JOURNAL OF THEORETICAL
AND APPLIED MECHANICS
3, **40**, 2002

SMART SENSOR FOR OPERATIONAL LOAD MEASUREMENTS

Tadeusz Uhl
Maciej Petko

University of Mining and Metallurgy, Cracow
e-mail: tuhl@rob.wibro.agh.edu.pl; petko@agh.edu.pl

The important technique enabling machine health monitoring and fault localisation is operational load measurement; however direct measurement is usually difficult or even impossible. The paper deals with the realisation of the idea of developing "smart sensor" which estimates the load based on structure response. In the case study, hardware neural network has been used to obtain the load course from the vibrations in specific points of the structure. The paper covers also problems with prototyping and implementation stages during development of signal processing algorithms with emphasis placed on ASIC/FPGA based hardware platform, for which a methodology of implementation is formulated and validated by practical application to the smart sensor problem. Details of the procedure are presented along with the tools used and results obtained during its realization. Performance of the device during experiment is analysed and, finally, conclusions are shown.

Key words: smart sensor, mechatronics, neural networks

1. Introduction

The aim of a diagnostic procedure of mechanical structures is to determine their technical state and if is not satisfactory, to localise faults and assess their size. One of the most commonly used diagnostics of rotating machinery is vibrodiagnostics that consist of vibration measurements during operation and estimation of a chosen state symptom. As a state symptom some signal estimates like mean value of amplitude, spectrum, RMS, p-p value can be used. The vibration can be measured in a continuos way by monitoring the system or can be measured periodically with a given state dependent period (Uhl, 1999). When during the testing the vibration level exceeds given standards or manufacturer recommendations reasons should be identified. The proposed diagnostic procedure is schematically shown in Fig. 1. There are two basic reasons of increasing vibration levels of operating machinery:

- too big excitation level caused by machinery faults,
- resonance of the structure.

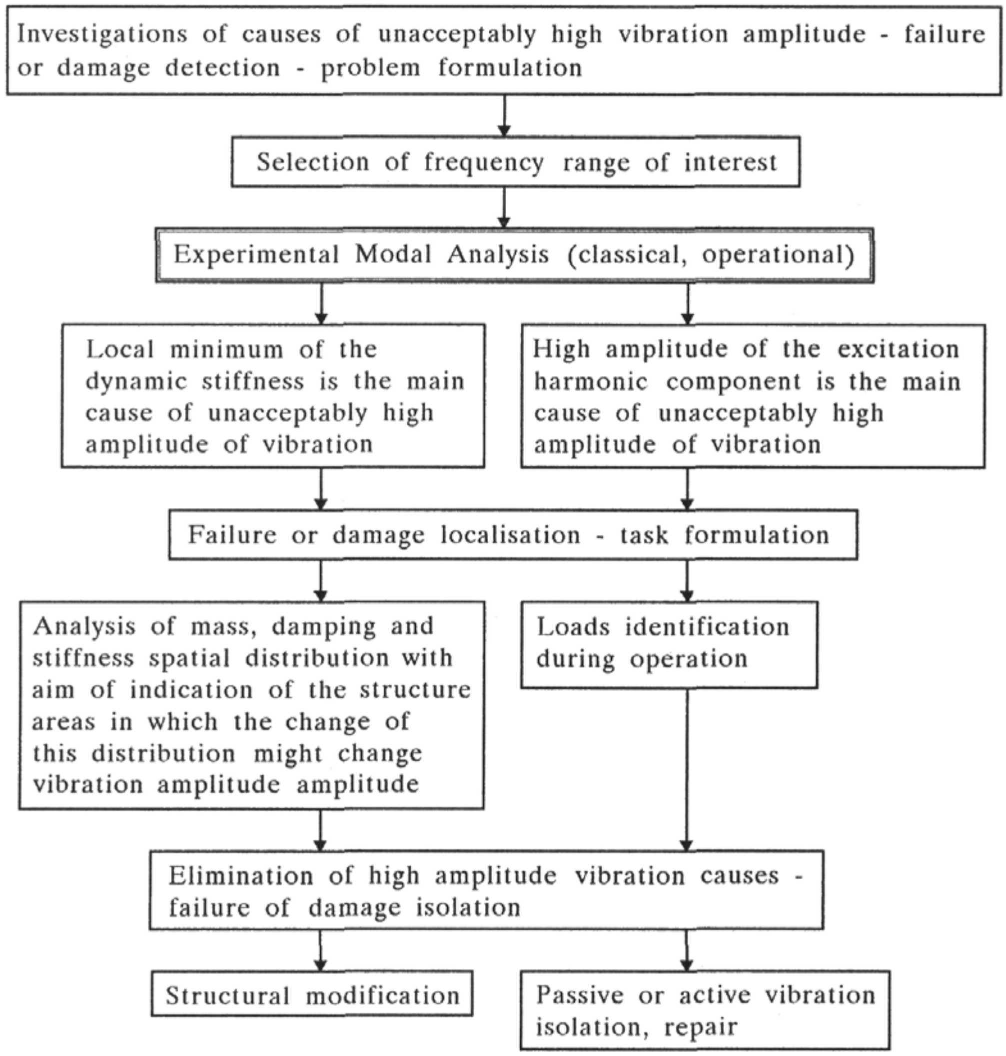

Fig. 1. Diagram of investigation of unacceptably high amplitude vibration

The considered procedure is composed of 2 parallel subprocedures aiming at determination of causes of unacceptably high amplitude vibrations i.e. failure or damage. This vibration might be caused by inappropriate structural properties of the structure and/or excessive excitation amplitude.

