XXXVII SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES E PROCESSAMENTO DE SINAIS - SBrT2019, 29/09/2019–02/10/2019, PETRÓPOLIS, RJ

Automatic quality control systems based on pattern recognition and sparse signal processing

Diogo Alfieri Palma[1] and Leonardo Tomazeli Duarte[2]

***Abstract*— This paper presents the application of two methods for automatic detection and classification of abnormal quality control patterns. A generator of synthetic concurrent control charts was implemented to create mixtures from patterns described in the literature. In order to obtain features for the classification step, the generated charts were initially processed via sparse regression using the Least Absolute Shrinkage and Selection Operator (LASSO) method. Then, we assess the performance of the classifier which was founded on an artificial neural network (ANN) on two different situations: i) with inputs given by the observed (raw) data and ii) with inputs given by the features generated by the LASSO method. ANN fed with sparse inputs performed extremely close to the ANN fed with raw data, using considerably less inputs.**

***Keywords*— Quality Control, Concurrent Control Charts, Sparse Regression, Artificial Neural Networks.**

I. Introduction

The search for productivity, safety and flexibility in industrial operations has played an important role with respect to competitiveness, market standards and legislation. In this context, quality management and control represents a critical success factor for organizations. A key aspect in quality control is maintenance, which is essential to reduce operational costs [1]. Indeed, maintenance plans —– corrective, preventive and predictive —– seek to mitigate and/or eliminate operational costs in three main components: nonconformity in products, inefficiency in processes and loss of opportunity in sales [2].

Once the root causes of operations poor quality are identified and action plans are elaborated, it becomes necessary to monitor and control processes, which allows the execution of these plans at appropriate times, so as to minimize unscheduled interruptions (corrective maintenance). Among the tools widely used to support maintenance are statistical process control (SPC) and, in particular, control charts, which help processes by indicating their control state, identifying causes and reducing variability, in order to achieve performance stability [3].

A control chart is essentially a time series (signal) drawn from data collected in a certain process. Data collection may occur manually or automatically by the use of sensors (e.g., temperature, humidity, pressure, dimensions of a product, etc.). Frequently, the data behind a control chart is the result of linear and/or noisy mixtures of particular patterns, which are often associated with abnormal process behavior (or abnormal causes). For instance, in Figure 1, there are some examples of abnormal patterns which are found in practice [4]. A challenging aspect in quality control, which is addressed in this paper, is how to identify and classify abnormal patterns in the case concurrent control charts, that is, when there are more than one abnormal pattern acting in the process.

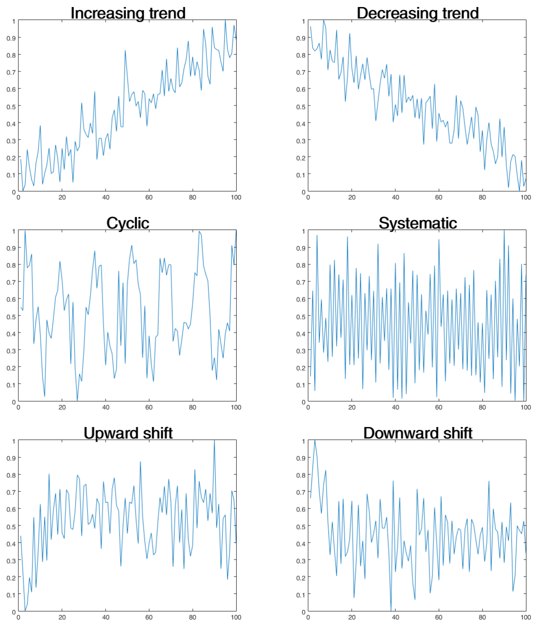

Fig. 1. Abnormal control chart patterns: (1) increasing trend, (2) decreasing trend, (3) cyclic, (4) systematic, (5) upward shift and (6) downward shift.

Previous studies have used different techniques for detecting and classifying patterns in concurrent control charts. For instance, we highlight the following ones:

- Singular spectrum analysis (SSA) and learning vector quantization network were applied by [5]. The authors also tested their methodology against real data acquired from aluminium smelting processes;
- In [6], the authors presented the application of RobustICA along with a decision tree in order to recognize patterns from the extracted features;
- Blind source separation methods were addressed by [7]. The authors extracted, selected, processed and classified data into patterns via support-vector machines (SVMs);
- A sparse regression approach was proposed by [8], using the Least Absolute Shrinkage and Selection Operator

*The authors would like to thank the research funding agencies CNPq and CAPES for the financial support.

