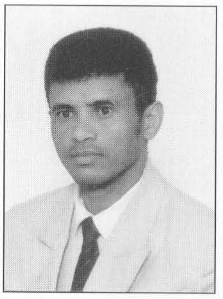
Getachew Alemu WONDIM
WARSAW UNIVERSITY OF TECHNOLOGY, INSTITUTE OF CONTROL AND INDUSTRIAL ELECTRONICS

A fault detection method in dynamic systems, based on an Euclidean measure, between the weight vectors of the model neural networks

Getachew Alemu WONDIM (M.Sc.)

He is a graduate of the Warsaw University of Technology, in the field of Computer Engineering. Currently, he is on a Ph.D. research, with the institute of Control and Industrial Electronics (WUT).

getachew@isep.pw.edu.pl



Abstract

In this paper, we present a certain method of fault detection in dynamic systems by assuming some boundary conditions. The proposed method relies primarily, on preparing a neural database of the model neural networks, which are supposed to represent the dynamic system on its different operating points. There after, we assume a certain fault state of the system composed of two or more of the faults in the database and we point out to which of the faults the assumed system belongs. The core of the method is computing the Euclidean distance between the output layer weight vectors of the model neural networks in the database, and the neural network model representing the new assumed state of the system. Based on the computed Euclidean measure, we conclude that, the fault model, which has the minimum Euclidean distance to the new assumed system state model, is the most probable to happen. The neural network models used are of the RMLP (Recurrent Multilayer Perceptron) types, each of which are assumed to possess only one output layer neuron.

Streszczenie

W artykule, przedstawiamy pewne metody wykrywanie uszkodzeń w układach dynamicznych przy pewnych założeniach. Przedstawiona metoda opiera się na budowaniu neuronowe bank modeli układu reprezentujące układ w róż nich punktach pracy. Po przegotowaniu takim, założymy nowe wadliwe stan układu składające z różnych stanów uszkodzeń z bazy, i na podstawie zaproponowaną metodę wnioskujemy do których stanów można zakwalifikować tego nowe założonego układu. Ważnym elementem metody jest obliczenia odległość Euklidesowa między wektorów wag wyjściowych modeli neuronowych w bazie danych i modeli sieci reprezentujące nowe stan układu. Na podstawie tę odległości wnioskujemy, że, model uszkodzenia, który ma minimalne odległość Euklidesowej na nowego modeli systemu, jest ten który nastąpiło. Wykorzystywane sieci są typu RMLP (Recurrent Multilayer Perceptron) i przyjęliśmy założenia że każde model sieci neuronowej zawiera tylko jeden neuron wyjściowe.

Keywords: Dynamic systems, neural networks, Euclidean measure, fault detection
Słowa kluczowe: Układ dynamiczne, sieci neuronowe, miara Euklidesowa, wykrywanie uszkodzeń

1. Introduction

The possibility of the approximation of any non-linear function to any reasonable accuracy degree, by neural networks, and their generalization capacity, suits their application for the modeling and identification of non-linear dynamic systems based only on the inputs and the outputs, without the knowledge of the exact mathematical model of the system.

In such a case, the general form of a nonlinear dynamic system which may be defined as

$$y(k)=f(y(k-1), y(k-2),\dots,y(k-n), u(k), u(k-1),\dots\dots,u(k-m)) \qquad (1)$$

where $u(j)$ and $y(j)$ are signals representing the input and the output of the system at the j'th instant, respectively, can be represented by a neural network which is well trained to identify (to approximate) the function f of the system.

Hence, by assuming that the model neural network is well trained and well represent the plant ($y \approx y_{NN}$), the neural network model of the system can be represented as:

$$y_{NN}(k)=\hat{f}(y_{NN}(k), y_{NN}(k-1),\dots , y_{NN}(k-n),u(k),u(k-1),\dots ,u(k-m)) \qquad (2)$$

where y_{NN} is the output of the trained model neural network, $\hat{f}$ is a nonlinear input output relation of the neural network which approximates the function f of the plant, n is the number of the delayed inputs and m is the number of the delayed outputs of the system.

