpha^+]$ to avoid the paralysis of its updating

ESCOLA POLITÉCNICA

Simulation Results

In all multikernel schemes, we consider

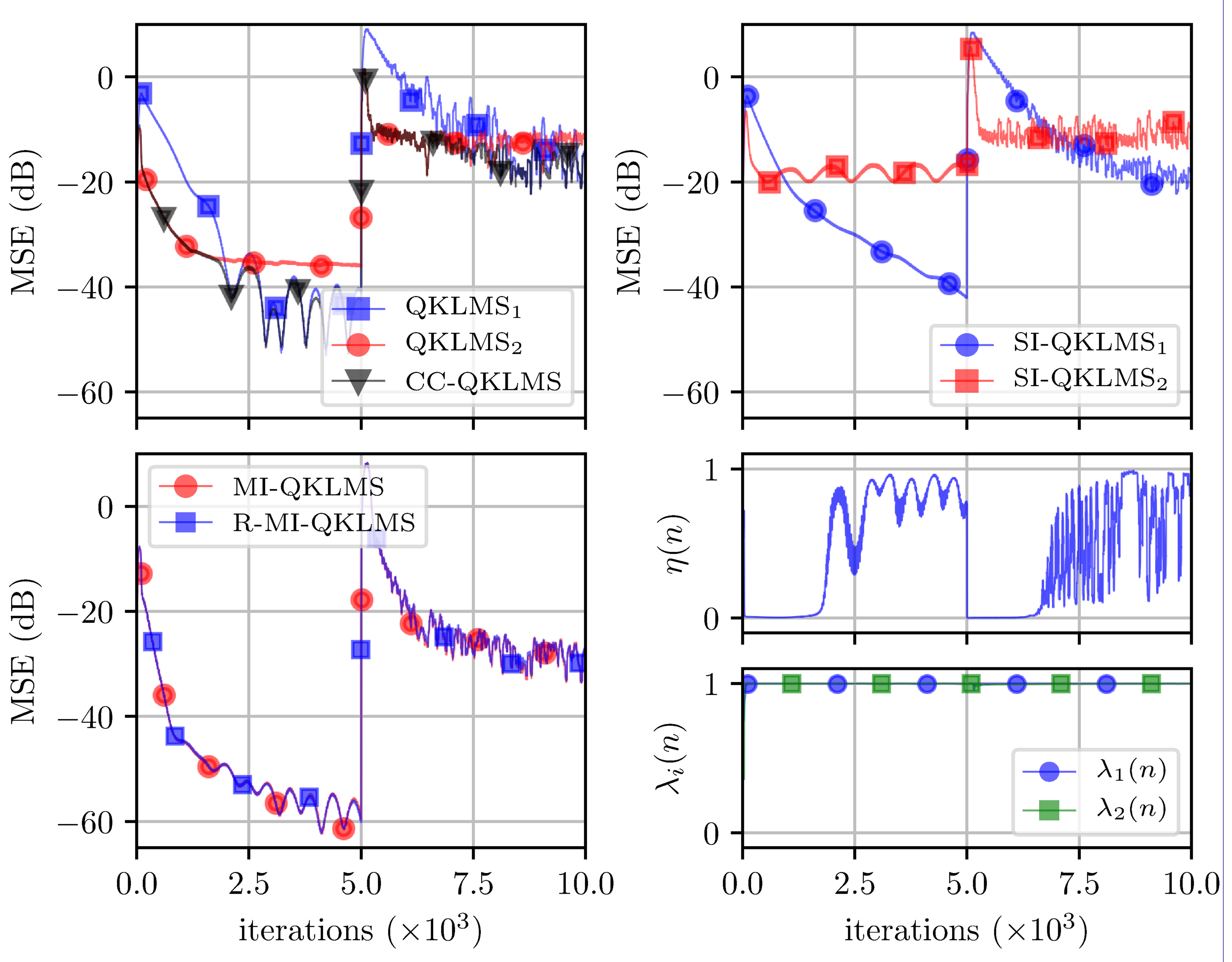
- the **QKLMS** algorithm due to its inherent advantages
- the **Gaussian kernel** function and $L = 2$ filters