[1]D. A. Palma is with the School of Applied Sciences, University of Campinas, Limeira, São Paulo, BR `diogoapalma at gmail.com`

[2]L. T. Duarte is with the School of Applied Sciences, University of Campinas, Limeira, São Paulo, BR `leonardo.duarte at fca.unicamp.br`

(LASSO) method and a clustering/k-means based dictionary learning process.

The present work addresses the use of two combined methods — LASSO and artificial neural networks (ANNs) — for the classification of concurrent control charts into abnormal patterns described in the literature. The paper is organized as follows. Section 2 details the methodology steps and tools used to solve the problem. Section 3 presents the experiments and the results obtained by the applied methods. Finally, Section 4 closes the paper with our conclusions.

II. METHODOLOGY

In the following, we shall present the methodological aspects of our work. In first part, we address the problem of generating samples from typical control charts. This step is important since it provides the datasets that were processed by the classifier. Then, in a second part, we introduce the main aspects of our proposal, including a brief review of the LASSO method.

A. Synthetic Data Generation

In the field of statistical quality control, there is a set of models which describe control charts with particular abnormal patterns. These patterns are often associated with shortcomings that may be found in a manufacturing process. For instance, an increasing trend pattern may indicate certain tool wearing gradually and a systematic pattern can relate to difference between shifts [4].

In the literature of quality control, there are well-established models that can be used to simulate abnormal patterns and also to generate concurrent control chart signals - Table I presents these models [7], [8], [6]. In this table, $x_i(t)$ denotes the signal that represents the i-th control chart; t is the discrete temporal index ($t = 1, \ldots, N$), where N in the number of samples. Also in Table I, there are other parameters associated with those models: σ is the control chart standard deviation, μ is the mean value of the control chart, $r_i(t)$ is a sample, at instant t, drawn from a standard normal distribution, d is the stratification deviation, a the amplitude, T the period, g the gradient and s the shift magnitude. The choice of these parameters was done according to [7]. Moreover, we consider the following values: $N = 100$, $\mu = 0$, $\sigma = 1$ and $T = 16$.

TABLE I

ABNORMAL PATTERN GENERATION.

Pattern	Equation	Parameters
Systematic	$x_i(t) = \mu + r_i(t)\sigma + d(-1)^t$	$d = 2\sigma$
Cyclic	$x_i(t) = \mu + r_i(t)\sigma + a\sin(2\pi t/T)$	$a = 2\sigma$
Inc./Decreasing Trend	$x_i(t) = \mu + r_i(t)\sigma \pm tg\sigma$	$g = 0.075\sigma$
Up/Downward Shift	$x_i(t) = \mu + r_i(t)\sigma \pm sk$	If $t > T/2$, $k = 0$. Else, $k = 1. s = 2\sigma$

Once the patterns are generated, the next step is to generate samples which correspond to mixtures of different patterns (concurrent charts). A mixture is obtained by the linear combination of two or more distinct patterns randomly chosen. In practice, it is common for the data composing a concurrent control chart to suffer from external noise, which is uncorrelated with respect to the abnormal behavior itself. In mathematical terms, this generative model is given by:

$$\mathbf{y} = \mathbf{x}_1 w_1 + \mathbf{x}_2 w_2 + \cdots + \mathbf{x}_n w_n + \mathbf{p}, \qquad (1)$$

where $\mathbf{y}$ represents the resulting linear mixture, $\mathbf{x}_i$ corresponds to an abnormal control chart pattern, w_i the weight applied to pattern i and $\mathbf{p}$ the additive white Gaussian noise (AWGN).

The number of patterns per mixture sample was set to two, because it is unlikely — given certain window size — to have more than two problem sources acting at the same time. Pattern weights were set to 0.6 and 0.4, respectively. Moreover, as in [8], we did not consider both increasing/decreasing trend and up/downward shift mixtures, because these would refer to the same behavior only with a change in direction or shift magnitude. Finally, the generated mixtures were normalized through min-max normalization, scaling data between [0, 1]. Figure 2 illustrates this process.