Assuming a dynamic system to have "n" different operation points one of which is the normal state of the system and the rest "n-1" are the different fault states of the same system in a certain pre-defined interval of the domain, named, the fault interval; in [6], a fault detection method which enables to classifies a new assumed state of the system for its belonging to one of the predefined database systems, is presented. The mentioned work assumes only one of the faults in the database to occur, in the whole of the assumed fault interval of the new state of the system.

2. Assumptions and problem formulation

In this work, we assume the new state of the system, which has to be classified, to be composed of more than one of the faults from the database in the assumed fault interval. In this assumption, the main fault interval of the new system-state is divided in to different fault-sub-intervals, so that, in each of the fault-sub-intervals occurs any one of the faults from the database.

The dynamic system is subjected to be in its assumed fault states, in our case, by adding randomly generated values from different random generation intervals, to the healthy (no fault) output of the system. Neural networks of the RMLP type are trained to capture the behavior of this system on its different operational points based on the inputs and outputs at those operational points.

Our aim consists of presenting a method of fault detection, in the case, where the new signal, which is to be classified, is composed of portions of different faults, which are identical or similar to those in the predefined model database states of the system.

We assume that the number of the fault-sub-intervals in the main fault interval of the new state, to be less than or equal to the number of neural models in our pre-defined database.

The neural network models used are of the RMLP (Recurrent Multilayer Perceptron) types, each of which are assumed to possess only one output layer neuron. Through out our discussion, in each of the model neural networks, we consider only the output layer weights, which we denote by vector W.

3. Problem solution

Assuming that an arbitrary dynamic system operates not properly in a certain interval of its domain, we denote this time interval as $[t_1 , t_2]$ and this is what we call the fault interval. Let in this fault interval, $[t_1 , t_2]$, occur "k" different faults. This means that, the fault interval $[t_1 , t_2]$ is divided in to k-portions, in each of which, one of the "k" faults occur. It is accepted that in a single fault portion, only one of the k-faults acts. We may denote these k sub-intervals of the main fault interval, as $[p_0 , p_1]$, $[p_1 , p_2]$, ..., $[p_{k-1} , p_k]$ where p_0 and p_k coincides respectively to t_1 and t_2.

We build "k" neural networks representing the behavior of each of these fault-states in these "k" fault-sub-intervals.

For the fault-sub-interval $[p_0 , p_1]$, the initial values of the output layer weights (W0n), n=1, 2, …, k of each of the neural networks at the beginning of the fault interval $t_1=p_0$, is identical for all of the k-networks $W01 = W02 =,.....,= W0k$
Each of the "k" networks having the above initial weight values are subjected to training on the basis of its respective model input output, beginning from the point p_0 and ending at the point p_1. At the point p_1, we receive the final weight values for each of the networks denoted as $W11,\ W12,\ .....,\ W1k$

Having received the weight values at the point p1, we calculate the Euclidean distance between the pair $(W0n\ ,\ W1n)$ defined as

$$Norm\_p1(n) = \sqrt{\sum_{i=1}^{Nmid} \left(W0n(i) - W1n(i)\right)^2} \tag{3}$$

for (n=1, 2, …, k), $Nmid$ is the number of the middle layer neurons together with the bias.

For the rest of the fault-sub-intervals $[p_{i-1} , p_i]$**, where i=2, 3, …, k-1:** except the first fault-sub-interval $[p_0 , p_1]$, where the initial weight vectors of all of the neural networks took identical values, for the rest of the sub-intervals (or for the points starting from the point p_1), the initial weight vector value at the beginning of that sub-interval, depends, on which of the faults has occurred in the interval-portion immediately before it. For example if in $[p_0 , p_1]$ occurred fault 3, then, the identification of the fault which may occur in the sub-interval $[p_1 , p_2]$ must have the initial weight vector value at p_1, which is the final weight vector value obtained when identifying fault 3 during its action in $[p_0 , p_1]$.

