Mathematical & Computational Applications, Vol. 8, No. 3, pp. 279-286, 2003

THE MONITORING SIMULATION OF A LATHE

George C. Balan and Calin Tarau
Universitatea *Dunarea de Jos* din Galati
Str. Domneasca nr. 47 Galati - 6200 ROMANIA
George.BALAN@ugal.ro

Abstract- The study of machine-tool dynamic is realised here as "monitoring", meaning checking and improving the functioning of the machine. The state of processing is followed with certain sensors which signs are processed inside the computer, then it takes the decision of monitoring, meaning the identification of a class from the set of classes (process conditions). For monitoring in turning, it is shown the classes (tool conditions).
The vibrograms that represent: - the components variations of the cutting force; - the relative displacement between tool and piece, on the repelling direction; - the power furnished by the electric engine, are realised with the functions *RANDN* and *RAND* (from MATLAB). Based on them it is calculating 11 monitoring indices. The class resulted at the experiment (simulate) *i*, which corresponds to the monitoring indices, we establish with the function *REM*. The ANN with 11 inputs (the number of monitoring indices) and *8* outputs (the number of classes) is realised with 3 layers. The network is made with the function *newff*, trained with the function *train* and simulated with the function *sim*.

Keywords- Simulation, monitoring, lathe, artificial neural network, learning, classification, MATLAB.

1. INTRODUCTION

The study of machine-tool dynamic is realised here as "monitoring", meaning checking and improving the functioning of the machine.

The state of processing is followed with certain sensors which signs are processed inside the computer, then it takes the decision of monitoring, meaning the identification of a class from the set of classes (process conditions) $c = [\, c_1, c_2, \ldots, c_n \,]$, according to:

$$\text{if} \quad t_{inf} < x \le t_{sup} \;, \quad \text{then} \quad c = c_i \;, \tag{1}$$

where x is the set (vector) of monitoring indices $x = [\, x_1, x_2, \ldots, x_m \,]$, and t – the admissible limited values. Example of index: x_2 = the mean value of the cutting force; class example: c_3 = the tool wear in certain limits.

The samples can be registered as in table 1, in which m is the number of monitoring indices, n is the number of classes, and N - the number of samples.

It can be introduced the function $Q : c \to x$, which is "obscure" because on it you cannot do but indirect measurements which are, or you can assume they are bound to the function. If for Q you cannot obtain any theoretical relation, you can use in consequence a method of interpretation of data which involves two phases: *learning* and *classification*. In the learning phase it is formed an empirical relation between x and c

for a set of samples in which both x and c are know (a part of data from the table 1). In the classification phase it is used the other part of the table 1, with a view to predicting of c, testing in this way and adjusting the empirical relation.

Table 1. The samples

Sam - ples	Monitoring indices						CLASSES (Process conditions)
	x_1	x_2	.....	x_i	......	x_m	
x_1	$x(1,1)$	$x(1,2)$	....	$x(1,i)$	....	$x(1,m)$	$c(x_1) \in [c_1, c_2, \ldots, c_n]$
x_2	$x(2,1)$	$x(2,2)$	....	$x(2,i)$	....	$x(2,m)$	$c(x_2) \in [c_1, c_2, \ldots, c_n]$
...	...	...	...	...	...	...	.....
x_N	$x(N,1)$	$x(N,2)$	....	$x(N,i)$	....	$x(N,m)$	$c(x_N) \in [c_1, c_2, \ldots, c_n]$

Now the empirical relation is able to classify a new sample x in a certain class c_k.

So Q can have different aspects: an analytical one, an artificial neural network (ANN), etc.

For monitoring in turning, in table 2 it is shown the classes (tool conditions), in which to the first three classes the working conditions are normal and the other ones are unnormal [2].