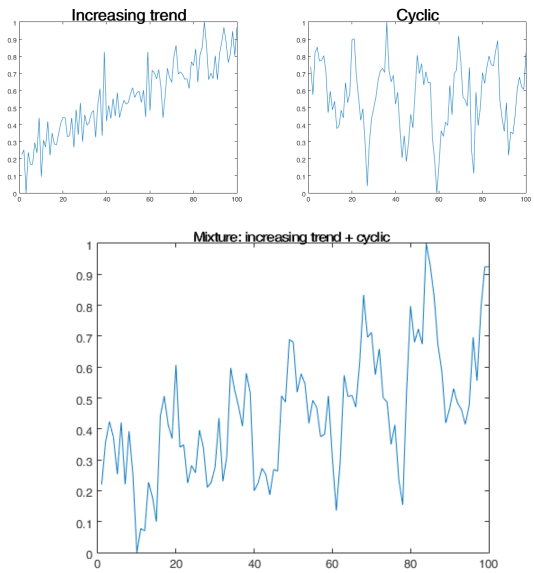

Fig. 2. Mixing process of two distinct abnormal patterns (increasing trend and cyclic) with AWGN.

B. Least Absolute Shrinkage and Selection Operator

Linear sparse regression aims at providing a linear generative model (according to (1)) with the fewest number of parameters. When the inputs $\mathbf{x}_i$ are known, sparse regression can be done by minimizing a loss function with a given constraint on the parameters w_i (i=1,...,n) related to sparsity. In our problem, the inputs, which correspond to the abnormal patterns, are not known in advance and, therefore, sparse regression can be tackled by defining a dictionary whose atoms are composed of signals that are associated with the abnormal patterns. In other words, the dictionary can be obtained by generating each one of the patterns previously described (Figure 1).

In mathematical terms, sparse regression relies on the following multiple linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon, \qquad (2)$$

which can be expressed as:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} x_1(1) & x_2(1) & \dots & x_p(1) \\ x_1(2) & x_2(2) & \dots & x_p(2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(N) & x_2(N) & \dots & x_p(N) \end{bmatrix} \times \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}$$

where vector $\mathbf{y}$ represents inputs from a control chart, $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the matrix composed by the atoms of the dictionary (each atom is arranged in column), $\boldsymbol{\beta}$ is the vector of sparse coefficients to be defined and ϵ is the noise associated with the error inherited by using a linear model.

The LASSO method was chosen since one of its main features is the assignment of value zero in several coefficients of the vector $\boldsymbol{\beta}$, which facilitates the interpretation of the solution. Such a feature is interesting in the context of variable selection/identification of relevant variables [9], [10]. Thus, the signal must be represented by the combination of atoms in the dictionary that uses a reduced number of non-null $\boldsymbol{\beta}$ coefficients. This leads to an optimization problem that can be expressed by:

$$\begin{aligned} \hat{\boldsymbol{\beta}}^{\text{lasso}} &= \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} \; \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\ &= \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} \; \underbrace{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2}_{\text{Residue}} + \lambda \underbrace{\|\boldsymbol{\beta}\|_1}_{\text{Penalty}} \end{aligned} \tag{3}$$

where λ represents a regularization parameter for the $\boldsymbol{\beta}$ penalty according to the L1 norm.

In the optimization problem expressed in (3), the greater the value assigned to λ, the greater the number of null values in $\boldsymbol{\beta}$. Therefore, the choice of λ in our context should be carried out in order to obtain a vector $\boldsymbol{\beta}$ with as many null values as possible; in this case, the coefficients that are not null would be related to the abnormal patterns that are active at a given window. However, this search for λ should also keep the ability to explain a signal given a dictionary. Figure 3 illustrates the trade-off between the value λ and the residue — mean squared error (MSE) — associated with the representation obtained by LASSO for an observed signal $\mathbf{y}$.

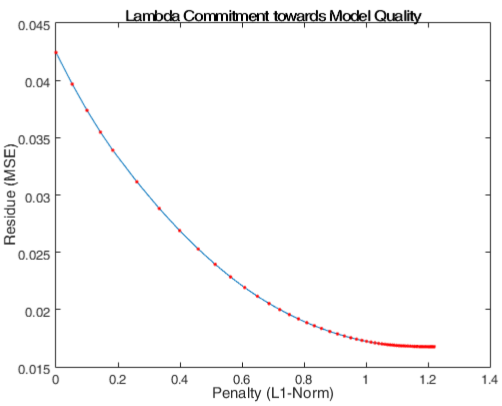

Fig. 3. Trade-off between sparsity (in the sense of the L1-Norm) and representation error for different values of λ.