Hence, for the fault which is supposed to happen in the interval portion $[p_{i-1} , p_i]$, i=2,3, …, k, having an aim to calculate the Euclidean distance between the pair of the vectors, i.e., the initial weight vector of this fault at the point p_{i-1} and the final weight vector at the point p_i, it needs the evaluation of many cases. There are "k" possible initial weight vectors at the point p_{i-1} depending on which of the "k" faults has happened in the immediate previous sub-interval, ending its action at the point p_{i-1}. We will be forced to model the fault in this interval by building as much neural networks as the number of the probable faults, which could occur in the previous sub-interval, this number from our assumption is "k".

So, if a fault "n" n=1, 2, 3,…, k is assumed to occur in the interval $[p_{i-1} , p_i]$, i=2,3, …,l k, then, we calculate the following values:

$$Norm\_p_i(n^m) = \sqrt{\sum_{q=1}^{Nmid} \left(W_{(i-1)m}(q) - W_{(i)n}(q)\right)^2} \tag{4}$$

where, i=2, 3, …, k, n=1, 2, …, k, m=1,2, …, k.

In the above expression, $Norm\_p_i(n^m)$ means the Euclidean distance between the weight vectors at the points p_{i-1} and p_i, in the cases, when in the fault sub-interval $[p_{i-2} , p_{i-1}]$ occurs fault m, and in the fault sub-interval $[p_{i-1} , p_i]$ occurs fault n.

Having prepared such a neural model bank, we assume an arbitrary new-fault-state of our system in the predefined fault interval $[t_1\ \ t_2]$. Additionally, we assume that, in the mentioned interval, all or part of the faults in our database neural bank may occur in the different fault-sub-intervals of $[t_1\ \ t_2]$ in any arbitrary order of occurrence.

This new-system state is identified by a RMLP type neural network, which of structure is mentioned above. The initial weight vector for this neural network is taken to be identical to the initial weights of those of the neural networks in the model bank, at p_0.

During training this neural network, beginning at the point t_1 (the beginning of the fault interval), for every of the fault-sub-intervals, $[p_0\ \ p_1],[p_1\ \ p_2],[p_2\ \ p_3],.......,[p_{k-1}\ \ p_k]$, we register the output layer weight vector values of this network, at each of the points p_1, p_2, …, p_k and we calculate the Euclidean distance of the pair of the weight vectors at the two consecutive points as.

$$Newnorm\_p_i = \sqrt{\sum_{q=1}^{Nmid} \left(W_{(i-1)}(q) - W_{(i)}(q)\right)^2} \tag{5}$$

where $Newnorm\_p_i$ is the Euclidean measure of the weight vectors, $W_{(i-1)}$ at the points p(i-1) and $W_{(i)}$ at the point p_i, in the new system-state identification. $Nmid$ is the number of the middle layer neurons together with the bias.

Having calculated the Euclidean distance at the points p_1, p_2, …, p_k in the new-system-state identification, the proposed fault detection method consists of comparing the calculated Euclidean measures in the new-system-state with those Euclidean distances in the prepared database model bank at the corresponding fault-sub-intervals.

The conclusion for the occurrence of a defined fault, in a given fault-sub-interval, can be done according to the following.

For each of the fault-sub-intervals $[p_0\ \ p_1],[p_1\ \ p_2],[p_2\ \ p_3],.......,[p_{k-1}\ \ p_k]$, we define the following minimum,

$$\min_n (\ Newnorm\_p_i\ -\ Norm\_p_i(n^m)\ ) \tag{6}$$

where $n = 1,2,....,k$; $i = 1,2,....,k$; $m = 1,2,.....,k$ and we conclude that, a fault "n" occurred in the given fault-sub-interval, $[p_{i-1}\ \ p_i]$, if it satisfies this minimum.