Table 2. The classes in turning

Class	Tool conditions	Identification on cutter	Identification on workpiece
c_1	Normal	$VB < 0.1$ mm , or $VB_{max} < 0.2$ mm	---------
c_2	Slight wear	$0.11 < VB < 0.2$ mm , or $0.21 < VB_{max} < 0.4$ mm	---------
c_3	Medium wear	$0.21 < VB < 0.3$ mm , or $0.41 < VB_{max} < 0.6$ mm	---------
c_4	Severe wear	$0.31 < VB < 0.4$ mm , or $0.61 < VB_{max} < 0.7$ mm	---------
c_5	Tool breakage	$VB > 0.41$ mm , or $VB_{max} > 0.71$mm	---------
c_6	Chatter	Fresh tool	Chatter marks
c_7	Transient cutting	Fresh tool	An axial slot
c_8	Air cutting	---------	---------

2. THE MONITORING INDICES

In order to obtain the monitoring indices you have to use sensors which measure the three components of the cutting force (F_x - the force on the direction of advance, F_y - the repelling force , F_z - the main force) , the accelerations of cutter holder vibrations (a_x , a_y , a_z) and the power given by the electric engine (W). The signs of the sensors are registered simultaneously and they are sampled.

The vibrograms that represent:

- the components F_x , F_y , F_z variations of the cutting force;
- the relative displacement y between tool and piece, on the repelling direction;
- the power W furnished by the electric engine,

are realised with the functions *RANDN* and *RAND* (from MATLAB).

RANDN generate normally distributed random numbers, with *mean zero, variance one* and the probability density function is Gauss'curve. *RAND* generate uniformly distributed random numbers on the interval [*0, 1*], so the probability density function is uniformly and the repartition function is linear. Both functions produce pseudo- random numbers. The sequence of numbers generated is determined by the state of the generator. Since MATLAB resets the state at start-up, the sequence of numbers generated will be the same, unless the state is changed. For experiments simulation (which involve the same conditions), to the random number series which are generate it is controlled an initialising parameter "*state*" of the generator: *RAND* ("*state*", *J*), for integer J resets the generator to its J -th state.

Considering the relations between the cutting force components [1]:

$$F_x = (0.25 \div 0.35)\, F_z \quad ; \quad F_y = (0.35 \div 0.50)\, F_z \ ,$$

we 'll take: $Fz = 100 + 20\,.*\, randn\,(1\,,N)$; $Fy = 42 + 10\,.*\, randn\,(1\,,N)$,

$$Fx = 30 + 6\,.*\, randn\,(1\,,N)\ , \tag{2}$$

in which N is the number of samplings.

Regarding the relative displacement: $y = 1.5\,.*\, randn\,(1\,,N)$. (3)

The cutting power: $W = 25 + 5.*rand\,(1\,,N)$. (4)

The monitoring indices are:

$X_1 = \overline{F} = (\overline{F}_x^2 + \overline{F}_y^2 + \overline{F}_z^2)^{1/2} \rightarrow$ the mean value of the cutting resulting force (where $\overline{F}$ is the mean value);

$X_2 = C_F = [\,max\,(F_y) - min\,(F_y)\,]\,/\,F_y \rightarrow$ the crest factor of force F_y ;

$X_3 = \dfrac{\overline{F}_y}{\overline{F}_v} = \dfrac{\overline{F}_y}{\sqrt{\overline{F}_y^2 + \overline{F}_z^2}} \rightarrow$ the rate of mean forces from the vertical plane;

$X_4 \rightarrow$ the rate of crossing the mean for force F_y (the number of oscillogram F_y crossings with its mean value);

$X_5 \rightarrow$ the mean of power spectral density of F_y in the frequencies band *1 - 125* Hz;

$X_6 \rightarrow$ the mean of power spectral density of F_y in the frequencies band *126 - 250* Hz;

$X_7 \rightarrow$ the mean of power spectral density of F_y in the frequencies band *251 - 500* Hz;

$X_8 \rightarrow$ the mean of power spectral density of vibration y in the frequencies band *0 - 125* Hz;

X_9 $\rightarrow$ the mean of power spectral density of vibration y in the frequencies band *126 - 250* Hz;

X_{10} $\rightarrow$ the mean of power spectral density of vibration y in the frequencies band *251 - 500* Hz;

X_{11} $\rightarrow$ the mean value of the cutting power W.