As expected, MSE increases by the value in λ and the degree of sparseness/number of null values. Thus, it is evident that the choice of λ indicates a trade-off between the sparseness degree and the quality of the approximation model.

In our work, we consider an implementation of LASSO in Matlab software (function *lasso*) that addresses the optimization problem expressed in (3). This function returns the adjusted coefficients of the least squares regression for a geometric sequence of λ values [11]. Figure 4 shows the approximation of a mixture given a vector $\boldsymbol{\beta}$ (coefficients) obtained through LASSO.

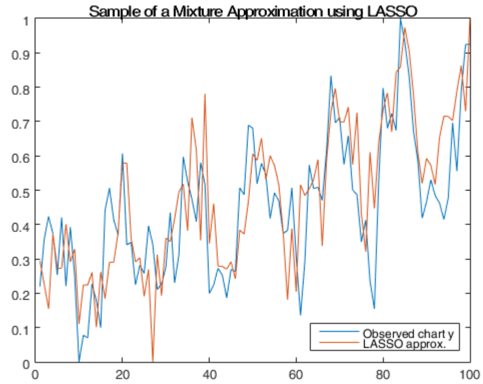

Fig. 4. Example of LASSO approximation in a mixture.

In order to select a $\boldsymbol{\beta}$ vector, a sparsity factor parameter was arbitrary set to 0.5 (50%). Basically, it means that, in addition to performing sparse regression through a dictionary containing the six abnormal patterns, the chosen vector $\boldsymbol{\beta}_{lasso}$ must have three null coefficients. Algorithm 1 was used for selecting the sparse representation given an observed signal $\mathbf{y}$, the dictionary and a sparsity factor. In case the sparsity requirement cannot be satisfied, the sparser $\boldsymbol{\beta}_{lasso}$ is returned.

Algorithm 1: SPARSE REPRESENTATION FROM LASSO

inputs: $\mathbf{y}$, dictionary, sparsityFactor
output: $\mathbf{y}_{\textbf{lasso}}$
begin
  $\beta_{matrix} \leftarrow$ *lasso*($\mathbf{y}$, dictionary)
  $\mathbf{y}_{\textbf{lasso}} \leftarrow$ last column in β_{matrix}
  minNulls $\leftarrow$ rows in $\mathbf{y}_{\textbf{lasso}}$ $\times$ sparsityFactor
  foreach β_{column} ***in*** β_{matrix} **do**
    if *nulls* ***in*** $\beta_{\text{lasso}} \geq$ minNulls **then**
      $\mathbf{y}_{\textbf{lasso}} \leftarrow \beta_{column}$
      return $\mathbf{y}_{\textbf{lasso}}$
    end
  end
  return $\mathbf{y}_{\textbf{lasso}}$
end

Once the sparse representation is defined, the resulting coefficients can be used as input variables in classifiers.

C. Artificial Neural Network

The multiclass classification with ANN was performed in Matlab with the function *patternnet*, which uses by default a multilayer perceptron (MLP) and a learning algorithm that is based on the minimization of classification error by means of a conjugate gradient type technique. This function is recommended by Matlab for training in pattern recognition problems involving a large amount of data, due to the lower memory consumption and higher execution speed when compared to algorithms traditionally set in gradient descent [12].

The ANN was configured to have only one hidden layer with the number of outputs equal to the number of patterns described: (1) increasing trend, (2) decreasing trend, (3) cyclic, (4) systematic, (5) upward shift and (6) downward shift. The default software settings related to splitting the dataset were used, so we considered 70% of data for training, 15% for validating and 15% for testing. The inputs, defined for each sample control chart, were: 100 for raw data (the total number of samples N) and 6 for sparse representation (coefficients in β). Default *patternet's* hyperbolic tangent sigmoid activation/-transfer function was used for the hidden layer. The output layer transfer function, on the other hand, has been defined as *softmax*. This function ends up normalizing the six outputs in the interval [0, 1], where the sum of these results in 1. Such a procedure renders easier the classification process in a multiclass context that must support class assignment to one or more outputs.

Table II shows an example of ANN outputs when fed with inputs (sparse representation) of a concurrent control chart sample. Once the ANN outputs are obtained, a threshold can be defined for the assignment or not of each of the classes. The indices of the classifications and their respective patterns can be obtained as shown the Table III. Finally, the obtained classifications could be compared with those patterns that originated the synthetically generated mixture sample.

