An Online Variable Selection Method using Recursive Least Squares

Francisco Souza and Rui Araújo
Institute of Systems and Robotics (ISR-UC), and
Department of Electrical and Computer Engineering (DEEC-UC),
University of Coimbra, Pólo II, PT-3030-290 Coimbra
fasouza@isr.uc.pt, rui@isr.uc.pt

Abstract

This paper proposes a method for online variable selection and model learning (AdaFSML-RLS) to be applied in industrial applications in the context of adaptive soft sensors. In the proposed method the model learning is made online and recursivelly, i.e it is not necessary to store the past values of data while learning the model. Furthermore, the proposed method has the capability of tracking the real time correlation coefficient between each variable and the target, allowing the knowledge about the importance of variables over the time. Moreover, in this method is not necessary to have any knowledge about the process or variables. The method was sucessfully applied in two datasets, an artificial dataset and in a real-world dataset.

Keywords: adaptive soft sensors, recursive least squares, adaptive feature selection, free lime estimation.

1 Introduction

Today, soft sensors have many applications in industry (e.g. fault detection, process monitoring, the prediction of critical variables, and control) [4, 7, 13]. Soft sensor consists on the prediction of critical or hard-to-measure variables, where easy-to-measure variables are used in a model to predict the hard-to-measure variables. The model can be constructed using the underlying knowledge about the process (white-box modeling), or using the available data to learn a data-driven model (data-driven modeling, or black-box modeling) or using both the underlying knowledge and the available data (gray-box modeling). This work will discuss soft sensor prediction using data-driven modeling, and for simplicity the term soft sensor will be used from now on to refer to this type of approach.

The traditional development of soft sensors has four main steps: (I) data acquisition and selection of historical data; (II) data pre-processing; (III) model selection, training and validation; (IV) soft sensor maintenance. In the first stage, data is selected for training and evaluation of the model. Then data is submitted to pre-processing (II). The goals of this second stage are the handling of missing data and outliers and to perform an input variable selection when the best variables are unknown [14, 4, 13]. The model selection, training and validation phase (III) requires the correct selection and learning of the model, so that it can correctly reproduce the hard-to-measure variable. The last step is soft sensor maintenance (IV), where the goal is to maintain a good soft sensor response even in the presence of process variations or some data change.

The objective during the variable selection step is to select a reduced subset of variables from the all available variables. Such variable selection is essential to obtain an accurate and reliable reproduction of the target variable as discussed in [5], mainly if the enumeration of candidate variables based on process insight is not feasible [14, 4, 11]. However, this variable selection and consequently the model learning, in the traditional soft sensors development, is done under two main assumptions: first is that the process is stationary (i.e. the distribution which generates the data remains the same over time); and second is that the historical dataset is sufficiently representative, so that the variable selection and model learning can be done using just the information provided by the historical data, and afterwards be deployed in the process.

However, the existence of non-stationary behavior in most industrial plants makes these assumptions above no longer met, and it is further reinforced by the expertise and scientific studies which show that the traditional soft sensor, which is constructed using limited information on historical datasets, starts to degrade with the changes of the process over time [8]. Thus, if the process is non-stationary and the soft sensor is constructed based on traditional methodology it can lead to wrong results regarding the selected variables and the learned model.

Therefore, if the most representative variables are known *a priori*, then the problem becomes limited to the model learning. This problem can be solved by developing a model that can adapt to these changes in order to maintain a correct operation over time. In the soft sensors literature this type of model is referred to as adaptive soft sensor (ASS). In the ASS the model is periodically updated, so that it can represent the current trend of the

process, see [8] for a detailed review about ASS.

However, if the most representative variables are unknown then model learning becomes challenging. The most usual ways to cope with this problem is described as follows.

1. An approach to cope with this problem is to learn the model with all available input variables. This approach has many drawbacks, mainly if the number of inputs is large, which can lead to problem such as overfitting, learning of noise and more [1].

2. It is usual to assume that all the information necessary to select the best variables is provided by the available dataset [12], then the variables are selected using the information of the available dataset only once for all at the beggining of the soft sensor design procedure. However, for the usual situation of non-stationary processes, this can lead to the selection of the wrong variables by the initial variable selection procedure, or, with the evolution of the process operation, the selected variables may become not the best/adequate variables for the prediction setting. This can be disastrous for the soft sensor prediction results.

3. Another approach is to select the best variables and retrain the model periodically, using the most recent samples.

Taking into account previous works, the development of soft sensors in non-stationary environments is conditioned by the knowledge of the best variables, or under the approaches and assumptions previously described. Differently from the previously discussed approaches, this work proposes a recursive method for variable selection and model learning using recursive least squares (RLS) for adaptive soft sensors applications. In the proposed method the importance of each variable is determined online, specifically, a RLS model is created for each available variable to predict the hard-to-measure variable and the correlation coefficient between each input variable and the real target variable is computed from the model parameters and updated with every new sample. Then, a new variable is created by a weighted sum of the outputs of the RLS models for all input variables, where the weights are determined using the correlation coefficient of each variable. As the proposed method is recursive, it does not need to store any past values of data, and it also does not need of any variables scaling on the pre-processing step. Moreover, the proposed method can also be used as a way to interpret the process for process control purposes as in [3], with the advantage that importance of each variable, to predict the target, is given by the well know correlation coefficient. It is important to properly choose the adequate variables to be used in the design of the control architecture.