4. Example

Consider a dynamic system expressed as:

$$y(k) = \frac{y(k-1)\,y(k-2)\,(y(k-1)+0.25)}{1+(y(k-1))^2+(y(k-2))^2} + (u(k))^3 + 0.5 \tag{7}$$

where $u(k) = (-1)^k \left|\sin\left(\frac{2\pi k}{90}\right)\right|$ is the input signal to the system.

Assume the fault interval of this system is [80, 160]. For this system, a neural bank representing the system on different operation points is prepared. Let the possible faults, which may occur to this system in the mentioned fault interval, be identical or similar to one or more of the following states which are supposed to be in the model database.

$$fault1\ (k) = y(k) + (rand(1) * 10 - 5) \tag{8}$$

$$fault2\ (k) = y(k) + (rand(1) * 7 - 3.5) \tag{9}$$

$$fault3\ (k) = y(k) + (rand(1) * 20 - 10) \tag{10}$$

where k takes values in [80, 160], $y(k)$ is as in (7), rand (1), is an arbitrary random value in [0 1].

For the simplicity, we assume that the main fault interval is divided in to only two fault-sub-intervals and let these sub-intervals be, $[p_0 , p_1]$=[80, 120] and $[p_1 , p_2]$=[121, 160]. In each of these two fault-sub-intervals, any one of the three faults defined by (8)-(10) can occur. For the portion $[p_0 , p_1]$ we train three neural networks representing the three possible fault-states. We know that at point p_0 all of the networks have identical initial weight values and at point p_1 we receive different final weight values for these three networks because of the difference in the input output.

For the sub-interval $[p_1\ \ p_2]$, to model the networks representing each of the three possible faults, the initial vector value at the point p_1 depends on the fault, which may have occurred in the previous fault sub-interval $[p_0\ \ p_1]$.

Table 1 shows the common for all initial weight vector values taken at the point p_0, the final weight vectors values of each of the networks at the point p_1, and the Euclidean measure between the initial weight vector and the final weight vector for the respective cases.

Table 1. Initial weight vector values at p_0, final weight vectors values at p_1, and the Euclidean measure between the initial weight vector and the final weight vector for the respective cases.
Tabela 1. Początkowe wartość wektorów wag w punkcie p_0, końcowe wartość wektorów wag w punkcie p_1, i odległość Euklidesowa między wektorami wejściowymi i wyjściowymi dla odpowiednich przypadków

The initial weight for all cases at p0 (W0)	The final weights at p1			The Euclidean distance between		
	Fault1 (W11)	fault2 (W21)	Fault3 (W31)	W11 and W0	W21 and W0	W31 and W0
0.00445	-0.00065	0.00008	-0.00012			
0.00894	-0.00070	-0.00041	-0.00047			
0.00199	-0.00020	-0.00007	-0.00011	0.14008	0.11352	0.29121
0.00672	-0.00069	-0.00029	-0.00028			
0.00190	-0.00085	-0.00015	-0.00024			
0.00594	-0.00025	-0.00007	-0.00019			
0.00818	-0.13110	0.12081	-0.28268			

On the process of the identification of the fault which is supposed to occur in the sub-interval $[p_1 \ \ p_2]$=[120, 160], depending on which of the faults took place in the previous sub-interval $[p_0 \ \ p_1]$=[80, 120], the initial weight values at point p_1 can be any of the three vectors $W11$ or $W21$ or $W31$, of Table 1.

This indicates that, the final weight vector value expected at the point p_2 is dependent on the initial weight value at point p_1, and on which of the fault occurred in the sub-interval $[p_1 \ \ p_2]$.

Table 2a-c shows all the possible cases.