The left branch of the diagram corresponds to analysis of structural properties of the considered object. In this case it is assumed that the reason for high vibration amplitude is a local minimum of the dynamic stiffness for some frequency subrange, and then the analysis of mass, damping and stiffness spatial distribution is performed with the aim of indication the structure areas, in which the change of this distribution might bring a decrease in the vibration amplitude. As a result, an appropriate structural modification is formulated. Usually, one encounters the problem of technical realisation of the formulated modification that has to take into account: properties of real construction materials, machining and assembly techniques as well as operational and economical constraints.

The right branch of the procedure deals with the situation in which the high vibration amplitude is caused by some source located in the considered structure or nearby. In such a case the amplitudes and spatial distribution of usually immeasurable operating loads as well as vibration transfer paths should be identified. As a result of the performed analysis the appropriate modification of the vibration source or its connection with the structure, or the connection between the structure and its boundary should be formulated.

Direct measurements of operational loads is very difficult and cost consuming task. Sometimes it is impossible to measure directly loads for given structure elements, or this measurement requires a special design. But it is easier to measure structural response caused by loads which should be known. Based on the knowledge of the response, in some cases, it is possible to determine external loads. This requires special procedures for identification of loading forces for operating structures based on response measurements. If the loads of the structure are known the essential advantage that can be achieved is usage of monitoring based on load cycles analysis. Sometimes it is impossible or impractical to use special transducers that are installed on a structure to measure the system response. Therefore methods based on process parameter measurements are recommended. The idea of identification of the load parameters based on the response measurements is schematicall shown in Fig. 2. But in many industrial cases the relations between the loads and process parametrs are very complex and difficult for analytical formulation. Relations between process data and load cycles of structures are commonly nonlinear and very difficult for analytical modelling. These reasons are the main ones in choosing neural networks as a basic tool for identification of loads based on process parameters measurements.

To fulfil an on-line monitoring and diagnostic system requirements the loads should be identified on-line, as well. This creates special demands for

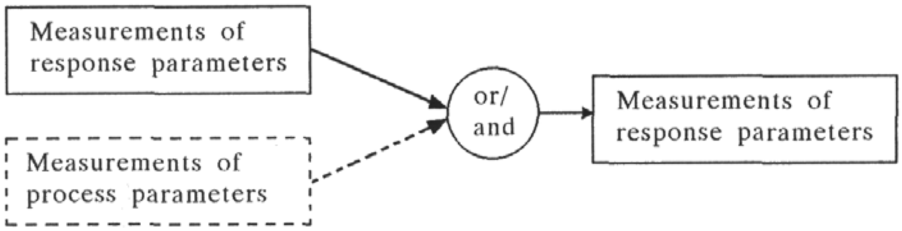

Fig. 2. The idea of loads identification

design and implementation of the identification procedures. The best solution seems to be a special design of smart sensors that make it possible to measure loads indirectly. Such a sensor requires implementation of the response (or process parameters) mapping on the loads. To make the sensor of more general purpose, not only for one case, the sensor should has the ability to learn this mapping based on data from an experiment or model simulation. There exist several different methods for identification of loads in mechanical systems. An overview of these methods is shown in the next section of this text.

2. Operational load measurements

To identify the operational load based on process parameters or system response measurements many methods can be employed (Busby and Trujillo, 19xx; Giergiel and Uhl, 1989a,b; Haas et al., 1995; Hackl, 1985; Li, 1988; Lisowski et al., 2001; Simonian, 1981; Trujillo, 1987; Uhl, 2001; Zion, 1994). This task is somtimes specified as an inverse identification problem. The methods that can be used to solve the inverse identification problem are classified into three main categories:

- deterministic methods
- stochastic methods
- artifical intelligence based methods.

The two main techniques belong to the deterministic methods of load identification:

- frequency domain methods (Giergiel and Uhl, 1989b; Lisowski et al., 2001; Uhl, 2001)
- time domain methods (Busby and Trujillo, 19xx; Giergiel and Uhl, 1989a; Simonian, 1981; Uhl, 2001).

This classification is based on the signal processing methodology applied for experimental data which are necessary to perform an estiamtion of load parameters process.

Basic methods of identification of excitation forces have been formulated for linear systems in which the assumptions of small damping and stationarity of parameters are valid. Methods in the frequency domain require information about Frequency Response Functions (FRF) for investigated structures and spectrum of system responses measured during operation. Based on this information the spectrum of excitation forces can be estimated.

Similar methods are formulated in the time domain, using a relation between the excitation and system responses in a form of convolution. An iterative formula for calculation of the excitation forces in mechanical structures is proved based on the properties of Toeplitz matrix (Giergiel and Uhl, 1989b).

The identification of excitation forces can be performed using the mutual energy theorem formulated by Heaviside in 1892 (Engel, 2000). This method allows one to identify the spectrum of loads based on the response (in a form of vibration velocity) measurements (Hackl, 1985; Li, 1988).

The above given identification methods can be used for systems which are linear but not for nonlinear. For both linear and nonlinear systems, methods based on minimization of a given objective function can be employed. Mainly, the least square error between the simulated and measured system response is used as the objective function in this identification method. The dynamic programming optimization method formulated by Bellman (Giergiel and Uhl, 1989a; Simonian, 1981) is commonly used for minimization of the objective function to estimate the excitation forces. Some examples of application of these methods are given by (Giergiel and Uhl, 1989a; Trujillo, 1987). A similar approach based on genetic algorithms is presented in Giergiel and Uhl (1989b).