The schemes were applied to **nonlinear prediction** and **nonlinear system identification** with an abrupt change in the middle of the simulation, assuming the parameters:

Algorithm	Parameters
QKLMS$_1$	$\mu_1=0.05,\ \sigma_1,\ \varepsilon_1=0.05$
QKLMS$_2$	$\mu_2=0.5,\ \sigma_2,\ \varepsilon_2=0.5$
CC-QKLMS	$\alpha^+=4,\ \beta=0.9,\ \mu_\alpha$
SI-QKLMS$_1$	$\mu=0.05,\ \beta_1=\beta_2=0.5,\ \sigma_1,\ \sigma_2,\ \varepsilon=0.05$
SI-QKLMS$_2$	$\mu=0.5,\ \beta_1=\beta_2=0.5,\ \sigma_1,\ \sigma_2,\ \varepsilon=0.5$
MI-QKLMS	$\mu_1=0.05,\ \mu_2=0.5,\ \sigma_1,\ \sigma_2,\ \varepsilon_1=0.05,\ \varepsilon_2=0.5$
R-MI-QKLMS	$\mu_1=0.05,\ \mu_2=0.5,\ \sigma_1,\ \sigma_2,\ \varepsilon_1=0.05,\ \varepsilon_2=0.5,\ \mu_{\alpha_1},\ \mu_{\alpha_2}$

Nonlinear prediction ①

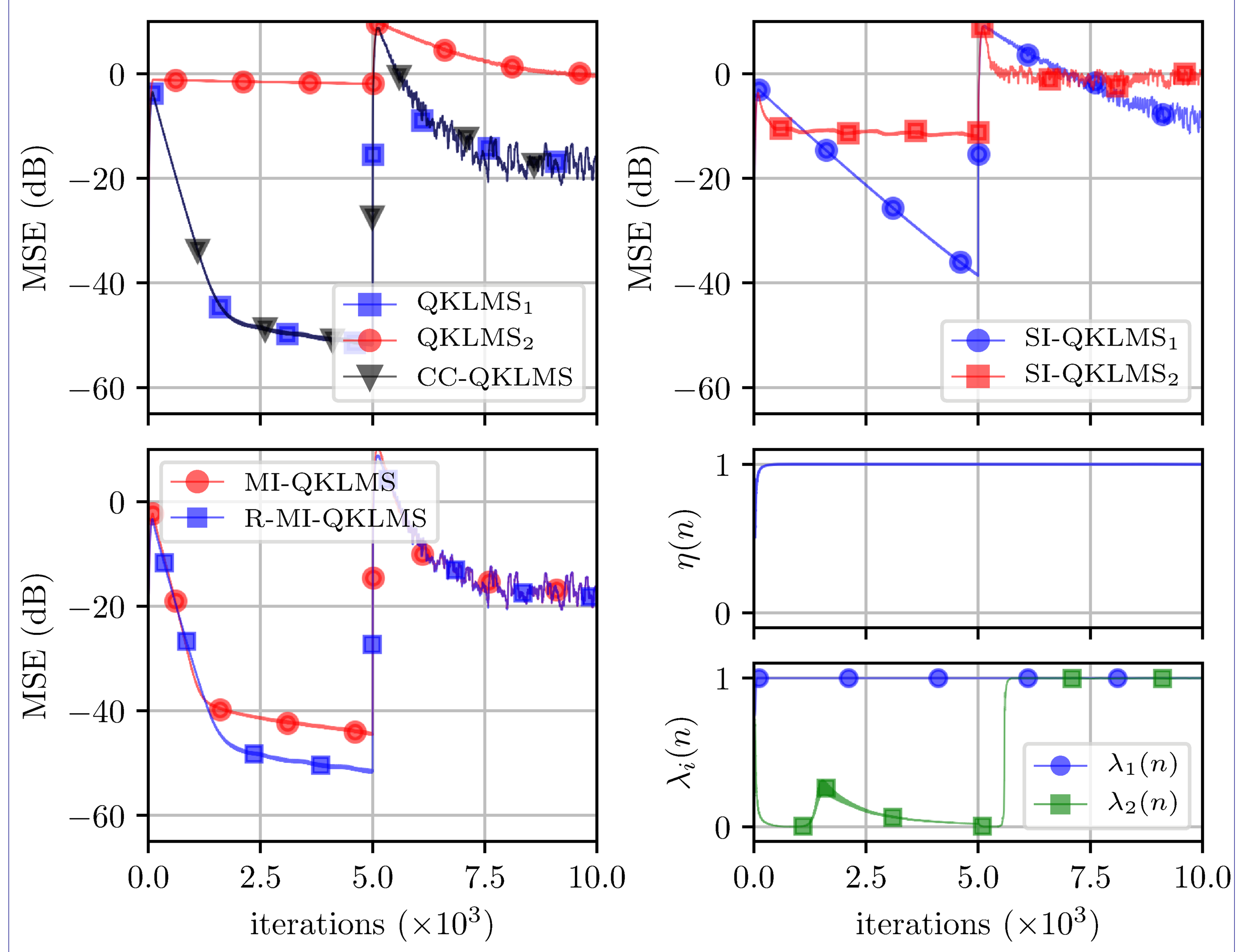
Settings: $\sigma_1=0.1,\ \sigma_2=1,\ \mu_\alpha=\mu_{\alpha_1}=\mu_{\alpha_2}=1.5$



