CUSTOMIZED NEURAL NETWORK SYSTEM FOR DYNAMIC SECURITY PREVENTIVE CONTROL

J.N. Fidalgo*

Abstract

This paper proposes a new methodology for dynamic security assessment and preventive control. In the first phase, an artificial neural network (ANN) is trained to provide the security status. ANN inputs are settled by a feature selection approach that takes into account the requisites of the control algorithm, to be applied in the second phase. The adaptive control methodology is based on the steepest descent method, where the usual explicit math functions to be dealt with are emulated by the trained ANN. To illustrate the developed approach, the methodology was applied to the control of dynamic security of Madeira island power system. Results attained so far show that the proposed approach was able to find the optimal control actions.

Key Words

Neural networks, adaptive control, optimization methods, power system security

1. Introduction

In modern liberalized electricity markets, the deferral of investments in power infrastructures is compelling the systems to operate closer to their security limits. Therefore, fast and accurate assessment of the system security is required to characterize the system behaviour following a given potential contingency. In case of dangerous insecurity, preventive control actions should be taken to avoid system collapse.

The dynamic behaviour of real systems is usually modelled by a set of non-linear differential equations. Sometimes, when the mathematical model of system components is not completely known, system representation may be acquired using system identification techniques or even by non-conventional approaches where the system mathematical description is replaced by, for instance, a set of rules. Machine learning techniques are frequently used in this task, exploring the input/output relations implicitly contained in the set of patterns. Yet, modelling the dynamic behaviour is just part of the problem. For instance, if one intends to use a gradient-based approach, it should also be possible to compute the derivatives of system outputs regarding its inputs. This ultimately means that the representation tool should be able to provide this kind of information.

In the last two decades, the application of optimization or control methodologies based on artificial neural network (ANN) has been reported in the literature [1–10]. In particular, [4] (a survey on neuro-control technologies) and [5] (a Ph.D. thesis) contain several references to this theme. The design of control algorithms to the transient stability problem is also considered in a number of publications [6–8]. However, application examples mentioned in a considerable part of these references are not related to real world problems and/or the control methodology does not consider explicitly the dependencies between the control variables. This may lead to unfeasible states, i.e., to problems in which several operation or security constraints are not satisfied. In [9], the search algorithm consists of decreasing the power in the critical machines and increasing it in the noncritical ones. Yet, the generation changes are not optimized as this approach does not consider the different sensitivities of the stability index with respect to individual power production. In this paper, we aim at giving a contribution to overcome these shortcomings of presently available approaches.

The research described in this paper was developed within the framework of an European Union R&D project that aimed at developing an advanced control system for isolated networks with high penetration of renewables [10]. In this project, ANN were used for system representation because they usually perform faster and better than concurrent tools, namely in the problem of dynamic security assessment. Besides, ANN provide the estimation of the system outputs, and not only a binary response like some other concurrent tools. Moreover, they offer effective mechanisms of computing the derivatives of system output regarding the input variables. This capacity allows, in one hand, the application of gradient-based methods for control purposes and, on the other hand, the implementa-
is the weight of the connection between the units. The contribution of the element of the input vector that stimulates the network. one or more hidden layers and one output layer. The input vector of the next layer is given by the product of this vector in the unit and the weight of the synapse connecting unit $i$ to $k$.

A typical ANN architecture of feedforward type, where $x$ and $O$ are respectively the input and output vectors, and $w_{ki}$ is the weight of the synapse connecting unit $i$ to $k$.

A typical ANN structure can present one input layer, one or more hidden layers and one output layer. The input layer receives the input vector that stimulates the network. The contribution of the element $i$ of this vector in the unit $j$ of the next layer is given by the product $x_i w_{ji}$, where $w_{ji}$ is the weight of the connection between the units $i$ and $j$. The net input of unit $j$ is then given by (1):

$$net_j = \sum_i w_{ji}x_i$$

The value $net_j$ is then passed through a non-linear function (activation function) that establishes the output state of unit $j$. The activation function considered in this study was the hyperbolic tangent $tgh$, resulting for unit $j$ output:

$$O_j = f(net_j + \theta_j) = tgh(net_j + \theta_j)$$

In this expression, $\theta_j$ is the bias value of unit $j$, whose role is to shift the transition region of function $f$. Transfer functions are generally of sigmoid type, like hyperbolic tangent, for hidden layer(s) and linear for output layer. Input units just distribute input values $x_i$. The outputs of all hidden units are added in the output layer units to compute the output values.