The matrix P, of dimensions ($11 \, x \, Q$) – which contain *11* monitoring indices ($X_1 - X_{11}$) and where Q is the number of experiments, have the form from relation (5).

P = 1.0e+004 *

(Columns 1 through 4)

0.01115160236262	0.01111776208232	0.01128320014418	0.01114477828469
0.00011872250749	0.00013863976981	0.00013464454684	0.00014977366854
0.00003837450623	0.00003824647822	0.00003865774920	0.00003882892721
0.01240000000000	0.01400000000000	0.01200000000000	0.01320000000000
1.19649632745768	1.14774467329114	1.19195116081064	1.16814906208573
0.01449785276532	0.00834934135711	0.01067465623032	0.01128964235457
0.00917486420559	0.01114635004294	0.01009532191768	0.01272595316720
0.00019465249851	0.00038721111922	0.00027689491860	0.00017997231379
0.00018259338286	0.00027753290547	0.00026326336937	0.00017247427052
0.00019509725158	0.00028420715164	0.00026315476562	0.00022039394774
0.00274914751134	0.00275807333095	0.00274997052944	0.00274503600945

(5)

3. THE POWER SPECTRAL DENSITY

For the evaluation of the power spectral density yields from the random process achievements $Y(t)$, on a finite interval $[-T,T]$: $x^{(1)}_T(t), x^{(2)}_T(t), \ldots, x^{(n)}_T(t)$; $x^{(i)}_T(t) = 0$ out of the interval $[-T,T]$. The direct Fourier transform is applied to these achievements:

$$X^{(i)}_T(j\nu) = F[x^{(i)}_T(t)] = \int_{-\infty}^{+\infty} x_T^{(i)}(t) \cdot e^{-j2\Pi \nu t} \cdot dt \quad , \; (i = 1, 2, \ldots, n) \; . \qquad (6)$$

We form the sizes $\frac{1}{2T}\left| X_T^{(i)}(j\nu)\right|^2, (i = 1, 2, \ldots, n)$, named *periodograms*.

The power spectral density P_{xx} is the limit (when $T \rightarrow \infty$) of the periodograms set mean:

$$P_{xx}(j\nu) = \lim_{T\to\infty} \sum_{i=1}^{n} \frac{1}{2T}\left| X_T^{(i)}(j\nu)\right|^2 . \qquad (7)$$

Power spectral density we calculate with function *PSD*: *Pxx = PSD (X , NFFT , Fs , WINDOW)* which estimates this characteristic of a signal vector *X* using Welch's averaged periodogram method. *X* is divided into overlapping sections, each of which is detrended, then windowed by the *WINDOW* parameter, then zero-padded to length *NFFT*. The magnitude squared (of the length *NFFT*) DFTs (Direct Fourier Transforms) of the sections are averaged to form *Pxx*. *Pxx* is length *NFFT / 2 +1* for *NFFT* even , or *(NFFT + 1) / 2* for *NFFT* odd. If you specify a scalar for *WINDOW* a **Hanning** window of that length is used. *Fs* is the *sampling frequency* which doesn't affect the spectrum estimate, but is used for scaling of plots: $Fs = 1/\Delta$, where Δ is the sampling period.

Shannon's sampling theorem says: $Fs \geq 2\, f_{max}$, where f_{max} is the higher frequency from the signal spectrum. In [3] it is shown that in the case of turning $f_{max} = 500\ Hz$, so $Fs = 1000\ Hz$.

[Pxx , F] = PSD (X , NFFT , Fs , NOVERLAP] returns a vector of frequencies the same sizes as *Pxx*, at which the *PSD* is estimated and overlaps the sections of *X* by *NOVERLAP* samples.

The default values of the parameters are: *NFFT = 256* (or *LENGTH (X)* , whichever is smaller) *NOVERLAP = 0, WINDOW = HANNING (NFFT), Fs = 2*. You can obtain a default parameter by leaving it off or inserting an empty matrix *[ ]*.