A statistical approach is formulated using statistical models of the relation between the system response or process parameters and opeartional loads of the structure. The approach based on the regression model of the relation between loads and flight data (data recorded using a standard flight recorder) is presented for a helicopter in Haas et al. (1995), Zion (1994). The quantitative measure of quality of the regression model is a coefficient of determination R^2 that is, for many components of the investigated helicopter type H-53, relatively low 78% (Zion, 1994). It was a motivation to find a better methodology of loads prediction and showed that the linear model was not sufficient to find the loads of particular helicopter components based on the flight data. A proposal of application of neural networks for such data is shown in this paper. But, conducting a regression analysis during the evolution of flight, a loads survey can be a valuable tool in providing a fatigue analyst with understanding of the process parameters that influence the loads. Such information is not available from the neural networks based approach.

3. Concept of a smart sensor for operational force measurements

To measured load of particular structural componenets of a mechanical structure, a special design of the sensor is needed. The sensor should have extended possibility of signal and data processing, specialized for load vector estimation. This creates specification requirements for such a design. The transducer should have microprocessor on board. The processor can be applied for data processing and be programmed or learned, communicated with data acquisition system. The transducers which are built in the processor and have communication capability are named smart sensors (Frank, 2000). The main functions of such sensors are the following:

- collect response signals from local transducers
- perform data processing
- estimate load vector parameters
- send information about the load to the acquisition system.

The scheme of such a sensor design is shown in Fig. 3.

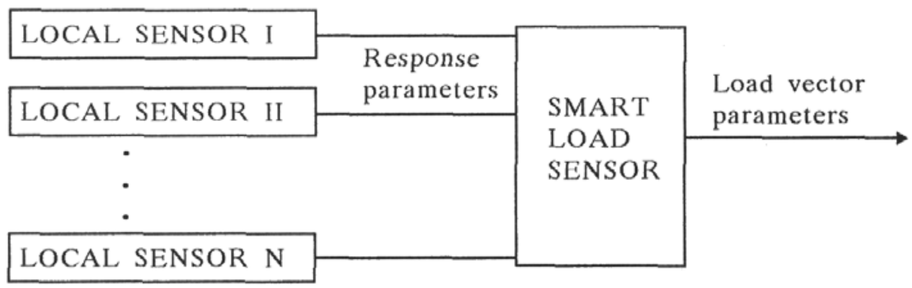

Fig. 3. Scheme of the smart sensor structure

In the case when the acceleration at several points on a struture is measured as the response parameters, typical accelerometers can be used as local transducers. Analog signals from local sensors are collected by a signal acquisition module builid in the smart sensor structure. The signals are converted to the digital form and recorded in the transducer memory buffer. The next step of the procedure implemented in the smart transducer is data processing using a model of the system in a form of a deterministic formula or intelligent algorithm which represents the mapping of the response signals to the load vector. In the case of the intelligent algorithm, mainly neural networks are implemented inside the transducer. The neural network needs to be learned before identifying the load vector. This requires one or more additional inputs to measure the load vector during the learning phase. The learning process can be done outside the sensor and the ready-to-use neural network should be loaded to the sensor memory. The solution based on a microprocessor or ASIC solution can be implemented. The scheme of the internal structure of the proposed sensor is shown in Fig. 4.

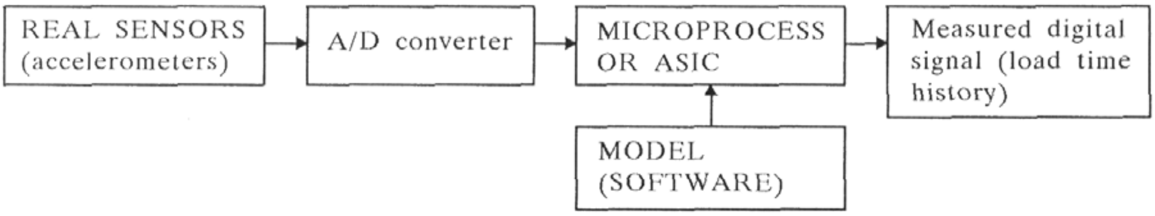

Fig. 4. Structure of the proposed smart sensor

The smart sensor for load measurements based on the neural network algorithm and ASIC/FPGA hardware platform is designed, implemented and tested in the Department of Robotics and Machine Dynamics at the University of Mining and Metallurgy. The inputs are the measured acceleration at two points at the tested structure and the output is the time history of the load which excited the measured vibration acceleration.

4. Case study

In this section the structure of the smart sensor, based on the concept shown above, is discussed. Design verification and performance of the manufactured prototype sensor are tested experimentally by the comparison between the measured vibration excitation force and identification results of the same loading force vector formulated and implemented in the smart sensor procedure.

4.1. Experimental setup

The goal of this analysis was to test the method itself, the efficiency of the force identification algorithms and the software in which the method was implemented. That is why the authors chose as an object of the analysis a steel frame – a simple laboratory testing object. Its shape and main dimensions are shown in Fig. 5.