The proposed method was evaluated by using two prediction problems, an artificial problem and a real one. In the artificial problem, the proposed method and the RLS method were applied in the prediction of an artificially created target; the comparison was then performed using several performance measurements between the predicted and the real targets. Moreover, to verify the sensitivity of the proposed method to irrelevant variables, the method was tested in two problems: with and without the presence of irrelevant variables in the dataset. Afterwards, the proposed method was successfully applied in a real scenario for free lime estimation in a cement kiln process, where the available input data is composed by 130 variables. In constrast with the proposed method, the standard RLS method did not converge in the free lime estimation problem, due to the presence of irrelevant and redundant variables in the dataset.

This paper is organized as follows. Section 3 presents the description of the recursive least squares algorithm. The new adaptive variable selection and model learning algorithm proposed in this paper is presented in Section 4. Section 5 presents experimental results. Finally, Section 6 gives concluding remarks.

2 Notation

The notation used here is as follows, $\mathbf{x}(k) = [x_1(k), \ldots, x_D(k)]^T$ and $y(k)$ are the vector of input variables and the output target at instant k. Moreover, $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_D$, $\mathcal{Y}$, denote the space of input variables values and the space of output values, respectively, where $\mathcal{X} \subset \mathbb{R}^D$ and $\mathcal{Y} \subset \mathbb{R}$.

3 Recursive Least Squares

The simplest linear regression model is composed by a linear combination of the input variables, as follows:

$$y(\mathbf{x}, \boldsymbol{\beta}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_D x_D, \qquad (1)$$

where $\mathbf{x}$ is the input vector and $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_D]^T$ is the vector of model parameters. Given a training set with k examples, denoted by $\mathit{\Gamma} = \{(\mathbf{x}(i), y(i)); i = 1, \ldots, k\}$, the parameters $\boldsymbol{\beta}$ can be found using the least squares estimator:

$$\boldsymbol{\beta} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y}, \qquad (2)$$

where $\mathbf{A}$ is a $k \times (D+1)$ matrix, called design matrix, and $\mathbf{Y}$ is a $k \times 1$ matrix, called output matrix, where:

$$\mathbf{A} = \begin{bmatrix} 1 & \mathbf{x}^T(1) \\ \vdots & \vdots \\ 1 & \mathbf{x}^T(k) \end{bmatrix}; \quad \mathbf{Y} = \begin{bmatrix} y(1) \\ \vdots \\ y(k) \end{bmatrix} \qquad (3)$$

However, instead of using the closed form (2), an incremental learning method [6] can be employed to find the weights vector $\boldsymbol{\beta}$ as follows. When a new sample $(\mathbf{a}(k+1), y(k+1))$ is available, where $\mathbf{a}(k+1) = [1, \mathbf{x}(k+1)]^T$, the weight vector $\boldsymbol{\beta}$ can be incrementally updated as follows:

$$\boldsymbol{\beta}(k+1) = (\mathbf{A}_{k+1}^T\mathbf{A}_{k+1})^{-1}(\mathbf{A}_{k+1})^T\mathbf{Y}_{k+1}, \qquad (4)$$

$$\boldsymbol{\beta}(k+1)=\left(\begin{bmatrix}\mathbf{A}_k\\ \mathbf{a}^T(k+1)\end{bmatrix}^T\begin{bmatrix}\mathbf{A}_k\\ \mathbf{a}^T(k+1)\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{A}_k\\ \mathbf{a}^T(k+1)\end{bmatrix}^T\begin{bmatrix}\mathbf{Y}_k\\ y(k+1)\end{bmatrix}. \tag{5}$$

Reorganizing equation (5) the new weight matrix $\boldsymbol{\beta}(k+1)$ becomes:

$$\boldsymbol{\beta}(k+1)=\boldsymbol{\beta}(k)+\mathbf{P}_{k+1}\mathbf{a}^T(k+1)\left(y(k+1)-\mathbf{a}^T(k+1)\boldsymbol{\beta}(k)\right), \tag{6}$$

where $\mathbf{P}_{k+1}=\left(\mathbf{A}_{k+1}^T\mathbf{A}_{k+1}\right)^{-1}$ is given by:

$$\mathbf{P}_{k+1}=\mathbf{P}_k-\frac{\mathbf{P}_k\mathbf{a}(k+1)\mathbf{a}^T(k+1)\mathbf{P}_k}{1+\mathbf{a}^T(k+1)\mathbf{P}_k\mathbf{a}(k+1)}. \tag{7}$$

However, equation (7) goes to zero as the number of samples increases [6], and also the adaptation gain $\mathbf{P}_{k+1}\mathbf{a}^T(k+1)$ in (6) decreases to zero when the number of samples goes to infinite, so that after several iterations, the new samples have little contribution to the model. Thus, this approach is not suitable for learning non-stationary systems.