So, we will used: *[Pxx , f] = psd (x , [ ] , 1000)* , where *length (Pxx) = 256/2+1 = 129* , therefore the elements from *Pxx* corresponding to the frequencies of monitoring indices $X_5 \div X_{10}$ we determine through the relations : $125 / u = 250 / v = 500 / 129 \rightarrow u = 32\ ;\ v = 64$.

The units of the power spectral density are such that, using *Parseval*'s theorem:
SUM (Pxx) / LENGTH (Pxx) = SUM (X.^2) / LENGTH (X) = COV (X). (8)
The *RMS* value of the signal is the square root of this: $RMS\,(X) = (COV\,(X))^{1/2}$.
If the input signal is in *Volts* as a function of time, then the units of *Pxx* are *[Volt^2 / Hz]*.

The power spectral densities for the force F_y and for the relative displacement y are shown in figure 1.

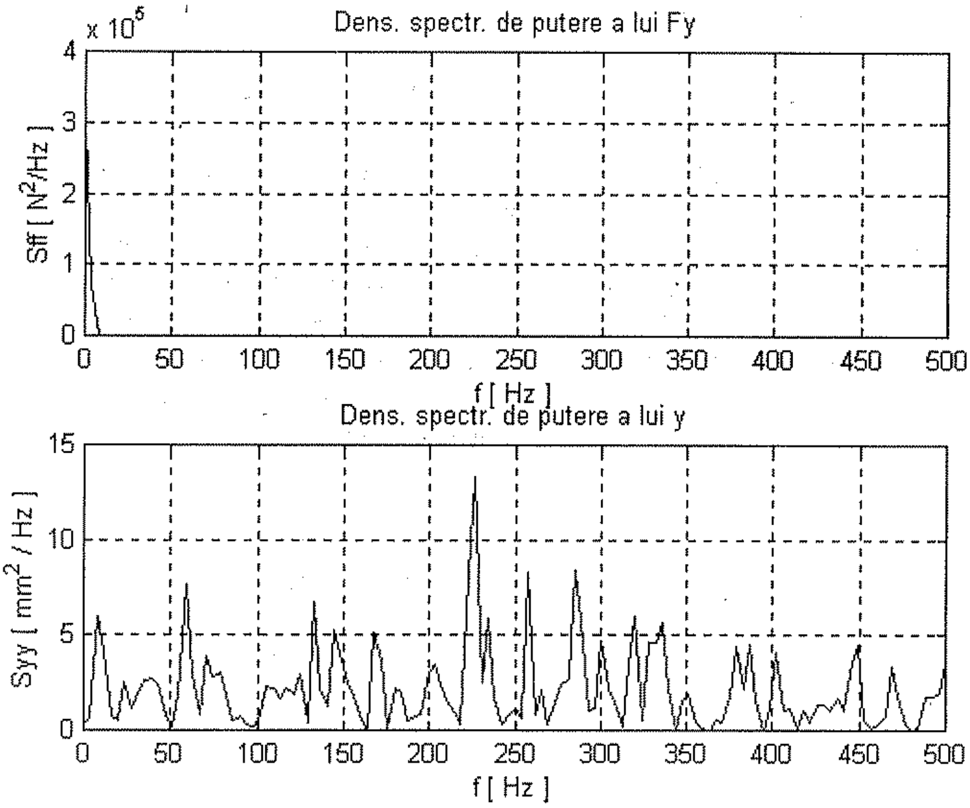

Figure 1. The power spectral densities

4. THE SIMULATION

The class resulted at the experiment (simulate) i, which corresponds to a serie of 11 monitoring indices, we establish with the function *REM*, in a MATLAB snapshot:

```
T = zeros ( 8 , Q ) ;
for i = 1 : Q
    in = ( i - 1 ) * N ;
    cl = rem ( in , 8 ) + 1 ;
    T( cl , i ) = 1 ;
end
```

where: Q = the experiments number, N = the samples number, 8 = the classes number, T = matrix of dimensions $(8 \times Q)$ in which the *rank* of the not null element from i column is the resulted *class* (one of the 8 classes).

For $Q = 20$ it is resulted the matrix T from relation (9).