Table 2a-c. The final weights at p_2, considering different fault cases in p_0-p_1
Tabela 2a-c. Końcowe wartość wag w punkcie p_2, dla różnych przypadków uszkodzeń w podprzedziale p_0-p_1

The final weights at p2, in the case when in p0-p1 occurs fault1						
The initial weight for all cases at p1 is (W11)	fault1 in p0-p1 is followed by fault1 in p1-p2	fault1 in p0-p1 is followed by fault2 in p1-p2	fault1 in p0-p1 is followed by fault3 in p1-p2	The Euclidean distance between		
	Fault1 (W12)	fault2 (W22)	Fault3 (W32)	W12 and W11	W22 and W11	W32 and W11
-0.00065	-0.00018	-0.00013	-0.00376			
-0.00070	-0.00065	-0.00064	-0.00979			
-0.00020	-0.00012	-0.00012	-0.00222	0.1412	0.1760	0.6007
-0.00069	-0.00058	-0.00058	-0.00833			
-0.00085	-0.00046	-0.00041	-0.00759			
-0.00025	-0.00020	-0.00019	-0.00318			
-0.13110	0.01008	0.04487	0.46942			

The final weights at p2, in the case when in p0-p1 occurs fault2						
The initial weight for all cases at p1 is (W21)	fault2 in p0-p1 is followed by fault1 in p1-p2	fault2 in p0-p1 is followed by fault2 in p1-p2	fault2 in p0-p1 is followed by fault3 in p1-p2	The Euclidean distance between		
	Fault1 (W12)	fault2 (W22)	Fault3 (W32)	W12 and W21	W22 and W21	W32 and W21
0.00008	-0.00016	-0.00008	-0.00240			
-0.00041	-0.00064	-0.00060	-0.00630			
-0.00007	-0.00011	-0.00009	-0.00148	0.1107	0.0759	0.3481
-0.00029	-0.00057	-0.00055	-0.00531			
-0.00015	-0.00044	-0.00036	-0.00484			
-0.00007	-0.00019	-0.00017	-0.00206			
0.12081	0.01008	0.04487	0.46873			

The final weights at p2, in the case when in p0-p1 occurs fault3						
The initial weight for all cases at p1 is (W31)	fault3 in p0-p1 is followed by fault1 in p1-p2	fault3 in p0-p1 is followed by fault2 in p1-p2	fault3 in p0-p1 is followed by fault3 in p1-p2	The Euclidean distance between		
	Fault1 (W12)	fault2 (W22)	Fault3 (W32)	W12 and W31	W22 and W31	W32 and W31
-0.00012	-0.00014	-0.00004	-0.00158			
-0.00047	-0.00062	-0.00056	-0.00395			
-0.00011	-0.00010	-0.00008	-0.00101			
-0.00028	-0.00055	-0.00052	-0.00330	0.2928	0.3276	0.7512
-0.00024	-0.00042	-0.00033	-0.00307			
-0.00019	-0.00019	-0.00015	-0.00131			
-0.28268	0.01008	0.04487	0.46847			

On the basis of such a prepared information, now let us assume a new-fault-state of the system in the predefined fault interval [80, 160], with this fault interval divided in to two sub-intervals [80, 120], and [121, 160].

In [80, 120] the new-state is defined as:

$$New\_state\ (k) = y(k) + (rand(1) * 7 - 3.5) \qquad (11)$$

and in [121, 160] it is defined as:

$$New\_state\ (k) = y(k) + (rand(1) * 20 - 10) \qquad (12)$$

where *y(k)* is as in (7), k takes values in [80, 120] in the first case and in [121, 160] in the second case. Our am is to point out which of the faults in the database constitutes this new-system-state.

The new-state is identified by a neural network of initial weight vector values and size defined as before.

Table 3 shows the weight vector values at the point $p_0 = 80$, $p_1 = 120$, $p_2 = 160$ during the process of the identification of this new-system-state, and the Euclidean distance between the weight vectors at the points p_0- p_1, and p_1- p_2.

Table 3. the weight vector values at the different points, and the Euclidean measures, during the process of the identification of the new-system-state.
Tabela 3. Wartość wektorów wag w różnych punktach, i odległość Euklidesowa obliczona w procesie identyfikacji nową założoną stan układu.