Therefore, in non-stationary systems it is necessary to introduce a forgetting factor λ in the examples so that the model could take into consideration recent data. This can be done by adding a forgetting factor λ in the recursive least squares estimator, which takes the following form:

$$\boldsymbol{\beta}=(\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{Y}, \tag{8}$$

where $\mathbf{W}=\text{diag}(\lambda^{k-1},\lambda^{k-2},\ldots,1)$, and $0<\lambda\leq 1$, so that the smaller the λ parameter, the more the recent data is weighted, and the more the RLS estimator can track the time-varying parameters, more specfically, according with [2] the effective lenght of data, (i.e the number of observations being used) is equal to $(1-\lambda)^{-1}$. Rewriting equation (8) in a forma similar to (5), the following update rule for $\mathbf{P}_{k+1}$ is obtained:

$$\mathbf{P}_{k+1}=\frac{1}{\lambda}\left(\mathbf{P}_k-\frac{\mathbf{P}_k\mathbf{a}(k+1)\mathbf{a}^T(k+1)\mathbf{P}_k}{\lambda+\mathbf{a}^T(k+1)\mathbf{P}_k\mathbf{a}(k+1)}\right), \tag{9}$$

4 Proposed Method

The objective of the proposed method is to select the variables and learn the model recursively, without the necessity of storing the input data. Moreover, the proposed method does not need any knowledge about the process to be modeled. For future reference the method will be called as adaptive feature selection and model learning using RLS modeling (AdaFSML-RLS). It is described below.

A linear model $f_j(x_j|\boldsymbol{\beta}^{(j)}):\mathcal{X}_j\rightarrow\mathcal{Y}$ in the form of (1) will be constructed for each available input variable $x_j(k)$, to predict the target $y(k)$, where $\boldsymbol{\beta}^{(j)}$ is the vector of parameters of model j, for $j=1,\ldots,D$. Thus, D different models will be constructed. Whenever a new input-output pair becomes available, the D models will be accordingly updated using equations (6) and (9), with a forgetting factor λ equal for all models. When a new sample $k+1$ is used to update the model, function $f_j(x_j|\boldsymbol{\beta}^{(j)})$, $j=1,\ldots,D$, takes the following form:

$$f_j(x_j|\boldsymbol{\beta}^{(j)}(k+1))=\beta_0^{(j)}(k+1)+\beta_1^{(j)}(k+1)x_j, \tag{10}$$

where similarly to (1) with $D=1$, $\boldsymbol{\beta}^{(j)}(k+1)=[\beta_0^{(j)}(k+1),\beta_1^{(j)}(k+1)]^T$.

Using this linear model (10) updated with sample $k+1$, it is possible to measure the importance that each variable x_j has to the prediction setting through the use of the correlation coefficient between x_j and y. The correlation coefficient measures the degree of correlation among two random variables, based on the quality of a linear adjustment of the data. It takes values between -1 and 1, where $\rho=1$ corresponds to a positive perfect correlation among the two variables, $\rho=-1$ corresponds to a perfect negative correlation among the two variables (i.e. if one increases, the other decreases), and $\rho=0$ means that the two variables are linearly independent. According to [9, Chapter 5] in a linear regression of the form (10), and taking into account the samples received up to instant $(k+1)$, the correlation coefficient, $\rho_{x_j,y}(k+1)$, between x_j and y has the following form:

$$\rho_{x_j,y}(k+1)=\left(\frac{\sigma_j^2(k+1)}{\sigma_y^2(k+1)}\right)^{-1/2}\beta_1(k+1), \tag{11}$$

where $\sigma_j^2(k+1)$ is the variance of x_j, and $\sigma_y^2(k+1)$ is the variance of y.

It is necessary to define formulas to update the variance of x_j and y when sample $k+1$ becomes available. In this work the following update equations for the variance, σ_j^2, and the mean, $\overline{\mu}_j$, defined in [10] were used, where a forgetting factor λ is taken into consideration in the update formulas:

$$\overline{\mu}_j(k+1)=\lambda\overline{\mu}_j(k)+(1-\lambda)x_j(k+1), \tag{12}$$

$$\begin{aligned}\sigma_j^2(k+1)=&\lambda\left(\sigma_j^2(k)+\left(\overline{\mu}_j(k+1)-\overline{\mu}_j(k)\right)^2\right)\\&+(1-\lambda)\left(x_j(k+1)-\overline{\mu}_j(k+1)\right)^2,\end{aligned} \tag{13}$$

where the variance of y is computed using the same update formulas above, then considering, in this particular case, $j=y$.

Using the mappings $f_j(x_j(k+1)|\boldsymbol{\beta}^{(j)}(k))$ of the input variables $x_j(k+1)$, $j=1,\ldots,D$, into the output space $\mathcal{Y}$, a new variable $\hat{x}(k+1)$ is created as follows:

$$\hat{x}(k+1)=\frac{\sum_{j=1}^{D}\left[\alpha_j(k+1)f_j\left(x_j(k+1)|\boldsymbol{\beta}^{(j)}(k)\right)\right]}{\sum_{j=1}^{D}\alpha_j(k+1)}, \tag{14}$$

where $\alpha_j(k+1)$ is is given by:

$$\alpha_j(k+1)=\rho_{x_j,y}^2(k+1). \tag{15}$$

$\hat{x}(k+1)$ can be seen as an ensemble of all the models $f_j(x_j(k+1)|\boldsymbol{\beta}^{(j)}(k))$, $j=1,\ldots,D$, where the input of model j is the input variable $x_j(k+1)$, and the contribution of each model to the ensemble is determined by the importance (15) of each variable at instant $k+1$. As can

be noticed, the closer $\alpha_j(k+1)$ gets to 1, the less important is variable x_j to the prediction of the target, y, and the less it contributes in (14) to calculate variable $\hat{x}(k+1)$.

