MATEC Web of Conferences **252**, 03012 (2019) https://doi.org/10.1051/matecconf/201925203012
CMES'18

Impact of feature selection on system identification by means of NARX-SVM

Michał Awtoniuk[1,*], *Marcin* Daniun[1], *Kinga* Sałat[2], and *Robert* Sałat[1]

[1]Warsaw University of Life Sciences, Faculty of Production Engineering, 164 Nowoursynowska, 02-787 Warsaw, Poland
[2]Jagiellonian University Medical College, Faculty of Pharmacy, 9 Medyczna, 30-688 Kraków, Poland

Abstract. Support Vector Machines (SVM) are widely used in many fields of science, including system identification. The selection of feature vector plays a crucial role in SVM-based model building process. In this paper, we investigate the influence of the selection of feature vector on model's quality. We have built an SVM model with a non-linear ARX (NARX) structure. The modelled system had a SISO structure, i.e. one input signal and one output signal. The output signal was temperature, which was controlled by a Peltier module. The supply voltage of the Peltier module was the input signal. The system had a non-linear characteristic. We have evaluated the model's quality by the *fit* index. The classical feature selection of SVM with NARX structure comes down to a choice of the length of the regressor vector. For SISO models, this vector is determined by two parameters: n_u and n_y. These parameters determine the number of past samples of input and output signals of the system used to form the vector of regressors. In the present research we have tested two methods of building the vector of regressors, one classic and one using custom regressors. The results show that the vector of regressors obtained by the classical method can be shortened while maintaining the acceptable quality of the model. By using custom regressors, the feature vector of SVM can be reduced, which means also the reduction in calculation time.

1 Introduction

A general definition describes a system as a set of elements separated from the environment together with their interactions. In every system, input and output signals can be distinguished. The relationship between these signals can be described mathematically and is called a model. In order to reveal the model, it is necessary to use the system identification.

There are two approaches in dynamic system modelling: first-principles modelling and data-driven modelling. The first approach is based on the laws of material, momentum and energy balances. These models are also called the white-box. The accuracy of these types of models is usually very high, but they require experience in selection of a high number of parameters.

An alternative to first-principles models are data-driven models (also called a system identification). Data-driven models are not based on a priori knowledge but on measured experimental data. This method is much more practical due to the fact that most systems (e.g. industrial processes) are too complex to be understood at a fundamental level. The most important advantages of data-driven models are as follows:

- modelling possibility without in-depth knowledge of the system,
- flexibility in the choice of a model's structure,
- convenience of their implementation in the form of a computer code.

These advantages make the system identification an integral part of modern control engineering.

On the other hand, the most serious disadvantages of data-driven models are:

- they require to conduct an identification experiment. The experiment has to be properly planned so that the data collected enabled a full description of the analysed system,
- these models are non-transparent, i.e. they do not reflect the interior physical mechanisms of the process.

Due to the lack of transparency data-driven models are termed as black-box models. In the recent years, numerous scientists [1,2] suggested using the method of artificial intelligence in the black-box identification. This suggestion resulted in numerous works in which different techniques were presented, such as artificial neural networks (ANN), fuzzy-neural networks (FNN), wavelets, [3,4]. Among the neural networks used for black-box identification, the most frequently reported networks are the multi-layer perceptron (MLP), fuzzy-neural networks (FNN), radial basis function (RBF), Runge-Kutta (RK) method, and digital recurrent network (DRN). Few authors made use of support vector machines (SVM) [1,5] as an identification tool. Usually, the authors also ignore the rules of choosing regressors and parameters of both ANN and SVM.

In the case of the NARX-SVM black-box model, selection from regressors are the feature selection problem. The problem with the choice of the feature

[*]Corresponding author: michal_awtoniuk@sggw.pl

vector is particularly important due to the fact that not all features always carry the significant information. Although choosing the optimal set of features is a difficult task, its solution brings many benefits that manifest themselves as:

- reducing the dataset to avoid curse of dimensionality,
- better estimation accuracy,
- application of more effective learning techniques,
- it helps to improve the network generalisation capability by removing insignificant features interfering with each other.