$T =$

(Columns 1 through 20)

1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0

(9)

The ANN with $r = 11$ inputs (the number of monitoring indices) and $s_3 = 8$ outputs (the number of classes) is realised with 3 layers, having $s_1 = 15$ neurones on the first layer, $s_2 = 19$ neurones on the second layer and $s_3 = 8$. The activation functions are: *tansig* for the first two layers and *logsig* for the last one. The network which is made with the function *newff*, trained with the function *train* (with *learning rate* $lr = 0.003$) and simulated with the function *sim*, has the evolution from figure 2.

The network furnishes at the output the matrix yr, having the dimensions $(8 \times Q)$ like the matrix T. A part of matrix yr it is shown in relation (10).

$yr =$

(Columns 8 through 14)

0.0064	0.0000	0.1623	0.0000	0.0043	0.0002	0.0000
0.0000	0.0000	0.0003	0.0000	0.0000	0.0000	0.0000
0.8282	0.0128	0.0145	0.0006	0.1757	0.0001	0.0463
0.0039	0.0000	0.6531	0.0004	0.3958	0.0007	0.0000
0.0002	0.0003	0.0000	0.1537	0.0001	0.0563	0.0091
0.8925	0.0930	0.0020	0.0030	0.3629	0.0022	0.0179
0.0000	0.1384	0.0017	0.6219	0.0004	0.1858	0.0128
0.0174	0.9997	0.0035	0.0180	0.0030	0.0016	1.0000

(10)

The **rank** of the **maximum** element from the column i is the class furnished by ANN.

5. CONCLUSION

From comparison of the matrix columns T (9) and yr (10) we can observe that the classification has succeeded only in the case of columns *8, 10, 11, 14.* In figure 3 are shown the target ("o") and the ANN outputs ("x"); from the total of $Q = 20$ *columns*

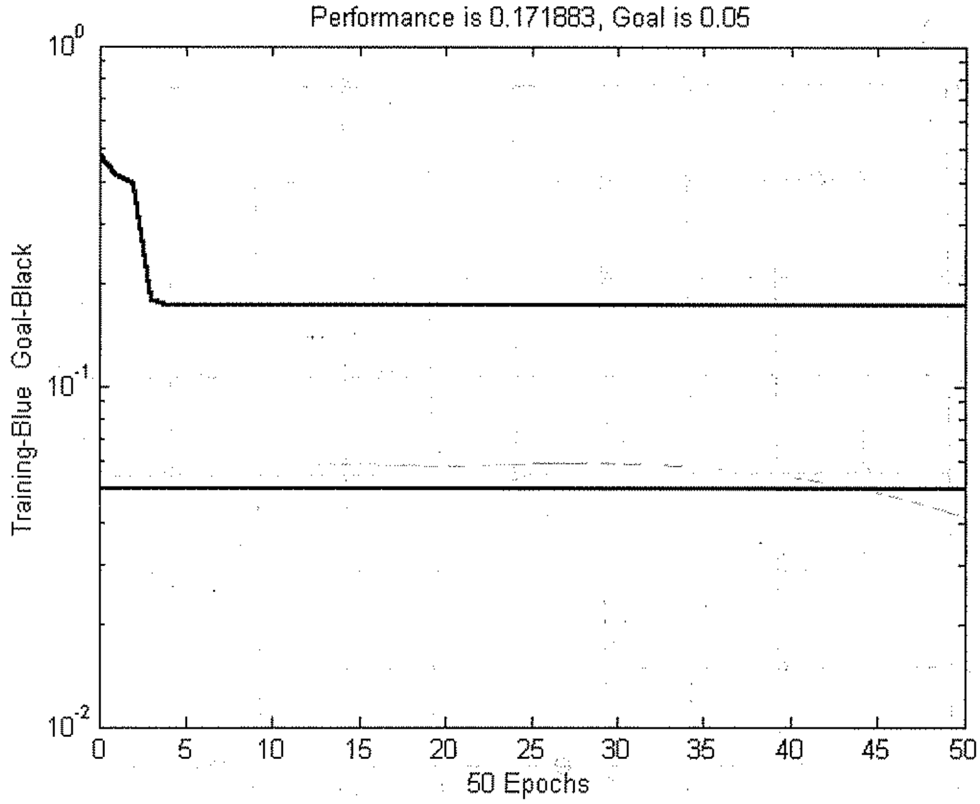

Figure 2. The network evolution

the classification has succeeded for *8* columns, so the success rate is $8 / 20 = 0{,}4 = 40\ \%$