Using the new variable $\hat{x}(k+1)$ a new RLS model $f(\hat{x}|\hat{\boldsymbol{\beta}}) : \mathcal{X} \rightarrow \mathcal{Y}$ is learned and updated iteratively. Whenever a new sample $\hat{x}(k+1)$ becomes available, $f(\hat{x}|\hat{\boldsymbol{\beta}})$ is learned and updated using the same methodology as the one that is used to learn and update the individual models $f_j(x_j|\boldsymbol{\beta}^{(j)})$; And also the same value of λ is used to update $f(\hat{x}|\hat{\boldsymbol{\beta}})$ and the individual models $f_j(x_j|\boldsymbol{\beta}^{(j)})$, $j = 1, \ldots, D$. Then, the final model is given by:

$$\hat{y}(n) = f\left(\hat{x}(n)|\hat{\boldsymbol{\beta}}(k)\right), \tag{16}$$

where $\hat{y}(n)$ is the estimated output given input sample $\mathbf{x}(n)$.

5 Experimental Results

In this section the proposed AdaFSML-RLS method is tested in one artificial dataset and in a real-world dataset. The artificial dataset is time dependent and has a linear input-output relationship, with a set of relevant and non-relevant variables in the pool of input variables. Moreover, an experiment has been performed to verify the sensitivity of the algorithm with respect to number of irrelevant variables.

To measure the performance of prediction setting the correlation coefficient (CC), the normalized mean square error (NMSE), and the root mean square error (RMSE), between the real and predicted outputs were used. The proposed adaptive AdaFSML-RLS method will be compared with the classic RLS model method.

5.1 Artificial Dataset

The output y of the artificial model is defined as follows:

$$y(k) = \begin{cases} 10(\pi x_1(k) + x_2(k)) - 20(x_3(k) - 0.5) \\ +10x_4(k) + 5x_5(k) + \mathcal{N}(0,1), \\ \text{if } k \leq 500, \\ \\ -10(\pi x_1(k) + x_2(k)) + 20(x_3(k) - 0.5) \\ -10x_4(k) - 5x_5(k) + 10x_6(k) + \mathcal{N}(0,1), \\ \text{if } 500 < k \leq 1000, \\ \\ 5(\pi x_1(k) + x_2(k)) + 8(x_3(k) - 0.5) + \mathcal{N}(0,1), \\ \text{if } 1000 < k \leq 1500, \\ \\ -10x_4 - 5x_5(k) + 10x_6(k) + \mathcal{N}(0,1), \\ \text{if } 1500 < k \leq 2000, \end{cases} \tag{17}$$

were all variables were generated independently of each other and uniformly distributed over $[0, 1]$ and $\mathcal{N}(0, 1)$ is a Gaussian noise with zero mean and unit variance. Analyzing equation (17) it can be noticed that the first 500 samples of output is a linear combination of the first five variables $(x_1, x_2, x_3, x_4, x_5)$, while for the samples $500 < k \leq 1000$ the output is composed by a linear combination of the same variables but with different weights plus another variable x_6. For the the samples $1000 < k \leq 1500$ the variables which compose the model are (x_1, x_2, x_3), while for the remaining of the samples $1500 < k \leq 2000$ the variables (x_4, x_5, x_6) are composing the model, Figure 1 shows the output y for all samples.