TABLE II

EXAMPLE OF ANN OUTPUTS CLASS ASSIGNMENT (THRESHOLD = 0.2).

Inputs	Outputs	Classification
-0.1053	0	0
0.4388	0.5424	1
0	0	0
0.2971	0.4205	1
0	0.0113	0
0	0.0258	0
-	1	-

TABLE III

EXAMPLE OF ANN OUTPUT CLASSES RESULTING PATTERNS.

Index	Pattern
2	Decreasing Trend
4	Systematic

III. EXPERIMENTS AND RESULTS

We have performed 100 simulations and the results below refer to the mean values. For each simulation, 1000 mixture samples — splitted into training, validation and test datasets — and a dictionary containing 6 distinct abnormal patterns were generated.

Two classification strategies using ANN were compared. First, we fed an ANN with sample's raw data. Then, an ANN with only six inputs was fed using the LASSO sparse representation of these generated samples. The ANN classification threshold was set to 0.2 and the mixture AWGN signal-to-noise ratio (SNR) to 25dB. Other parameters related to synthetic data generation as well as the ANN setup are detailed in Section II.

A. Inputs: Raw Data

The ANN used for classifying raw data reached a precision of 98.59% and an error of 1.41%. The mean values obtained during tests are presented in Table IV and in Figure 5.

TABLE IV

INDICATORS OBTAINED FOR RAW DATA CLASSIFICATION ($\text{SNR} = 25dB$).

Indicator	Occurrences
FP	3 (0.31%)
FN	10 (1.10%)
TP	290 (32.23%)
TN	597 (66.36%)
Total	900 (100%)

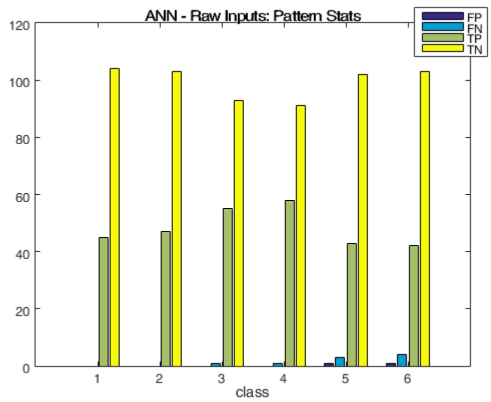

Fig. 5. Raw data classification results per class ($\text{SNR} = 25dB$).

In addition to the good results obtained by the raw data classifier, a good balance of patterns in terms of the mixtures generated for classification during tests is noted by the distribution of the classes.

B. Inputs: Sparse Coefficients

The ANN used for the classification of sparse entries generated via LASSO reached a precision of 96.88% and error of 3.12%. The mean values obtained during tests are presented in Table V and in Figure 6.

TABLE V

INDICATORS OBTAINED FOR CLASSIFICATION OF SPARSE ENTRIES ($\text{SNR} = 25dB$).

Indicator	Occurrences
FP	11 (1.21%)
FN	17 (1.91%)
TP	283 (31.42%)
TN	589 (65.45%)
Total	900 (100%)

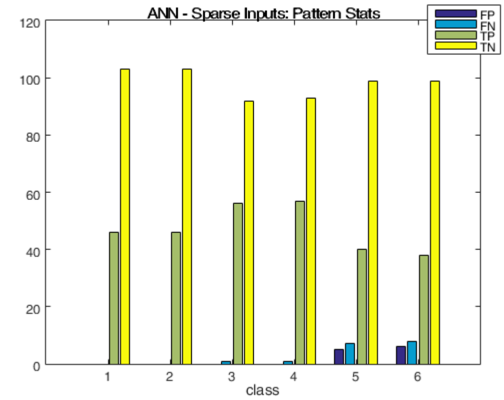

Fig. 6. Sparse entries classification results per class (SNR $= 25dB$).

When compared to the classification of raw data it is possible to observe a small increase of FP and FN in classes (5) upward shift and (6) downward shift.

C. Noise Level Variation Tests

This section presents the summarized result for trials running different noise levels. The methodology was the same used in the classification already presented for a low noise level (SNR $= 25$dB), but considered other SNR values.