In harder regression tasks, the inclusion of additional hidden units allow the generation of more complex decision regions, able to interpolate among the training examples. On the other hand, if the classification boundaries are simple, the ANN will not need many hidden units.

2.2 ANN Training

A most interesting characteristic of ANN is its ability to learn from training data rather than from a set of rules. Suppose that Fig. 1 represents an ANN trained to approximate a given function $y = f(x)$, where $y$ and $x$ represent, respectively, output and input vectors. In this case, ANN output vector $O$ should be close to $y$, for all the patterns considered in the training set. The parameter usually chosen to represent the network performance is given by (3):

$$E = \frac{1}{2} \sum_{k=1}^{N_p} (y_k - O_k)$$

In this expression, $y_k$ is the desired output. $O_k$ represents the actual output, calculated with the current values of weights and input vector and $N_p$ is the number of training patterns. $E$ should be reduced to a minimum during the training phase.

The backpropagation (BP) algorithm is a learning rule based on the steepest descent method. In the BP algorithm, the weights are updated iteratively according to a gradient optimization procedure using the entire training set according to (4) where $\eta$ is the learning rate parameter:

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}}$$

A way to increase the learning speed of this training phase, without causing oscillations, consists of modifying expression (4) to include a momentum term:

$$w_{kj}(n+1) = -\eta \frac{\partial E}{\partial w_{kj}} + \alpha \Delta w_{kj}(n)$$

A more detailed description of the training algorithm can be found in [12–14].

In this work, we used the adaptive backpropagation (ABP) training algorithm [15]. The ABP is based on BP but it uses an individual adaptive learning rate for each
weight, which provides a much faster learning process. The stop training criterion was based on the well-known cross validation principle, which fights against overfitting [13, 14]. The patterns were normalized to have zero mean and a standard deviation of one. This removes of offset issues and measurement scales.

3. ANN Sensitivity Analysis

Considering that the training phase is completed, how can we assess the sensitivities of ANN outputs regarding its inputs? Computing these derivatives using a trained ANN is in fact quite simple and can be based on the methodology described in the next paragraphs that is quite well known in the scientific literature [1, 2, 9].

However, to facilitate the understanding of the control approach described in Section 7, a short explanation is provided. Figure 2 presents a “one unit per layer” scheme,” where the derivative of the output \( O_j \) with respect to \( O_k \), the output of unit \( k \), is given by expression (6) where \( f'_k \) is the derivative of \( f_k \) at point \( n_k \). Using a chain rule we can then obtain (7). In these expressions the index \( j \) represents all units fed by input \( x_i \):

\[
\frac{\partial O_k}{\partial O_j} = f'_k w_{kj} \tag{6}
\]

\[
\frac{\partial O_k}{\partial x_i} = \sum_j f'_j w_{ji} \frac{\partial O_j}{\partial x_j} \tag{7}
\]

If the ANN is a fully connected one, index \( k \) refers to all units in the hidden layer. This simple and elegant formula is similar to the weights adaptations used in the BP algorithm [13, 14].

Gradient components for each input (each individual axe direction in the gradient hyperspace) are settled by (7). Note that this calculation is performed automatically for any input pattern, i.e., for any system state. This way, optimization gradient-based methods are easily applied. Naturally, some application constraints may have to be included depending on the particular problem to be dealt with, as it will be addressed later in this paper.

4. Dynamic Security Characterization

The dynamic security behaviour is formally described by a set of differential equations that represent the electrical and mechanical components of conventional generation machines [16, 17]. Wind asynchronous generators are modelled following the third-order electrical–mechanical transient model [16, 17]. The assessment of the security state involves the numerical integration of these equations, which requires large computational times. In the present approach, ANN is used for fast and accurate estimation of the security state. ANN training is preceded by a data set (DS) generation phase.