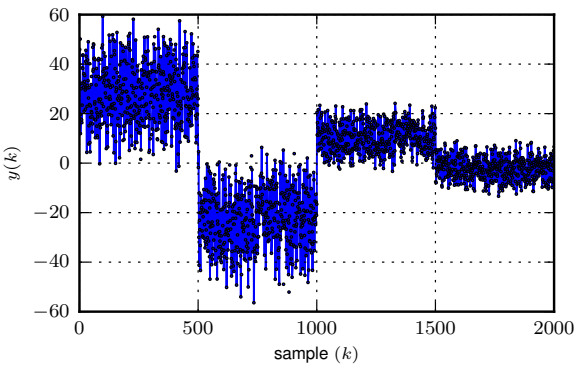


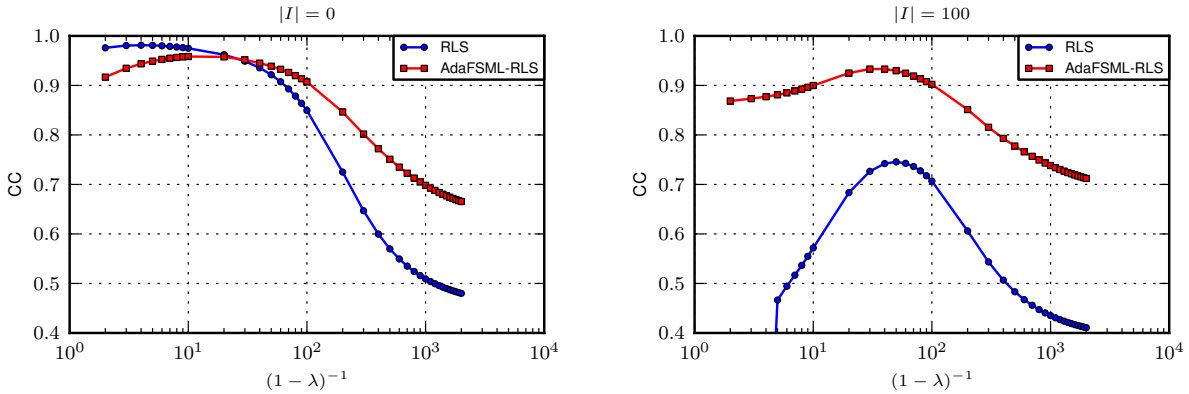
Figure 1: Output y defined in equation (17).

The objective while learning the equation (17) using the AdaFSML-RLS and the RLS models is to predict the next example $y(k+1)$ using the input $x(k+1)$ and parameters updated up to the k sample, and then after the predicting $y(k+1)$, both models (AdaFSML-RLS and RLS) are updated using the pair $(\mathbf{x}(k+1), y(k+1))$.

In the first experiment, the RLS model and the AdaFSML-RLS were learned using variables $(x_1, x_2, x_3, x_4, x_5, x_6)$, for different values of the forgetting factor λ, where for the $x-$axis is used the effective number of data being used, i.e $(1-\lambda)^{-1}$. Figure 2a shows the curve indicating the correlation coefficient (CC) value between the target and the predicted output for different values of λ, where the $|I| = 0$ indicates the number of irrelevant variables present in the set of training, and in this first experiment is zero.

In the second experiment, the RLS model and the AdaFSML-RLS were learned using variables $(x_1, x_2, x_3, x_4, x_5, x_6)$ and more 100 irrelevant variables were added to the set, for different values of the forgetting factor λ. The irrelevant variables were generated independently of each other and uniformly distributed over $[0, 1]$. Figure 2b shows the curve indicating the correlation coefficient (CC) value between the target and the predicted output for different values of λ.

As can be noticed, when the correct variables are presented to the RLS and AdaFSML-RLS models, the RLS shows better performance than the AdaFSML-RLS for some of λ values, in terms of CC. However, the best CC value found for the RLS and AdaFSML-RLS are close: The CC reached 0.98 and 0.96 for the the RLS and AdaFSML-RLS, respectively. Another point which should be highlighted in Experiment 1 is that the RLS can handle well irrelevant variables. In fact, for some samples some variables become irrelevant to the model, as for example between the sample $1500 < k \leq 2000$ the variables



(a) Result on the artificial dataset for the RLS and AdaFSML-RLS models, without irrelevant variables $|I| = 0$.

(b) Results on the artificial dataset for the RLS and AdaFSML-RLS models, in the presence of irrelevant variables $|I| = 100$.

Figure 2: Correlation coefficient between the predicted and real output on the artificial dataset, with and without irrelevant variables, for different values of λ. The red line indicates the proposed method, while the blue line indicates the RLS model.

Table 1: Summary of the results of the RLS and AdaFSML-RLS models for the best value of λ on the artificial dataset, with and without irrelevant variables.

	$\|I\| = 0$		$\|I\| = 100$	
	RLS	AdaFSML-RLS	RLS	AdaFSML-RLS
RMSE	4.01	5.91	15.0	7.48
CC	0.98	0.96	0.75	0.93
NMSE	0.04	0.08	0.53	0.13
λ	0.75	0.90	0.98	0.97

(x_1, x_2, x_3) are not part of the model, but the model still provides good results.

However, when the number of irrelevant variables is high, as in the second experiment, the RLS model exhibits a lower prediction capability when compared with AdaFSML-RLS. As shown in Figure 2b the best RLS performance is a value of CC of 0.75. On the other hand, the AdaFSML-RLS shows more robustness to irrelevant variables, which makes the AdaFSML-RLS model a good choice in the modeling of unknown systems, the best performance of the AdaFSML-RLS in terms of CC is 0.93.

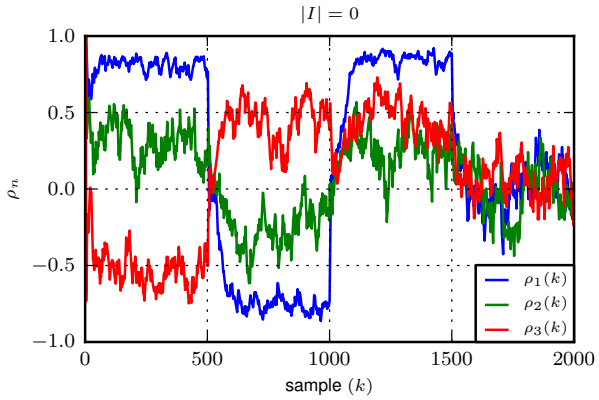
Table 1 presents the summary of the results measured by the CC, NMSE, and RMSE performance indicators for the value of λ which gave the best results of prediction. The results reinforce the previous discussion which indicated that the AdaFSML-RLS model is more robust when irrelevant variables are present in the learning dataset, while the RLS model decreases its performance in such case. However, when there is no presence of irrelevant variables in the dataset the RLS model outperforms the AdaFSML-RLS model in all indicators, but the AdaFSML-RLS still has good prediction results.