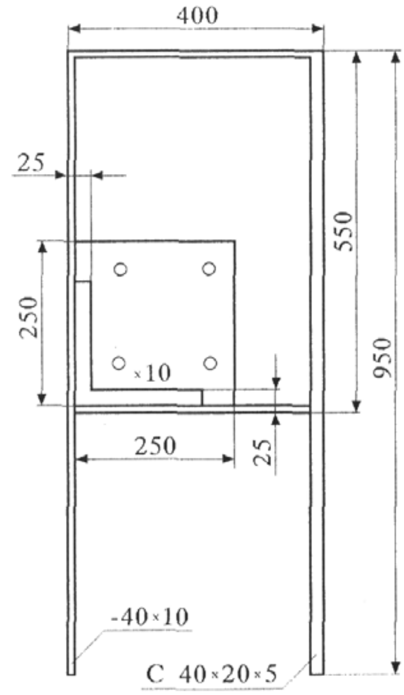

Fig. 5. The steel frame – object of the analysis

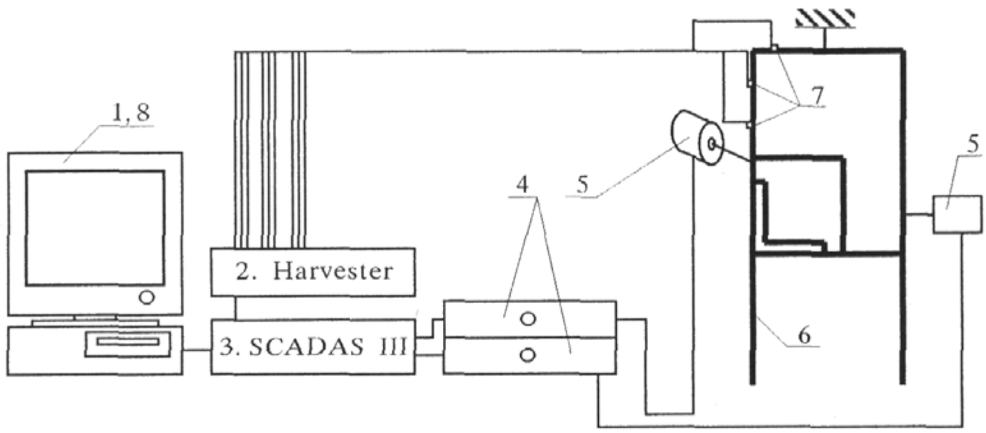

Fig. 6. Scheme of the measuring stand; 1 – workstation HP 9000 715/80, 2 – acquisition server SCADAS III, 3 – acquisition server PCB Harvester F537B05, 4 – power amplifiers, 5 – electrodynamic shakers, 6 – tested object (frame), 7 – accelerometers PCB 330 A, 8 – MIMO FRF, Throughput AM modules of the LMS CADA-X software

The frame was excited by an electrodynamic exciter at one point and the response in a form of accelerations at two points were measured. To verify the performance of the designed sensor the force was directly measured using the force sensor. The experiment was done using the measurment system shown in Fig. 6.

The measurement results have been used for training and testing of the neural networks.

4.2. Data collecting for the training set

For the purpose of the neural network (NN in short) training and performance assessment, some amount of experimental data has been collected. It consists of records of the excitation force and resulting vibrations in several points of the frame structure in different directions, altogether eleven vibration and one excitation signals. Thirty-five experiments have been conducted, each time with different, shaker caused excitation. The excitation signals used can be divided into several groups:

- 7 pure sines of different frequency and amplitude
- 14 two-tones of different frequencies and component amplitude ratios
- 2 chirps
- 6 "square"-waves of different frequency and amplitude
- 6 noises of different bandwide and level.

Some of them are shown in Fig. 7. The force curves differentiate from the described above due to the fame reaction. All signals have been sampled synchronously at 512 Hz for 3 seconds to obtain 3072 time points per experiment.

After normalization to the full scale of the analog-to-digital converter (± 1 range), three subsets, each consisting of 1000 time points, taken from available data, were created: the training, the validation and the test set. The first, consisting of fifteen excitation cases, was used for computing the gradient and updating the network weights and biases, while the second, consisting of ten excitation cases, served for early stopping of the training process to avoid overfitting and to improve generalization ability of the NN (Jang et al., 1997). The test set was not used during training, but for the performance assessment and comparison between different network architectures.

4.3. Selection of neural network architecture

In order to make an implementation phase easier, some constrains have been imposed on the neural network architecture. The choice has been ma-

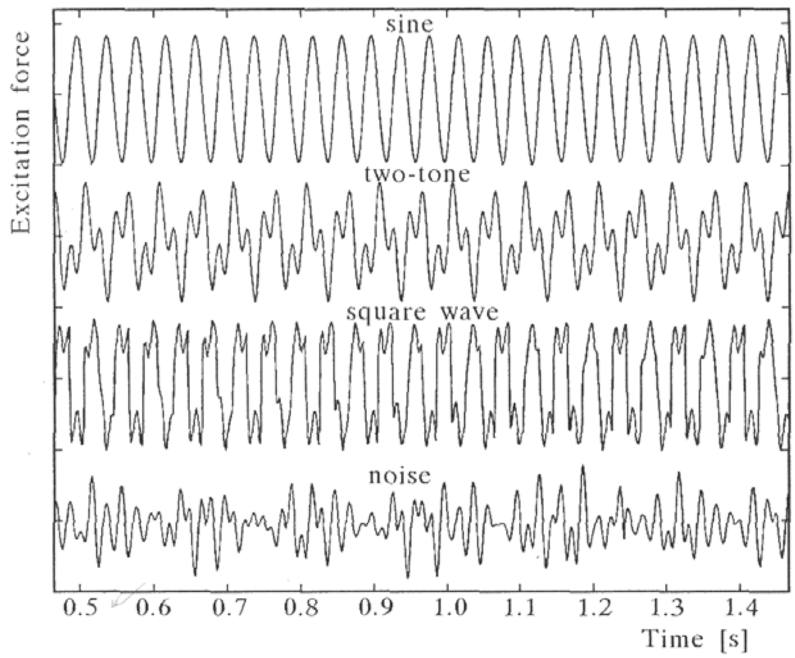

Fig. 7. Excitation forces used in the experiment. In reality, they did not contain a DC component

de between two-layer networks with symmetric sigmoid neurons in the hidden layer and symmetric linear neurons in the output layer. The given task requires a dynamic NN to be applied, so several possibilities have been tested, namely delayed inputs, feedback from a delayed output to input and other forms of recurrence. The input signals have been selected based on a correlation analysis.