The new-system state				
The initial weight at p0 (WNew0)	The final weight at p1 (WNew1)	The final weight at p2 (WNew2)	The Euclidean distance between	
			WNew0 and WNew1	WNew2 and WNew1
0.00445	-0.00034	-0.00135		
0.00894	-0.00046	-0.00136		
0.00199	-0.00008	-0.00043	0.04993	0.32638
0.00672	-0.00057	-0.00131		
0.00190	-0.00062	-0.00164		
0.00594	-0.00011	-0.00057		
0.00818	-0.03958	-0.36596		

Comparing the Euclidean distance between the pair of the vectors at the points p_0 (Wnew 0), and p_1 (Wnew 1), in the table 3 with the corresponding Euclidean distance in the Table 1 we get:

- from the table 3 the Euclidean distance between the vectors at the points p_0 and p_1 is 0.04993
- from the table(1) the Euclidean distance between the vectors at the points p_0 and p_1 we have (0.14008 0.11352 0.29121); from these values, the nearest value to 0.04993 is 0.11352 this indicates that, in the first fault sub-interval [80, 120], occurred fault2.

The knowledge that fault 2 occurred in the first sub-interval simplifies the problem of finding which of the faults occur in the second sub-interval, i.e., from the Tables 2 we consider only Table 2b.

From the Table 3 the Euclidean distance between the vectors at the points p_1 (Wnew1) and the point p_2 (Wnew2), is 0.32638 from the Table 2b the Euclidean distance between the vectors at the points p_1 and p_2 we have (0.1107 0.0759 0.3481); from these values, the nearest value to 0.32638 is 0.3481, this tells us that, the fault that occurred in the second-sub-interval is fault3.

Hence, our new-assumed system is composed of fault 2 and fault 3 of the database faults, in the assumed fault interval [80, 160].

5. Conclusion

In this work, a fault detection method in dynamic systems, which assumes the occurrence of the fault in a fixed pre-defined fault-interval, is presented. The paper gives the solution method, in the case, when the fault interval is divided in to different fault-sub-intervals and the dynamic system under consideration is assumed to be under the influence of two or more of the faults from the database, in the whole of the fault-interval. Only one of the faults is assumed to occur in each of the fault-sub-intervals. The proposed fault detection method relies, on computing the Euclidean distances between the output layer weight vectors of the model RMLP type neural networks, registered at the beginning point and the end point of each of the fault-sub-intervals. Based on the computed Euclidean measures, a fault model in the database, which has the minimum difference to the Euclidean distance of the new-system- state model, is, concluded to occur.

References

[1] Duch W, Korbicz J, Rutkowski L, Tadeusiewicz R.: Biocybernetyka i inzyniera biomedyczna vol. 6 Sieci neuronowe, Akademicka oficyna wydawnicza EXIT. Warszawa 2000

[2] Dzielinski A.: Modelowania i sterowanie ukladow nieliniowych metodami neuropodobnimi, Oficyna Wydawnicza Politechniki Warszawskiej. Warszawa 2002.

[3] Kaczorek Tadeusz: Adaptation algorithms for 2-D feed forward neural networks, IEEE transaction on neural networks Vol.6, No 2 1995.

[4] Korbicz J., Koscielny J.M., Kowalczuk Z., Cholewa W.: Diagnostyka procesow (Modele metody sztucznej inteligencji zastosowania), Wydawnictwa Naukowo-Techniczne. Warsawa 2002

[5] Osowski Stanislaw: Sieci neuronowe w ujeciu algorytmicznym. (In Polish), Wydawnictwa Naukowo-Techniczne. Warszawa 1996.

[6] Getachew Alemu Wondim: An alternative approach to fault detection in dynamic systems, on the basis of output layer weight values comparison of the model RMLP type neural networks, Measurement Automatics and Control. No. 11, November 2003. Poland.

Tytuł: Metody wykrywania uszkodzeń w układach dynamicznych, na podstawie miary Euklidesowej między wektorami wag wyjściowych modeli neuronowych

Artykuł recenzowany