Another advantage of the AdaFSML-RLS method is that it can evaluate directly and in real-time the correlation coefficient between each input variable and the target. Figures 3a and 3b show the importance of each variable for the case where there are no irrelevant variables in the dataset. Figure 3a indicates the importance of variables x_1, x_2 and x_3 over the time. From this figure it is possible to note the variation of the correlation coefficient of x_1 with respect to the target. In the first 500 samples it shows a positive correlation coefficient of approximately 0.8. From (17) its contribution of the output in this period is given by 10π. Between samples 500 and 1000 it has shown a negative correlation coefficient of approximately -0.8 while, consistently, its contribution in equation (17) between these samples is of -10π. Between samples 1000 and 1500 the contribution of x_1 is reduced to a positive value of 5π and it has show a positive correlation coefficient of about 0.6. For the remaining of the samples the contribution of x_1 to the model is null, and its correlation coefficient becomes around the 0 in Figure 3b for these samples. Thus, it is possible to conclude, through the analysis of the correlation coefficient of variable x_1 with respect to the target over the time in Figure 3b that the AdaFSML-RLS method can be used to track the correlation coefficient (i.e. the importance of variable) between the input variables and the target in real time successfully.

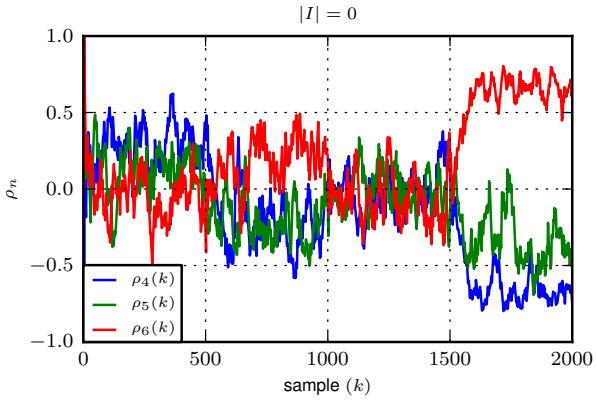
Making a similar analysis to Figures 3c, 3d taking into acocunt the time-evolution of the correlation coefficient values, and equation (17) that was used to generate the dataset, it can be concluded that the importance of each variable, as measured by the CC, follows the changes expected according to the generator equation. The given correlation coefficient by the AdaFSML-RLS method to irrelevant variables can be seen in Figure 3e and it could be noted that its values are around 0.

5.2 Free Lime Estimation

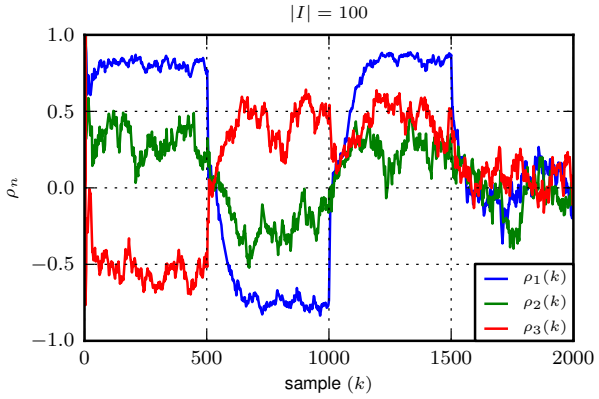
This section presents experimental results of a case study concerning the free lime estimation in a real cement kiln plant. The free lime is one of the most important qual-



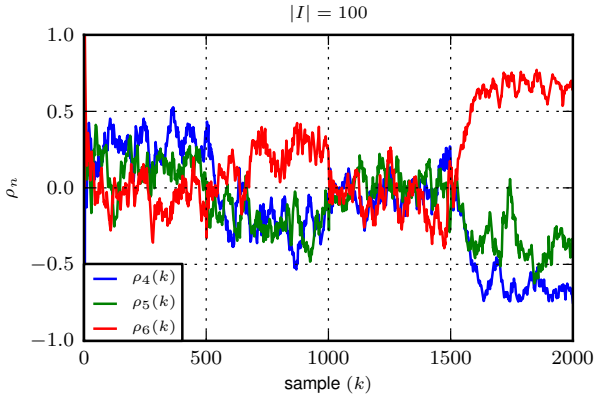
(a) CC for variables x_1, x_2, x_3 over the time with $\lambda = 0.90$ on the artificial dataset, without irrelavant variables $|I| = 0$.



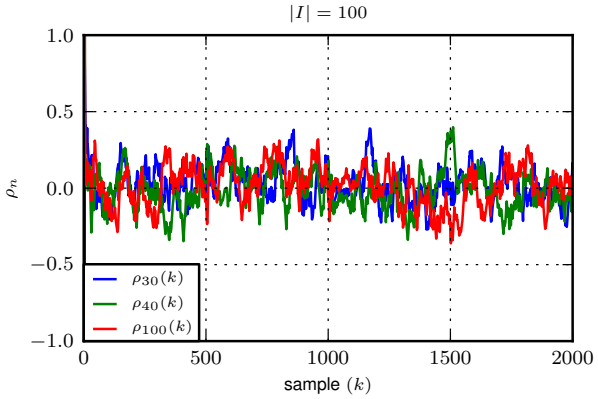
(b) CC for variables x_4, x_5, x_6 over the time with $\lambda = 0.90$ on the artificial dataset, without irrelavant variables $|I| = 0$.