- CC-QKLMS performs as its best component filter
- SI-QKLMS$_1$ outperforms SI-QKLMS$_2$
- MI-QKLMS and its robust version present the same performance and outperform other multiple kernel solutions

Nonlinear prediction ②

Settings: $\sigma_1=0.2,\ \sigma_2=100,\ \mu_\alpha=\mu_{\alpha_1}=\mu_{\alpha_2}=0.3$



- $\sigma_2 = 100$ does not lead to good results for a monokernel filter
- CC-QKLMS follows QKLMS$_1$
- SI-QKLMS$_1$ scheme presents a lower convergence rate than that of the monokernel QKLMS$_1$
- For $1200 < n < 5000$, $\sigma_2 = 100$ degrades the performance of MI-QKLMS, which is avoided by R-MI-QKLMS

Nonlinear system identification ③

Settings: $\sigma_1 = 1,\ \sigma_2 = 0.1,\ \mu_\alpha=\mu_{\alpha_1}=\mu_{\alpha_2}=0.1$

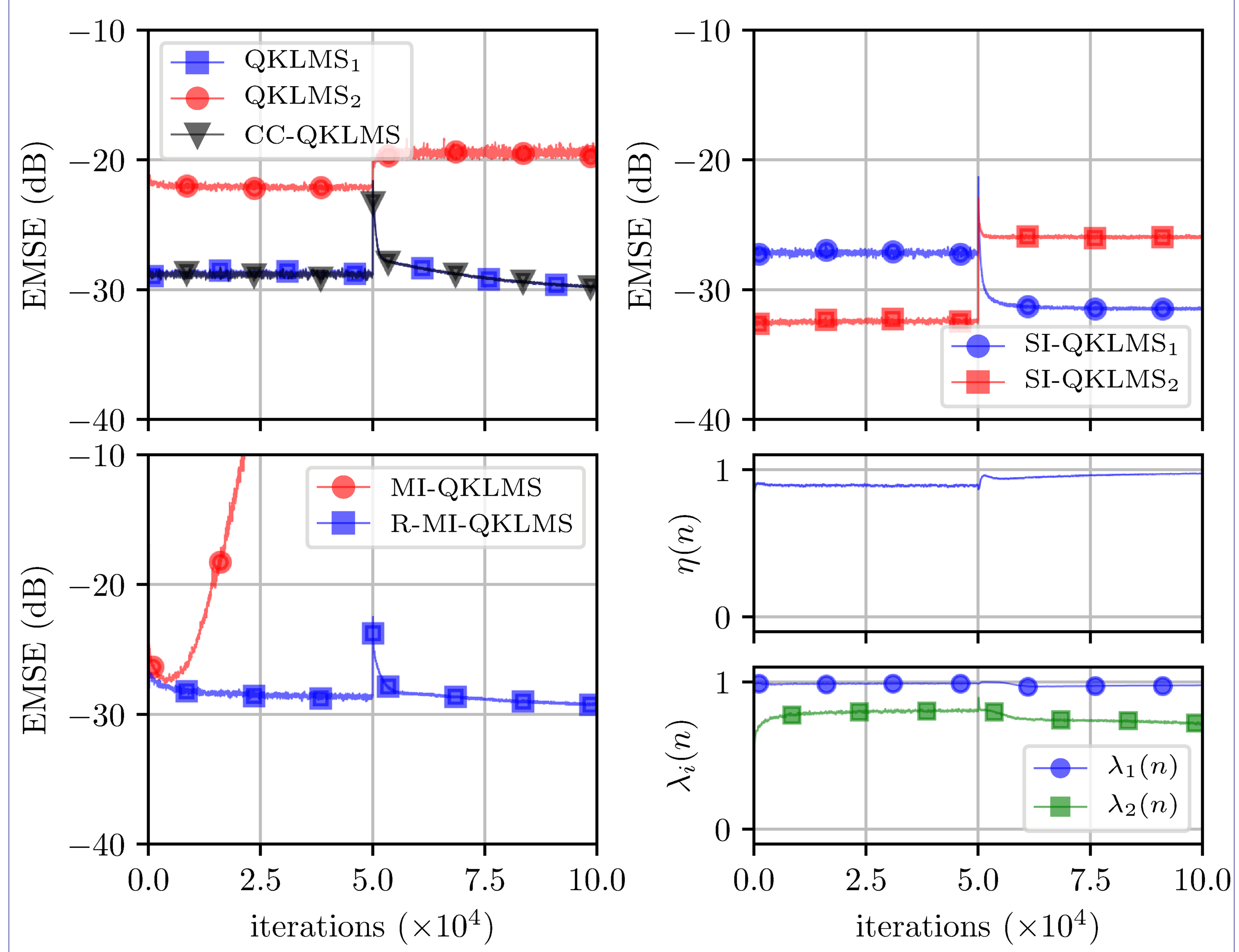
Gain of R-MI-QKLMS in relation to other schemes in terms of steady-state EMSE (dB) for 2nd system