The goal of this study is to analyse how the model quality will change after modification of the SVM feature vector. The modification will be based on the use of custom regressors.

2 Methods for regressor selection

There are many classes of identification methods for non-linear objects. One of them is a quite commonly used methods based on recursive input-output models, and using different types of parameterised nonlinear expansions, such as ANN and SVM. In particular, non-linear autoregressive with exogenous input (NARX) models are used in identification. It has been proven in numerous studies [6] that non-linear objects can be modelled with high accuracy using the NARX structure. However, if the structure of the model is not known a priori, the number of regression candidates for NARX structure can be very large. There are several techniques for selecting regressor candidates for NARX-SVM models. In general, they are based on forward/backward regression techniques which progressively increase the structure of the model using appropriate indicators. The most popular ones are based on the Prediction Error Minimisation (PEM) techniques: the Forward-Regression Orthogonal Estimator (FROE) and the Fast Recursive Algorithm (FRA). An alternative to PEM methods are The Simulation Error Minimisation (SEM) methods together with its modification of The Simulation Error Minimisation with Pruning (SEMP) which increase in robustness model interference and have small requirements for the input excitation. The disadvantages of these methods are the high requirements for computational power and the time needed to obtain the results.

In our research we used the methods of regressor selection described in [5,7] together with the modifications shown in this paper.

3 Materials and methods

Fig. 1 presents the picture of the main components of the plant. The key elements that we have used in our research are:

- data acquisition (DAQ) board National Instruments NI PCI-6226,
- power bridge Wobit SDD187,
- Peltier module TEC1-12706,
- aluminium plate 54x85x6 mm,
- temperature sensor Pt100,
- air cooler with aluminium heat sink, six ϕ6 mm heat-pipes, and 140 mm fan,
- Matlab&Simulink software.

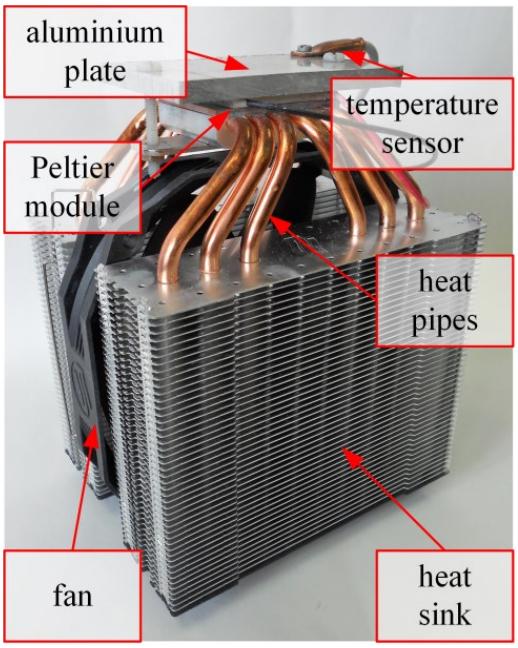

Fig. 1. Picture of the main components of the plant.

The main task of the system is to maintain the temperature of the aluminium plate within the range of 0-50°C. The Peltier module is an actuator. In order to enable a continuous control of the module, we have used a power bridge. It converts the control signal u generated by the analogue output of a DAQ board with a range of -10...+10 VDC into a PWM signal controlling the supply voltage of the Peltier module in the range of 0...15 VDC. The signal u is the system input. The power bridge also allows polarity to be changed so that the aluminium plate is heated or cooled as required. In order to enable effective cooling of the aluminium plate, we have used an air cooler to transfer the heat from the warm side of the Peltier module. The output signal of the system, denoted as y, is the temperature of the aluminium plate.