(c) Correlation cofficient for variables x_1, x_2, x_3 over the time with $\lambda = 0.97$ on the artificial dataset, with irrelavant variables $|I| = 100$.



(d) CC for variables x_4, x_5, x_6 over the time with $\lambda = 0.97$ on the artificial dataset, without irrelavant variables $|I| = 100$.



(e) CC for variables x_{30}, x_{40}, x_{100} (irrelevant variables) over the time with $\lambda = 0.97$ on the artificial dataset, without irrelavant variables $|I| = 0$.

Figure 3: Correlation coefficient between the input variables and the target over the time in the artificial dataset, with and without irrelevant variables, measured using the AdaFSML-RLS model.

ity parameter to be monitored and controlled in a cement kiln plant, and it is normally obtained by laboratory. In this case study, free lime is measured by laboratory analysis at approximately every 15 minutes. For this study, 130 variables associated with the cement kiln were acquired and used to build the model.

To predict the free lime only the AdaFSML-RLS will be applied, because the RLS did not converge due the quantity of redundant and irrelevant variables. However, as the AdaFSML-RLS method is robust to the irrelevant and redundant variables, it is a suitable model to be applied in the modeling of free lime.

Table 2: Summary of the results of AdaFSML-RLS model for free-lime prediction.

	AdaFSML-RLS
RMSE	0.34
CC	0.80
NMSE	0.39

The dataset used for this application is composed of 43469 samples, collected with a sampling interval of 1 minute for all variables. However, as the free lime is collected only at every 15 minutes, the update of the model will occur only at every 15 samples, approximately. Then, the provided soft sensor will predict the value of the free lime at time instants where it is not available, i.e. during the intervals of approximately 15 minutes between laboratory analysis. For evaluation purposes, if a new valid input-output pair $(\mathbf{x}(n), y(n))$ is available for update, then the output $y(n)$ will be first predicted using the input $\mathbf{x}(n)$, and then the model parameters will be updated. There is more, if the number of valid samples for update is less than the number of total samples to predict, then the real-time correlation between each input variable and the output, equation (11), will be updated just when a new valid sample become available.

Many variables in this data-set suffers from outliers and measurement errors, as an example, Figures 4a and 4b show the time-evolution of variables x_{40} and x_{123} on the Free-Lime dataset, it can be noted that around the $5000, 14000, 15000, 27000$th samples and between around the $37000-38000$ samples these variables have the presence of outliers. Even having this knowledge about it, the method was applied in this data-set without any pre-processing, propositally, to simulate the application of the method in unknown enviroments ans assuming no knowledge about the process or the variables. A forgetting factor λ of 0.90 was chosen. The performance values for the free lime prediction using the AdaFSML-RLS model are indicated in Table 2

A value of correlation of 0.80 was attained between the predicted output and the target output. The prediction is exhibited in Figure 5, validating and showing the effectiveness of the proposed method to perform prediction in unknown environments. However, some spikes can be noted in the prediction phase, which can indicate the presence of outliers in the most important input variables. Therefore, the presence of these spykes are pontual and can be alleviated by application of a mean average filter in the output prediction or it can be removed by applying an online outlier detection and respective treatment of the input data, during the application of the proposed algorithm.

6 Conclusions

This paper proposed a method for adaptive variable/feature selection and model learning (AdaFSML-RLS) for soft sensors applications in industrial processes for linear modeling using the recursive least squares learning. Furthermore, the proposed method has the capability of tracking the real time correlation coefficient between each variable and the target, allowing the knowledge about the importance of variables over the time, which can be useful while design control systems.

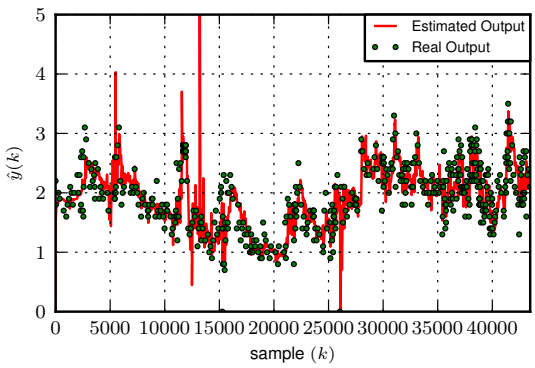


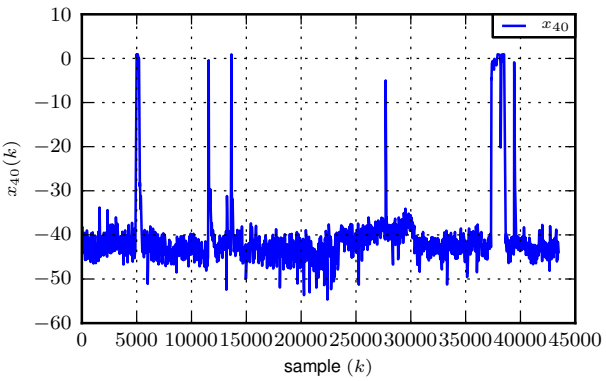
Figure 5: Free lime prediction. The sampling interval is 1 [minute].