5. Data Set Generation

The preventive control algorithm will be applied whenever insecurity is detected, which demands high-quality assessment performance of the security index. The DS should be shaped diligently to represent the diversity of feasible operation states. The assortment of operation points (OPs) was granted by a stratified sampling procedure (structured Monte Carlo) [8, 10]. Several operating settings were considered:

- total active load (\( P_{load} \));
- active power in each non-dispatchable generation station (\( P_{nd} \)) – wind and private generators, or machines with hard dispatch constraints;
- scheduling and dispatch solution for the other generators.

The system domain is formerly delimited and subdivided into smaller partitions, according to the range and resolution assigned for the independent operating conditions to change, namely \( P_{load} \) and the \( P_{nd} \) productions. For each partition, \( P_{load} \) and all \( P_{nd} \) variables are randomly sampled. Extra OP near the security boundary were generated to provide an improved accuracy for these operating conditions. The DS also includes operation scenarios that do not correspond to the optimum economical dispatch, because ANN should provide accurate security indexes even when the control algorithm changes the dispatch solution.

After the definition of each DS operating scenario, a power flow is run to check the feasibility of the network steady-state operating conditions, defining the system pre-disturbance operation point. The security index following a given disturbance is then computed by the numerical integration of the system state equations. In this work the security index adopted is the maximum value reached by negative frequency deviations \( \Delta f \). If the frequency deviation caused by a given disturbance is greater than a certain threshold, protective devices (such as frequency relays) become active, performing the protection actions defined by local utility: usually, disconnection of a power plant or load shedding.

6. ANN Security Modelling

In this research, the system security behaviour is emulated by an ANN. The design of the security assessment structure involves two main interactive stages: feature selection and ANN training. The goal consists of a set of attributes (ANN inputs) having the following characteristics:

- to present a high-quality assessment performance of the security index;
• For a matter of simplicity, the number of attributes should be as low as possible without losing relevant information. This explains the use of the concept of “equivalent machine” – a group of similar generators swinging together in the same station. System control variables, namely dispatch variables, should be selected as potential feature candidates;
• Simulations performed in some systems showed that security assessment performance was improved when some dependent variables (like $P$ – generated power and $SR$ – spinning reserve) were chosen as system attributes. However, to simplify the control algorithm, controllable variables should be independent or, at least, its relation should be “clear-cut.” This requirement will be explained later in this paper.

In the example of Section 8, the final vector of inputs comprises 33 variables: $P_{load}$ – total active load, $P_{wj}$, $N_{wj}$ – active power and number of operating units in each equivalent wind generator, $P_{ci}$, $SR_{i}$ – active generated power and spinning reserve in the remaining equivalent generators.

7. Proposed Control Algorithm

7.1 Overview

The security module performs a continuous check of the security state for each selected disturbance – an ANN was trained to output the expected security index for each considered disturbance. The main goal of the preventive control algorithm (PrevC) consists of presenting an alternative secure dispatch solution after a potential insecure control algorithm (PrevC) consists of presenting an alternative secure dispatch solution after a potential insecure control. Let us consider the virtual variable $P_{ci}$ instead of $P_{ci}$ and $SR_{i}$. One may even imagine a very simple ANN (dashed line) to perform this task. Using the unitary weights shown near each synapse, it is easy to confirm that:

$$P_{ci} = P'_{ci}$$

$$SR_{i} = P_{ci}^\text{max} - P'_{ci}$$

So, using the chain rule, that results on (16):

$$\frac{\partial \Delta f}{\partial P'_{ci}} = \frac{\partial \Delta f}{\partial P_{ci}} \frac{\partial P_{ci}}{\partial P'_{ci}} + \frac{\partial \Delta f}{\partial SR_{i}} \frac{\partial SR_{i}}{\partial P'_{ci}}$$

7.3 Computing Gradient Components

Gradient vector components (derivatives of ANN outputs with respect to controllable inputs) are given by (7), in Section 3. This would be enough to apply the gradient technique if there were no constraints. In fact, optimizing the function (8) would be much simpler if constraints (9–13) had not to be considered. Note, for instance, that if an ANN input $P_{ci}$ was changed without considering the other inputs, one would simply increase $P_{ci}$ if $\Delta f/\partial P_{ci} > 0$, and decrease $P_{ci}$ if $\partial f/\partial P_{ci} < 0$. However, when $P_{ci}$ is increased, the spinning reserve $SR_{i}$ should be decreased in the same amount. So, one should only increase $P_{ci}$ if $\partial f/\partial P_{ci} - \partial f/\partial SR_{i} > 0$.