The system under investigation is an example of a non-linear object. This is confirmed by the static characteristics (Fig. 2). The characteristics can be divided into two parts at the point u = 0. The static gain of the system is lower in the first part of the characteristics than in the second part. Negative values of u mean cooling the aluminium plate, while positive values mean heating.

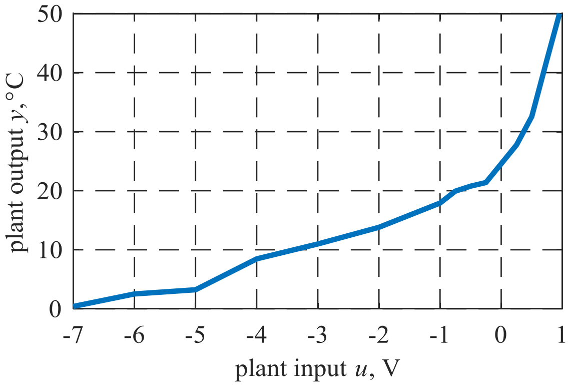

Fig. 2. Static characteristics of the plant.

To model the system, we have built an SVM model with the NARX structure. In the case of NARX-SVM, the output of the model depends on past inputs and outputs (features input set) and can be described as follows:

$$\hat{y}(t) = f([u(t-1),\ldots \qquad \ldots$$
$$\hat{y}(t) = f\left(\boldsymbol{\varphi}(t),\boldsymbol{\theta}\right) \tag{1}$$

where:

- $\hat{y}(t)$ is the output of the model (NARX-SVM),
- $f(\cdot)$ is a non-linear function, in the case of SVM it is the RBF kernel function,
- u is the input of the plant at time t-1,…,t-n_u,
- y is the plant output at time t-1,...,t-n_y,
- n_u and n_y indicate the order of the NARX model (number of lags),
- $\boldsymbol{\theta}$ is the parameter vector,
- $\boldsymbol{\varphi}$ is the regression vector.

The general block diagram of the set feature vector covering both the input and output is shown in Fig. 3. The learning stage was conducted in a prediction mode, while validation was made in both prediction and simulation modes. We have optimised the SVM parameters (i.e. C and γ) according to the procedure described in detail in [7].

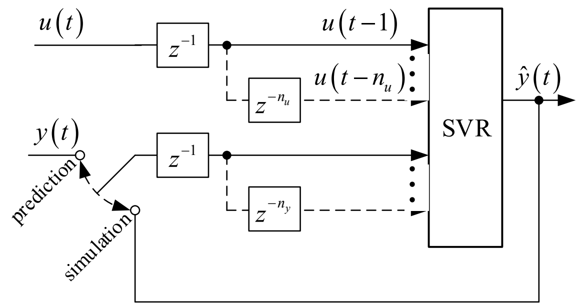

Fig. 3. General block diagram of the input vector and the output of the SVM in a prediction or simulation modes.

The evaluation of model fitness to data measured was performed using the *fit* index, calculated pursuant to the formula (a higher value means a better model quality)

$$fit = \left(1 - \frac{\left\| y(t) - \hat{y}(t) \right\|_2}{\left\| y(t) - \bar{y} \right\|_2}\right) \cdot 100\% \tag{2}$$

where:

- $y(t)$ is the measured plant output,
- $\hat{y}(t)$ is the simulated plant output,
- $\bar{y}$ is the mean of the measured plant output.

4 Data collection

The collecting of data was carried out according to the procedure described previously [5]. During the experiment we were collecting data according to the block diagram shown in Fig 4.

The input variable u and the output variable y of the plant have been administered and recorded using a PC with Matlab and Simulink (MathWorks) software. In the Matlab and Simulink toolbox, the real-time windows target was used. The connection between the plant and the PC was maintained through a DAQ board. The frequency for recording the samples in Matlab and Simulink was 10 Hz. The input signal u was administered in the form of uniformly distributed random signals of a different amplitude and frequency in the interval of -10…+10 VDC. We divided the data into learning (first 2000 samples) and validation (last 3000 samples) datasets. The datasets collected for our study are shown in Fig. 5.