All networks have been trained and simulated in Matlab, using Levenberg-Marquardt algorithm with early stopping, based on a validation set of input-target sequences taken from the experiment. Each epoch – the presentation of the whole training set to the network – consisted of fifteen independent time sequences, corresponding to fifteen kinds of excitation.

Decent results have been obtained for NN with the delayed inputs. Further selection, aimed at reduction of the number of: input signals, input delays and neurons in the hidden layer, has given the network with two inputs, ten delays and twenty-two hidden neurons. After careful analysis of its behaviour, it has occurred that some of hidden layer neurons generally worked on the linear part of the transfer function, which motivated the idea of changing the type of the hidden neurons to symmetric saturating linear one, as easier for hardware implementation. Analyses, conducted after short additional training, have shown that it did not degrade the NN performance and that most of the neurons went beyond the linear part of their transfer function during the work, which indicates a nonlinear character of the network. The finally accepted NN architecture is shown in Fig. 8.

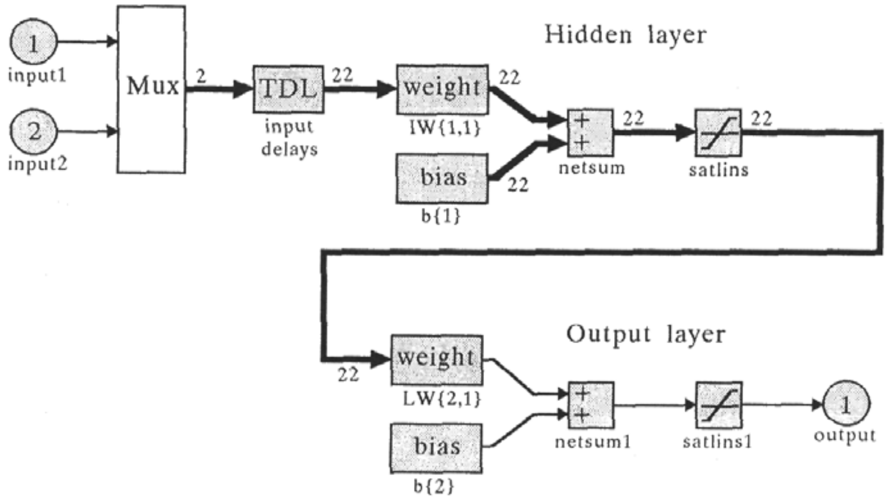

Fig. 8. Final neural network architecture

4.4. Intelligent sensor hardware

From the hardware point of view, the intelligent sensor consists of several parts, shown schematically in Fig. 9. The actual sensing elements – primary transducers – convert a mechanical quantity into electrical one.

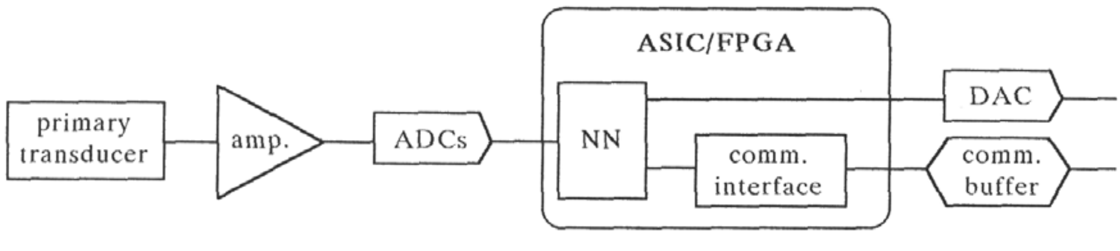

Fig. 9. Intelligent sensor hardware

The resulting signals usually require some kind of amplification or buffering and antialising filtration. Then, they need to be transformed to a sequence of digital samples by means of analog-to-digital converters (ADCs). All digital signal processing takes place in a single integrated circuit, namely ASIC, or, especially during prototyping, in its user programmable counterpart (FPGA). The output of the intelligent sensor can be analog, which requires a digital-to-analog converter, or digital. In the latter case, some mechanism of data exchange – communication interface – has to be implemented in the same ASIC/FPGA chip (Frank, 2000).

At present, piezoelectric accelerometers are used as primary transducers, Altera APEX 20KE family FPGA chip and the system output is of an analog type.

4.5. Hardware implementation of neural network in FPGA

A neural network, as other signal processing algorithms, to be implemented in FPGA/ASIC is, a continuous-time and continuous-value system, described by a set of mathematical equations or by a block diagram. Since such a form is not appropriate for the implementation, it requires several transformations, which influence not only the notation, but also the algorithm itself (Petko, 1999).

First of all the discretisation time is needed. The second substantial transformation is the amplitude discretisation (quantisation). The use of only fixed-point arithmetic allows for significant reduction of the algorithm realisation cost. Nevertheless, care should be taken, because the fixed-point representation can hardly cope with signals of high dynamics (which amplitude changes over several orders of magnitude), which are especially probable to be met in neural networks. Its functioning is also influenced by phenomena peculiar to fixed-point calculations as overflow and saturation. For this reason the performance of the algorithm after the above transformation should be checked.

EDA (Electronic Design Automation) tools used for synthesis of FPGA programming files or ASIC topology accepts chip descriptions in one of HDLs (Hardware Description Language). Coding in HDL is time consuming and an error prone task (Petko, 2001). When using such a notation, the problem arises with testing and verification of the algorithm performance.

In order to solve, or at least reduce the presented problems a procedure of implementation of signal processing algorithms in FPGA/ASIC is being developed. Its present state will follow.