Figure 3 shows another interpretation of this conclusion. Let us consider the virtual variable $P'_{ci}$ instead of $P_{ci}$ and $SR_{i}$. One may even imagine a very simple ANN (dashed line) to perform this task. Using the unitary weights shown near each synapse, it is easy to confirm that: The next equations describe the mathematical formulation of the problem:

$$\text{Max } \Delta f(P_{ci}, SR_{i}, N_{wj}, P_{wj})$$

Subject to:

$$P_{ci}^\text{min} \leq P_{ci} \leq P_{ci}^\text{max}$$

$$SR_{i} = P_{ci}^\text{max} - P_{ci}$$

$$N_{wj} \in \{0, 1, \ldots, N_{j}\}$$

$$P_{wj} = N_{wj}P_{wj}^\text{nd}$$

In this formulation:

$P_{load}$ – total active load;
$P_{ci}$, $SR_{i}$, $P_{ci}^\text{min}$, $P_{ci}^\text{max}$ – active generated power, spinning reserve, minimum and maximum technical limits of the equivalent “non-wind” generator $i$;
$N_{wj}$, $N_{j}$ – number of operating units and number of available units in the equivalent wind generator $j$;
$P_{wj}$, $P_{wj}^\text{nd}$ – power generated by the equivalent wind generator $j$ and power generated by each individual unit of this cluster.
And using (14) and (15) we obtain:

$$\frac{\partial \Delta f}{\partial P_{c_i}} = \frac{\partial \Delta f}{\partial P_{c_i}} - \frac{\partial \Delta f}{\partial SR_i}$$  \hspace{1cm} (17)

For wind type generators, a similar approach can be developed, exploiting again the ANN inputs relation described by Fig. 3 and (13). This leads to (18):

$$\frac{\partial \Delta f}{\partial P_{w_j}^*} = \frac{\partial \Delta f}{\partial P_{w_j}} + \frac{\partial \Delta f}{\partial N_{w_j}} \frac{1}{P_{w_j}^{ind}}$$  \hspace{1cm} (18)

Constraint (16) means that if $P_{c_i}$ is increased, the global generation from all the other generators should be decreased to keep the balance between generation and load. Constraint (17) should also be considered, so that scenarios outputted by the control algorithm would always correspond to feasible power flow solutions. The other constraints (12 and 13) are related with wind power generation. Constraint (12) states that the number $N_{w_j}$ of operating units in the equivalent wind generator $j$ may assume integer values from zero to $N_j$ – the number of available units in this grouping. Constraint (13) states that all generators in group $j$ are generating the same amount of power $P_{w_j}^{ind}$. Changing the wind power generation for a given group of generators is then made on discrete steps, by turning on/off individual wind generators. To deal with all these constraints, and following at the same time the ultimate goal of increasing $\Delta f$, the subsequent approach was adopted:

1. Compute ANN sensitivities for each individual ANN input;
2. For each controllable equivalent machine, evaluate the “composed” sensitivity using (18) for wind type machines and (17) for the remaining ones;
3. To satisfy constraint (9), the sum of all generation changes must be null, i.e.:

$$\sum_i \Delta P_{c_i} + \sum_j \Delta P_{w_j} = 0$$  \hspace{1cm} (19)