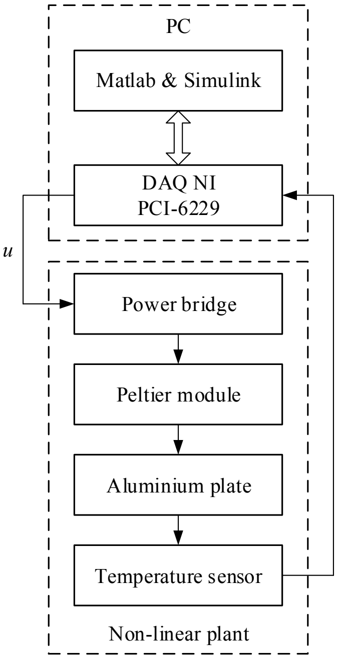

Fig. 4. Block diagram of the dataset collection.

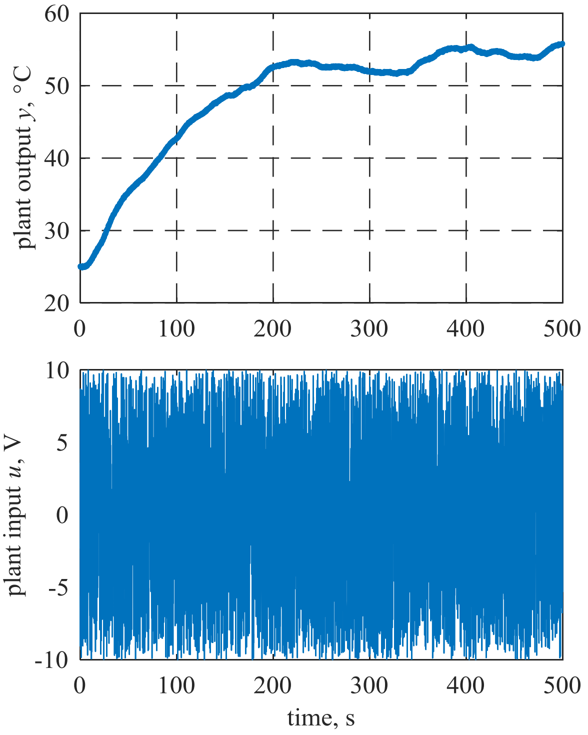

Fig. 5. Learning (first 200 seconds) and validation datasets (last 300 seconds).

5 Results

We have investigated NARX models with different number of regressors (number of lags) assuming that $n_u = n_y \in \{1,\ldots$. It is a classical method of building the regressor vector. Fig. 6 shows the value of *fit* index for all investigated models. The *fit* index was calculated for the validation dataset in a simulation mode.

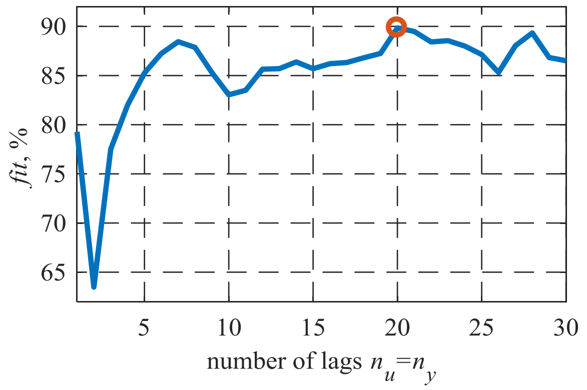

Fig. 6. Impact of the number of lags on model quality (validation dataset in the simulation mode).

The best model (red circle), denoted as *M*0, had number of lags equal to $n_u = n_y = 20$. Therefore, the vector of regressors, i.e. the SVM feature vector, consisted of 40 elements and can be expressed as

$$\boldsymbol{\varphi}(t) = \begin{bmatrix} u(t-1), u(t-2), \ldots & 9), u(t-20), \\ y(t-1), y(t-2), \ldots & 9), y(t-20) \end{bmatrix} \quad (3)$$

The parameters of SVM were $nSVM = 1707$, $C = 50$, $\gamma = 0.01$.Table 1 presents the value of the *fit* index for learning and validation datasets, while Fig. 7 shows the model response for both datasets.