The main objectives during the procedure development were reduction of the number and range of "hand-made" transformations, the use of commercially available software when possible, and trial to keep the part of the code, which does not belong to the algorithm clearly separated. The first objective results from the cost of the hand made conversion – it is time consuming and error prone. The second decreases the cost of the procedure development and allows for taking advantage of new software version improvements immediately, without excessive adaptation efforts. Fulfilling the last objective creates possibility for realisation of fast prototyping in target hardware.

Schematically shown in Fig. 10 procedure consist of a few steps, some of them need to be performed manually, other are automatically done by appropriate software. First transformation – conversion to the fixed-point, discrete time form – is accomplished in Simulink with the use of Fixed-Point Blockset [25]. The latter allows for testing signal dynamics and behaviour of fixed-point calculations, performed with a selected signal representation precision, range, saturation and overflow characteristics. This transformation needs to be done manually, but the parameter values can be tuned up to meet the performance of continuous-time, floating-point algorithm using Nonlinear Control Design Toolbox (NCD) [28], carefully selected cost function and optimisation technique.

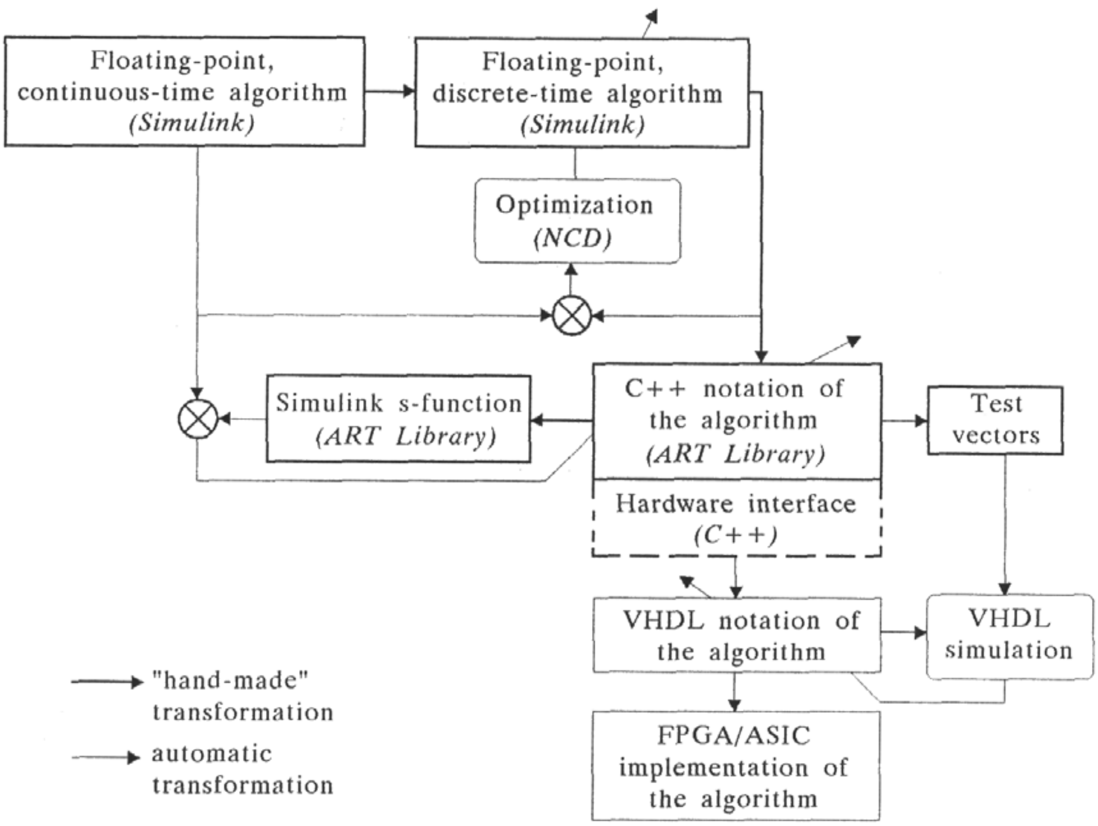

Fig. 10. Procedure of implementation of signal processing algorithm in ASIC/FPGA

The next stage, also manually done, uses software from Frontier Design. Its A|RT Library [24] is a library for several C++ implementations, providing fixed-point classes, which allows for simulation of fixed-point calculations in C++. The C++ code utilizing fixed-point classes can be then synthesized into VHDL or Verilog (two of the most popular HDLs) using A|RT Builder [23]. A|RT Library uses the same types as Simulink's Fixed-Point Blockset blocks, so one-to-one mapping between the block diagram and C++ code is possible, which makes the conversion task easier and reduces errors.

Useful C++ code of an algorithm can be encapsulated by Simulink interface code to create the so-called s-function. Such a s-function defines behaviour of a new block, which can be included into each system model in Simulink. Concurrently, the same s-function source can be synthesized by A|RT Builder into VHDL.

The important advantage of A|RT Builder is its transparency. It means that the generated VHDL code depends only on the C++ programming style. By choosing suitable style, resulting chip topology can be optimised for speed or for area thanks to synthesizing more sophisticated data flow control techniques as pipelining or resource sharing. It is well known which VHDL code is generated for each C++ statement or expression and all identifiers are preserved. It gives high degree of control over the generated VHDL code, at the same time setting free from the hand coding. Moreover, it makes it possible to link a VHDL control algorithm notation with other pieces of VHDL code, e.g. describing hardware interfaces with peripheral devices as Analog-to-Digital Converters (ADC) or Digital-to-Analog Converters (DAC) as well as calculation time frame $(-s)$.