In two experiments one in an artificial dataset and the other with a real-world dataset, the proposed AdaFSML-RLS method has been shown to be feasible, effective and robust under irrelevant variables. This constrasted with the RLS algorithm, which suffers in terms of prediction performance when irrelevant variables are included in the dataset. On the artificial dataset, the AdaFSML-RLS has shown to have close values to the RLS algorithm when there are no irrelevant variables present to the model. In the presence of irrelevant variables, the RLS decreases its capability of prediction, while the AdaFSML-RLS method maintains a good response in terms of prediction performance. Moreover, the real time correlation coefficient measured by the proposed method works properly and it can give a real-time insight about the importance of each process variable.

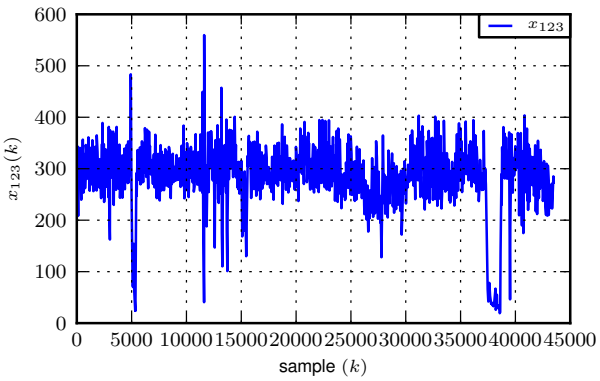
In the real application, the free lime estimation, only the AdaFSML-RLS was applied, and it was applied without any knowledge about the process or variables. The total number of variables was 130, so that the RLS model could not be applied, the proposed method reached good prediction results, allowing its application in real-time. The estimation of free-lime in the cement kiln problem, at each 1 minute, by the proposed method, will allows the operators to take properly actions when necessary, without the necessity of waiting to the laboratorial analysis. In future research the problem of the real time control of the free-lime will be taken.

Acknowledgment

This work was supported by Mais Centro Operacional Program, financed by European Regional Development



(a) Example of variable x_{40} on Free Lime dataset.



(b) Example of variable x_{123} on Free Lime dataset.

Figure 4: Examples of variables x_{40} and x_{123} on the Free Lime dataset.

Fund (ERDF), and Agência de Inovação (AdI) under Project SInCACI/3120/2009. Francisco Souza have been supported by Fundaçâo para a Ciência e a Tecnologia (FCT) under grant SFRH/BD/63454/2009.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1 edition, 2006.

[2] B. S. Dayal and J. F. MacGregor. Recursive exponentially weighted pls and its applications to adaptive control and prediction. *Journal of Process Control*, 7(3):169–179, 1997.

[3] M. Eastwood and P. Kadlec. Interpretable, online soft-sensors for process control. In *Proc. 2011 International Conference on Data Mining Workshops*, pages 581–587, December 2011.

[4] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia. *Soft Sensors for Monitoring and Control of Industrial Processes*. Advances in Industrial Control. Springer, 1 edition, December 2006.

[5] I. Guyon. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[6] J.-S. R. Jang, C.-T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, 1 edition, September 1997.

[7] P. Kadlec, B. Gabrys, and S. Strandt. Data-driven soft sensors in the process industry. *Computers & Chemical Engineering*, 33(4):795–814, 2009.

[8] P. Kadlec, R. Grbic, and B. Gabrys. Review of adaptation mechanisms for data-driven soft sensors. *Computers & Chemical Engineering*, 35(1):1–24, 2011.

[9] J. F. Kenney and E. S. Keeping. *Mathematics of Statistics Part One*. D. Van Nostrand Company Inc, 3 edition, 1962.

[10] W. Li, H. Yue, S. Valle-Cervantes, and S. Qin. Recursive pca for adaptive process monitoring. *Journal of Process Control*, 10(5):471 – 486, 2000.

[11] O. Ludwig, U. Nunes, R. Araújo, L. Schnitman, and H. A. Lepikson. Applications of information theory, genetic algorithms, and neural models to predict oil flow. *Communications in Nonlinear Science and Numerical Simulation*, 17(7):2870–2885, 2009.

[12] M.-D. Ma, J.-W. Ko, S.-J. Wang, M.-F. Wu, S.-S. Jang, S.-S. Shieh, and D. S.-H. Wong. Development of adaptive soft sensor based on statistical identification of key variables. *Control Engineering Practice*, 17(9):1026–1034, 2009.

[13] F. Souza and R. Araújo. Variable and time-lag selection using empirical data. In *Proc. 2011 IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2011)*, pages 1–8, September 2011.

[14] K. Warne, G. Prasad, S. Rezvani, and L. Maguire. Statistical and computational intelligence techniques for inferential model development: a comparative evaluation and a novel proposition for fusion. *Engineering Applications of Artificial Intelligence*, 17(8):871–885, 2004.