Table 1. Value of the *fit* index for the *M*0 model.

Model	*fit* [%]		
	learning (prediction)	validation (prediction)	validation (simulation)
*M*0	99.74	90.90	89.84

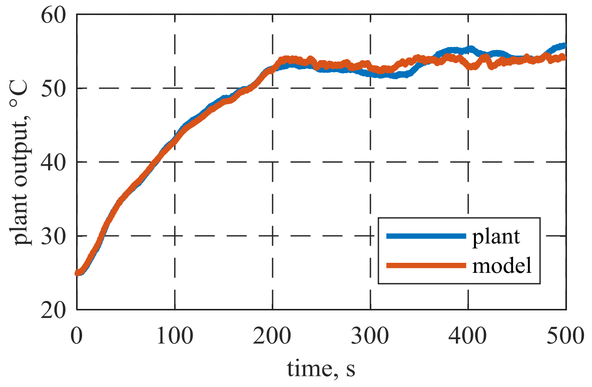

Fig. 7. Output of the plant, as well as the corresponding simulated outputs of the model (simulation mode).

In the next step, we have built several models with custom regressors. Our goal was to reduce the SVM feature vector by eliminating some lags according to a couple of different scenarios. This resulted in models from*M*1to *M*8. Model *M*1 was built with only odd regressors. The parameters of SVM were $nSVM = 1880$, $C = 5000$, $\gamma = 0.0001$.The feature vector consisted of 20 elements and can be expressed as

$$\boldsymbol{\varphi}(t) = \begin{bmatrix} u(t-1), u(t-3), \ldots & 7), u(t-19), \\ y(t-1), y(t-3), \ldots & 7), y(t-19) \end{bmatrix} \quad (4)$$

Model *M*2 was built with only even regressors. The parameters of SVM were $nSVM = 1642$, $C = 50$, $\gamma = 0.001$. The feature vector consisted of 20 elements and can be expressed as

$$\boldsymbol{\varphi}(t) = \begin{bmatrix} u(t-2), u(t-4), \ldots & 8), u(t-20), \\ y(t-2), y(t-4), \ldots & 8), y(t-20) \end{bmatrix} \quad (5)$$

Model *M*3 was built with every second even regressors created from signal *u* and every second odd regressors created from signal *y*. The parameters of SVM were $nSVM = 1610$, $C = 500$, $\gamma = 0.0001$. The feature vector consisted of 20 elements and can be expressed as

$$\boldsymbol{\varphi}(t) = \begin{bmatrix} u(t-2), u(t-4), \ldots & 8), u(t-20), \\ y(t-1), y(t-3), \ldots & 7), y(t-19) \end{bmatrix} \quad (6)$$

Model *M*4 was built with every second odd regressors created from signal *u* and every second even regressors created from signal *y*. The parameters of SVM were $nSVM = 1838$, $C = 5000$, $\gamma = 0.0001$. The feature vector consisted of 20 elements and can be expressed as

$$\boldsymbol{\varphi}(t) = \begin{bmatrix} u(t-1), u(t-3), \ldots & 7), u(t-19), \\ y(t-2), y(t-4), \ldots & 8), y(t-20) \end{bmatrix} \quad (7)$$

Model *M*5 was built with every third regressors. The parameters of SVM were $nSVM = 1734$, $C = 50$, $\gamma = 0.1$. The feature vector consisted of 14 elements and can be expressed as

$$\boldsymbol{\varphi}(t) = \begin{bmatrix} u(t-1), u(t-4), \ldots & 5), u(t-19), \\ y(t-1), y(t-4), \ldots & 6), y(t-19) \end{bmatrix} \quad (8)$$