For this moment this rate satisfies us, because the classification is **not** realised for a real case. In a real case, the ANN performance will be improved by:

- the increase of learning rate (lr) , of the neurones number (s_1, s_2), of the epochs number and of the layers number;
- the change of the activation functions and of weights and bias initialising point.

REFERENCES

1. G. Balan, *Contributii la studiul stabilitatii neliniare si stocastice a proceselor de aschiere la danturare , strunjire si frezare*, Ph. D. Thesis, Polytechnic Institute, Iasi, 1982

2. G. Balan, Fuzzy Systems used in the Study of Machine - Tools Dynamic, *New Developments on Modelling and Manufacturing Process Control. Proceedings of the*

International Conference on Advanced Materials Processing Technologies (AMPT'01), Vol. I I, Sep. 2001, Universidad Carlos I I I de Madrid, Leganes, Spain, pp. 1033-1038.
3. R. Du, M. A. Elbestawi and S. M. Wu, Automated Monitoring of Manufacturing Processes, Part 1: Monitoring Methods, Part 2: Applications, *ASME Journal of Engineering for Industry*, may, vol. 117, 1995, Part 1-pp. 121 - 132, Part 2 – pp .133 - 141.

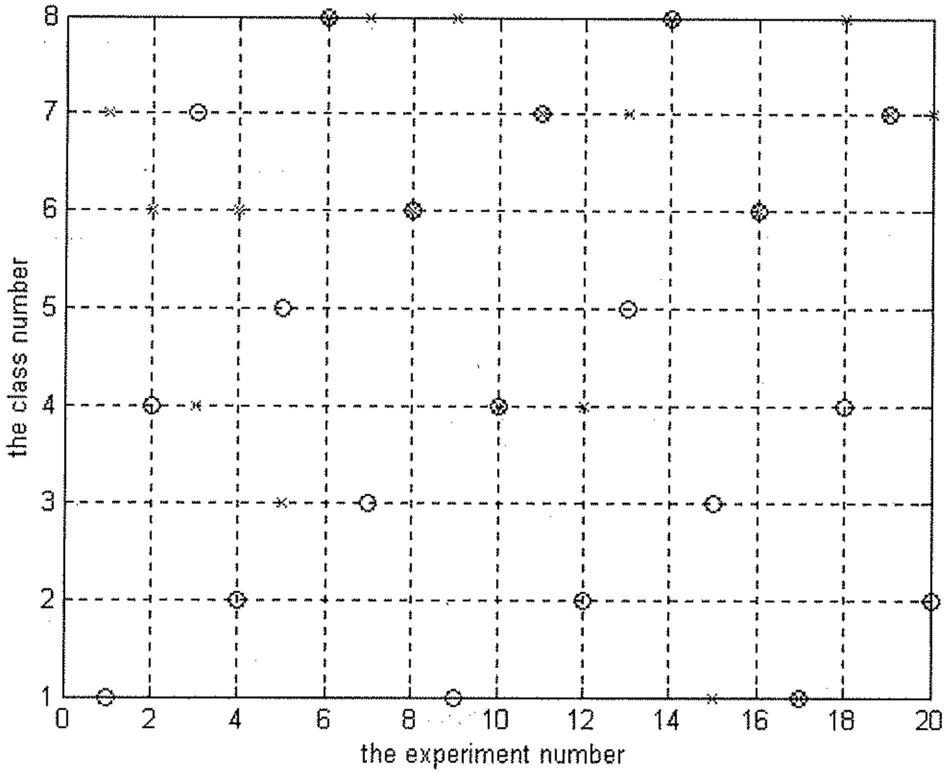

Figure 3. The comparison between target ("o") and the ANN outputs ("x")