SNR	CC-QKLMS	SI-QKLMS$_1$	SI-QKLMS$_2$	MI-QKLMS
−25	10.5	9.4	23.9	20.7
−20	6.7	6.8	15.3	16.6
−15	3.3	3.2	11.4	13.4
−10	1.4	0.9	8.5	9.9
−5	0.6	0.5	6.3	8.4
0	0.4	0.4	4.5	5.2
5	0.1	0.3	2.2	3.5
10	0.1	0.5	1.6	2.8
15	0	0.2	0.5	1.6

- R-MI-QKLMS outperforms other multikernel schemes for the considered range of SNR, especially for low SNRs

Nonlinear system identification ④

Settings: $\sigma_1=1,\ \sigma_2=0.1,\ \mu_\alpha=\mu_{\alpha_1}=\mu_{\alpha_2}=1$



- CC-QKLMS performs as the best component filter
- SI-QKLMS$_2$ outperforms SI-QKLMS$_1$ in the identification of the first system but for the second system, SI-QKLMS$_1$ outperforms SI-QKLMS$_2$
- MI-QKLMS presents a poor performance
- R-MI-QKLMS is able to minimize the degrading effects of unnecessary kernels

Conclusions

The proposed scheme

- presents a computational cost slightly higher than that of MI-QKLMS and
- **can outperform other multikernel solutions** when the settings of one or more kernels are not appropriate and/or the SNR is low

References

- W. Gao, C. Richard, J. C. M. Bermudez, and J. Huang, “Convex combinations of kernel adaptive filters,” in *Proc. of IEEE MLSP*, Sep. 2014
- 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, pp. 22–32, Jan 2012
- M. Yukawa, “Multikernel adaptive filtering,” *IEEE TSP*, vol. 60, no. 9, pp. 4672–4682, Sep. 2012
- F. A. Tobar, S. Y. Kung, and D. P. Mandic, “Multikernel least mean square algorithm,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 265–277, Feb. 2014
- G. R. G Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, “Learning the kernel matrix with semidefinite programming,” *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.
- M. Lazaro-Gredilla, L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-Garcia, “Adaptively biasing the weights of adaptive filters,” *IEEE TSP*, vol. 58, pp. 3890–3895, Jul. 2010

FAPESP

CNPq

ICASSP2018