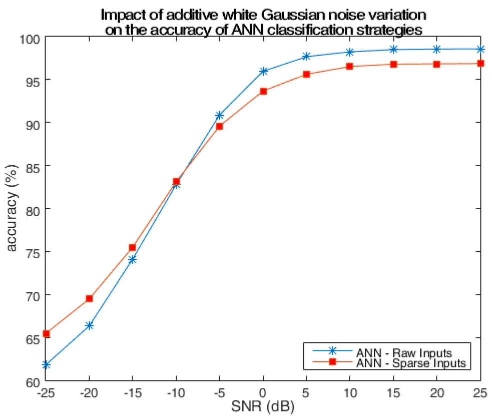

Fig. 7. Accuracy of the classification strategies for different levels of noise.

It is possible to observe (Figure 7) an advantage of the sparse representation in scenarios of stronger noise, that is, when SNR assumes the most negative values. In general, there is a small difference between the ANN fed with raw data when compared to the sparse representation in the other cases. Therefore, it can be said that the ANN fed with sparse inputs showed a performance extremely similar to the strategy using raw data.

The higher the noise level, the most a mixture starts to resemble a systematic pattern, hence why there is an accuracy decrease in this case, being the sparse representation approach able to deal with this issue in a better way when compared to the ANN fed with raw data.

IV. CONCLUSION

In this work we studied the identification and classification of concurrent control charts abnormal patterns in two different situations. In the first one, an ANN was fed with the raw data of the synthetically generated mixture samples. Then, the same ANN topology — with the exception of the input layer — was fed with the coefficients of the vector β obtained from a sparse regression step using the LASSO method.

The small difference in precision between ANN fed with 100 raw data inputs and ANN whose input layer had only six elements — of which half were null — corroborates the sparse regression ability to act as a feature selector in this classification context. In scenarios of additive Gaussian noise with positive SNR the classifiers presented a good precision. In case of more intense noise considered — with SNR -25dB — the accuracy of the classifiers proved to be superior to 50%, with a noticeable advantage to the sparse representation that resulted in a better noise attenuation.

Another aspect that can be pointed out is the lower cost of memory and disk allocation when using the sparse representation, which makes it possible to use classification techniques in scenarios in which a large volume of data is necessary or available for processing.

Overall, the results demonstrated a good performance when tested with synthetically generated data, which encourages prototyping and testing in real world situations. Future research may also consider other sparse regression methods.

REFERENCES

[1] Nigel Slack, S Chambers, C Harland, A Harrison, and R Johnston. Administração da produção. *Atlas, São Paulo*, 2009.
[2] Joseph M Juran and A Blanford Godfrey. Juran's quality handbook 5th ed, 1999.
[3] Douglas C Montgomery. *Introduction to statistical quality control*. John Wiley & Sons (New York), 6 edition, 2009.
[4] Western Electric Company. *Statistical quality control handbook*. Western Electric Company, 1956.
[5] Liangjun Xie, Nong Gu, Dalong Li, Zhiqiang Cao, Min Tan, and Saeid Nahavandi. Concurrent control chart patterns recognition with singular spectrum analysis and support vector machine. *Computers & Industrial Engineering*, 64(1):280–289, 2013.
[6] Shilpy Sharma, David A Swayne, and Charlie Obimbo. Identification of concurrent control chart patterns in time series. *International Journal of Innovative Research in Science, Engineering and Technology*, 4(6), 6 2015.
[7] Guilherme Dean Pelegrina, Leonardo Tomazeli Duarte, and Christian Jutten. Blind source separation and feature extraction in concurrent control charts pattern recognition: Novel analyses and a comparison of different methods. *Computers & Industrial Engineering*, 92:105–114, 2016.
[8] Akram Sadat Mozafari and Majid Aminnayeri. Concurrent control chart pattern recognition using a novel source separation method based on sparse representation. In *46th International Conference on Computers & Industrial Engineering 2016 (CIE46)*, 2016.
[9] Peter Filzmoser, Moritz Gschwandtner, and Valentin Todorov. Review of sparse methods in regression and classification with application to chemometrics. *Journal of Chemometrics*, 26(3-4):42–51, 2012.
[10] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.
[11] Matlab. Lasso or elastic net regularization for linear models – matlab lasso. `https://www.mathworks.com/help/stats/lasso.html#bs25w54-6`. Accessed in March 20, 2019.
[12] Matlab. Train and apply multilayer neural networks – matlab & simulink. `http://www.mathworks.com/help/nnet/ug/train-and-apply-multilayer-neural-networks.html`. Accessed in March 20, 2019.