The generated VHDL code of an algorithm, together with VHDL description of necessary peripheral interfaces is then compiled into a gate level netlist, which describes the chip architecture using gates, flip-flops and other primitives available in a particular technology. Later, the gate level netlist needs to be fitted, which means that physical chip resources are to be allocated to each primitive in the netlist. In this case, Synopsys FPGA Express [26] was used as a compiler and Altera Quartus II was used as a fitter.

In the case of necessary manual modification of a VHDL code, some way of checking its consistency is needed. The testing can be performed using VHDL simulators. The creation and verification of test vectors needed for VHDL simulation is supported by A|RT Library.

During the implementation of the earlier described neural network, necessary interfaces and time frame for calculations were designed and written down as a C++ code ready for linking with a code of the NN algorithm.

The fixed-point, discrete-time equivalent NN was designed in the form optimised for range, resolution, number of operations and parameter values. In the next step, the fixed-point form was coded in C++ using A|RT Library. The C++ controller code was converted into the s-function and simulated against the fixed-point network, to verify C++ coding correctness. The next operations: compiling VHDL source and fitting yielded a programming file for the APEX chip, which ended the implementation procedure.

4.6. Performance of smart sensor

In order to check the correctness of the implementation and to evaluate the performance of the digital signal processing part of the smart sensor, experiments have been conducted, using excitations described earlier. During these experiments, the sensor worked on-line and its output was recorded together with the real force applied to the frame. Results of four selected cases, different from those used for neural network training and validation, are shown in Fig. 11.

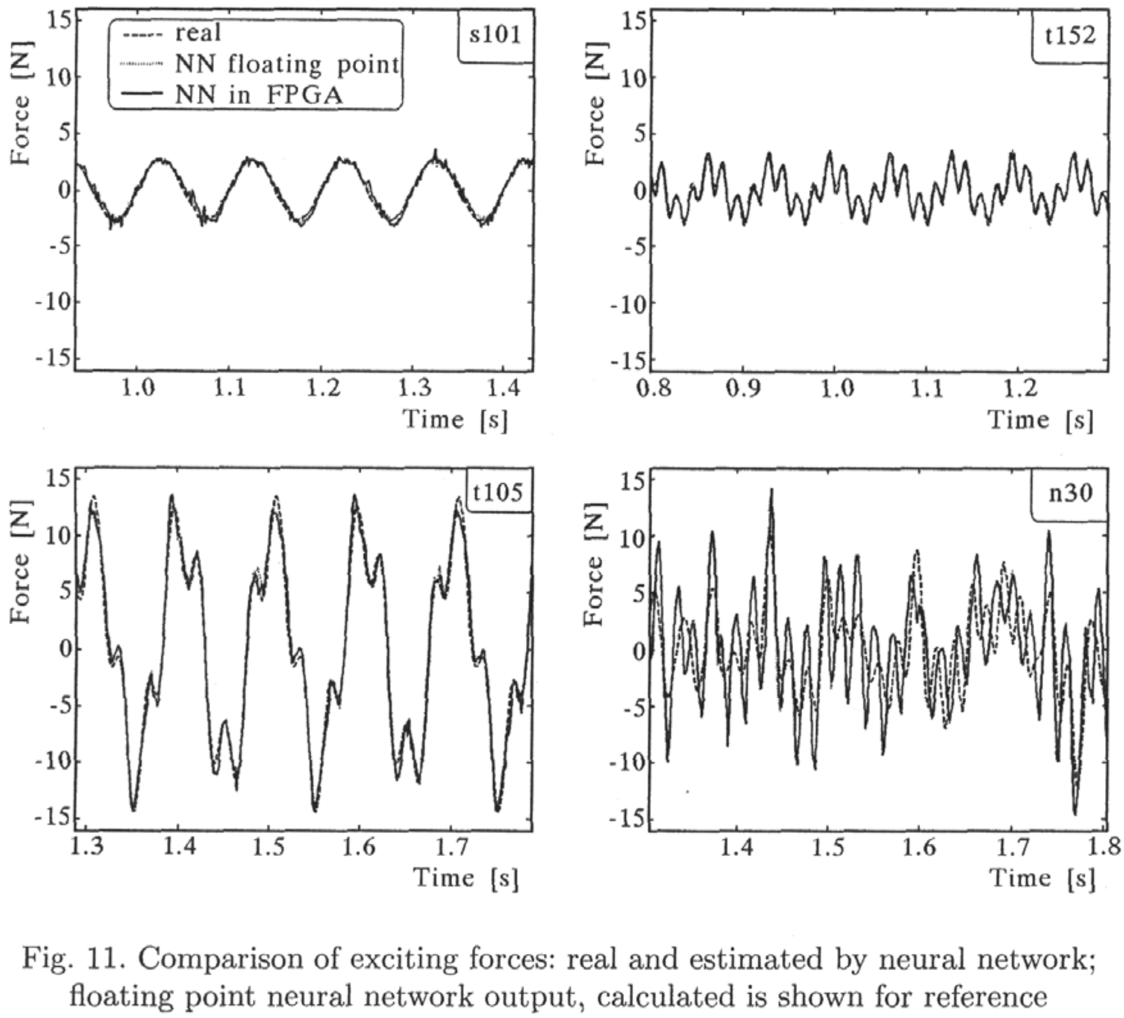

Fig. 11. Comparison of exciting forces: real and estimated by neural network; floating point neural network output, calculated is shown for reference

To obtain quantitative information about the signal processing performance, a mean squared error (MSE) and correlation coefficient (CC) have been adopted as a measure of conformity between the real and estimated courses of the force. Their values for ten test excitations are presented in Table 1. In the cases with "smooth" force changes: sine (symbols beginning with s) and two-tone (symbols beginning with t), MSE is below 0.4% and CC is above 0.96. The neural network performs worse for "wideband" excitations: square-wave (p171) and noise (n30), with MSE up to 3% and CC down to 0.6. The values, not shown and calculated for excitations used for early training stopping and especially for training, are much better, however with a similar relationship between the cases with a simple and complex frequency spectrum.