Constrain (19) is automatically enforced if the composed sensitivity indexes of controllable variables are normalized to have zero mean. This way, the power increment on group $i$ will be given by:

$$P_i = hs_i$$  \hspace{1cm} (20)

where $h$ represents the gradient step and $s_i$ is the normalized value of the composed sensitivity. The new operation point will be given by $P_i + \Delta P_i$. Thus, the total power production change will be $\sum h s_i = h \sum s_i = 0$, because $s_i$ is settled to have zero mean. However, during the iterative process, constraint (9) is only partially satisfied by the sensitivities normalization strategy because:

1. Wind power may only assume discrete values (13).
    Generally, the new $P_{w_j}$ will not be matched by one of the combinations $N_{w_j} P_{w_j}^{ind}$. In this case, PrevC will search these combinations and chooses the closest value to $P_{w_j}$. The obtained difference (usually a small value) is then distributed by non-wind controllable generators, following their $s_i$ values;
2. Occasionally, the gradient step may suggest a new operating point that tends to violate the technical generation limit of a given generator ($P_i > P_i^{max}$). In this case, PrevC considers $P_i = P_i^{max}$ and the remaining power is distributed again like in point 1.

8. Results

8.1 Power System Description

It is not in the scope of the present paper to provide a full description of power systems under analysis (Madeira Island), where the approach previously described was applied. A detailed picture of this system can be found in [6]. However, for a matter of illustration, a few general details are provided. The power system of Madeira comprises 2 thermal power stations, with a total of 37 diesel generators, 1 waste to energy power plant, 6 hydropower plants and 6 wind parks. The total installed power is slightly above 200 MW.

Three types of disturbances were considered in the tests [6], although this article refers only to one of them: short circuit that causes the disconnection of two wind parks and two power plants. This disturbance usually causes large frequency drops, leading to the operation of protection devices and/or to system insecurity. The system was considered insecure if the negative frequency deviations ($\Delta f$) go below $-2$ Hz, as protection devices are set to operate when negative frequency drops $-2$ Hz regarding the nominal value of 50 Hz.
8.2 Security Assessment Performance

An ANN was trained for the considered disturbance. A total of 8,028 patterns were generated: 70% of them were randomly selected for ANN training and validation, and the remaining 30% for performance evaluation purposes (testing set). Table 1 presents the obtained performance indexes: mean absolute deviation (MAD) and root mean square deviation (RMSD).

8.3 Preventive Control

Figure 4 shows two successful examples of the preventive control algorithm. Each case corresponds to a particular initial operating point. Lines represent the power generated by available controllable machines and are referred to the left abscissa axe; the other generators (independent or switched off) are not shown. The triangles represent the security index $\Delta f$ value (referred to the right abscissa axe) that increases during the iterative process until it goes beyond the security threshold of $-2$ Hz. In each algorithm cycle, the gradient direction commands the changing of the values of the control variables. These variables’ evolution is not monotonous during the iterative process. For instance, in the first example, Pc11 increases in iteration 3 but it decreases in iteration 4. This confirms that the gradient should be computed in every cycle.

There were also some cases for which PrevC was not able to find a secure solution. In fact, PrevC is based on power exchanges among available generators and sometimes the security domain cannot be reached without including a new dimension in the search space, namely by switching on other generators. In the next development phase of this work, new unit commitment alternatives will also be included in the search space to enhance the current version of this approach.

9. Conclusions

This paper describes a robust and innovative adaptive control procedure. The search algorithm is based on a steepest descent iterative process where “small” quantities are exchanged among active controllable variables. The proposed control methodology is based on a trained ANN and it is able to deal with non-continuous (e.g., wind park production) and/or non-controllable (e.g., active power from independent producers).

The illustration example shows that system functional math may be accurately represented by a trained ANN, despite the need to enforce several constraints in the choice of the ANN inputs. Different types of constraints were handled during the control process, providing new feasible (and secure) operation states.

References

Biography

J. N. Fidalgo was born in Porto, Portugal, on August 30, 1961. He is an assistant professor in the Department of Electrical Engineering and Computers of the Engineering Faculty of Porto University and a Senior Researcher at Power Systems Unit of Institute of Engineering Systems and Computers (INESC Porto). He obtained a Ph.D. degree in Electrical Engineering and Computers from the same University in 1995. His interests include pattern recognition, computational intelligence and neural networks and its applications to power systems.