Model *M*6 was built with every fourth regressors. The parameters of SVM were $nSVM = 1613$, $C = 50$, $\gamma = 0.001$. The feature vector consisted of 10 elements and can be expressed as

$$\boldsymbol{\varphi}(t) = \begin{bmatrix} u(t-1), u(t-5), \ldots & 3), u(t-17), \\ y(t-1), y(t-5), \ldots & 3), y(t-17) \end{bmatrix} \quad (9)$$

Model *M*7 was built with every fifth regressors. The parameters of SVM were $nSVM = 1608$, $C = 50$, $\gamma = 0.001$. The feature vector consisted of 8 elements and can be expressed as

$$\boldsymbol{\varphi}(t)=\begin{bmatrix}u(t-1),u(t-6),u(t-11),u(t-16),\\ y(t-1),y(t-6),y(t-11),y(t-16)\end{bmatrix}\qquad(10)$$

Model M_8 was built with every sixth regressors. The parameters of SVM were $nSVM=1650$, $C=50$, $\gamma=0.001$. The feature vector consisted of 8 elements and can be expressed as

$$\boldsymbol{\varphi}(t)=\begin{bmatrix}u(t-1),u(t-7),u(t-13),u(t-19),\\ y(t-1),y(t-7),y(t-13),y(t-19)\end{bmatrix}\qquad(11)$$

Table 2 presents value of the *fit* index of models M_1–M_8 for learning and validation datasets.

Table 2. Value of the *fit* index for M_1–M_8 models.

Model	*fit* [%]		
	learning (prediction)	validation (prediction)	validation (simulation)
M_1	99.32	94.68	84.24
M_2	99.12	93.75	84.45
M_3	99.22	94.77	82.43
M_4	99.17	94.44	87.04
M_5	99.57	86.44	86.55
M_6	99.12	94.29	87.13
M_7	99.14	94.41	87.59
M_8	99.13	94.33	87.49

All models built with custom regressors (M_1-M_8) had a lower value of the *fit* index for learning dataset in a prediction mode than the model M_0. The average difference was 0.52 percentage points.

In the case of the validation dataset in the prediction mode, almost all models (except for M_5) had a higher value of *fit* index than M_0. The average difference (not including M_5) was 3.48 percentage points. In the case of the validation dataset in the simulation mode, all models had a lower value of *fit* index than the M_0 model, with an average difference of 3.98 percentage points. Considering the value of *fit* index for the validation dataset (in the simulation mode) the best models were the M_7 and M_8.

6 Conclusion

The goal of this study was to analyse the impact of custom regressors on the model quality in system identification. We have proven that the vector of regressors obtained by a classical method can be shortened, while maintaining the acceptable quality of the model. In the case of the two best models, i.e. M_7 and M_8, a significant reduction in the feature vector (from 40 to 8 elements, i.e. by 80%) resulted in a negligible decrease in the quality of the model, which was equal to 2.5% only.

This work was supported by the National Science Centre [grant number UMO-2015/17/B/NZ7/02937].

References

1. A. Gretton, A. Doucet, R. Herbrich, P.W.J. Rayner, J. Schölkopf, 11th IEEE Workshop on Statistical Signal Processing, 443-449 (2001)
2. V.M. Ranković, I.Z. Nikolić, FME Transactions, **36**, 87-92 (2008)
3. B. Huyck, K. De Brabanter, F. Logist, J. De Brabanter, J. Van Impe, B. De Moor, IFAC Proc Volumes, **44**, 471-476 (2011)
4. T. Bernd, M. Kleutges, A. Kroll, Neural ComputAppl, **8**, 151-162 (1999)
5. R. Sałat, M. Awtoniuk, K. Korpysz, IFAC-PapersOnLine, **50**, 1559-1564 (2017)
6. M. Bonin V. Seghezza L. Piroddi, IET Contr Theory Appl, **4**, 1157-1168 (2010)
7. R. Sałat, M. Awtoniuk, Neural Comput Appl, **26**, 723-734 (2015)