Table 1. Comparison between the real and estimated force

Test run symbol	MSE relative to output range	Correlation coefficient
s101	0.00203162	0.988793
s102	0.000201001	0.973509
t105	0.00103453	0.992258
t106	0.000202366	0.976937
t151	0.0034432	0.965016
t152	0.000216336	0.962248
t153	0.00303747	0.965809
t154	0.000321294	0.96195
p171	0.027893	0.73342
n30	0.017831	0.596783

The whole application, i.e. neural network calculations and peripheral interfaces, occupies ca. 50% of logic elements and ca. 15% of memory in EP20K100E FPGE chip, leaving enough resources for future realisation of a communication protocol and interface. Working at 6.25 MHz, the system carries out neural network calculations in 88.5 μs. After addition of 4.2 μs, for servicing external devices, it gives a maximum sampling rate of over 10 kHz. A faster clock (20 MHz is permissible at the moment) would allow one to achieve 32 kHz sampling frequency, and even more, with further increased clocking speed after time optimisation of the design, which should be feasible as no effort have been taken in this field yet.

A good agreement between the directly measured loading force and the one estimated by the smart sensor can be observed. Another advantage of the application of the smart sensor presented in the paper is the possibility to determine the loads in real time, which is very important for the application to monitoring and control systems.

5. Conclusions

The concept of a smart sensor presented in the paper gives new possibilities for application of operational force measurements for monitoring and diagnostics purposes. Variability of the operational force seems to be one of the most sensitive symptoms of damages in operating structures. But practical application of such a sensor requires an individual design of the sensor for each case. The design methodology formulated by the authors helps to design a smart sensor for identification of loading forces in complex mechanical structures.

The proposed methodology of the implementation of signal processing algorithms in FPGA/ASIC has been successfully applied to the problem of a neural network based intelligent sensor development. It allowed for automation of the most time consuming, error prone phases of the implementation, which normally require electronics specialists, yet leaving a full control over each operation.

The biggest problems have been met during the testing of different approaches toward organisation of NN calculation data flow, which required writing several versions of the code. There is a need for a tool facilitating the testing of several data flow architectures of the same algorithm.

The test results show applicability of the designed sensor for load measurements based on the measured response signals.

Acknowledgment

This research has been supported by the State Committee for Scientific Research (KBN) under grant No. 7 T07 C03519.

References

1. BUSBY H.R., TRUJILLO D.M., 1986, Solution of an invesrse dynamics problem using an eigenvalue reduction technique, *Computer and Structures*, **25**, 1

2. ENGEL Z., 2000, *Mutual Theorems in Mechanics*, Wyd. AGH, Kraków (in Polish)

3. FRANK R., 2000, *Understanding Smart Sensors*, Artech House, Norwood

4. GIERGIEL J., UHL T., 1989a, Identification of the impact force in mechanical systems, *Archives of Machine Design*, **XXXVI**, 2-3

5. GIERGIEL J., UHL T., 1989b, Identification of the input impact type forces in mechanical systems, *The Archives of Transport*, **1**, 1

6. HAAS D.J., MILANO J., FLITTER L., 1995, Prediction of helicopter component loads using neural networks, *Journal of the American Helicopter Society*, **1**
7. HACKL M., 1985, Anwendung des Satzes von der wechselseitigen Energie, *Acoustica*, **58**, 4
8. JANG J.-S.R., SUN C.-T., MIZUTANI E., 1997, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, Upper Saddle River
9. LI J., 1988, Application of mutual energy theorem for determining unknown force sources, *Proc. of Internoise 88*, Avignion
10. LISOWSKI W., MENDROK K., UHL T., 2001, Identidfication of loads basing on output signal measurement, *Mat. V Konferencji Naukowej nt. Metody doświadczalne w budowie i eksploatacji maszyn*, Wrocław-Szklarska Poreba (in Polish)
11. PERRY D.L., 1998, *VHDL*, McGraw-Hill
12. PETKO M., 1999, Mechatronics product realization (in Polish), In: *Mechatronics design*, edit. T. Uhl, Wyd. KRiDM AGH, Krakow
13. PETKO M., 2001, Implementation of control algorithms in ASIC/FPGA, *Mechatronics. 5th Franco-Japanese Congress and 3rd European-Asian Congress of Mechatronics. Proceedings*, 70-74, Besancon
14. PETKO M., UHL T., 2001, Embedded controller design-mechatronic approach, *Proc. of the Second International Workshop on Robot Motion and Control RoMoCo'01*, Politechnika Poznańska, Bukowy Dworek, 195-200
15. SIMONIAN S.S., 1981, Inverse problems in structural dynamics, *Int. Journal on Numerical Methods in Engineering*, **17**, 357-365
16. TRUJILLO D.M., 1987, Application of Dynamic programming to the general inverse problem, *International Journal on Numerical Methods in Engineerring*, **23**, 613-624
17. UHL T., 1998, *Computer assisted identification of mechanical structures*, WNT, Warszawa (in Polish)
18. UHL T., 1999, Trends and progress in monitoring and diagnostic systems, *PAK*, **4** (in Polish)
19. UHL T., 2001, Identification of loading forces in mechanical systems using genetic algorithms, *Proc. of AIMECH 01*, Gliwice
20. UHL T., MROZEK Z., PETKO M., 2000, Rapid control prototyping for flexible arm, *1-st IFAC Conference on Mechatronic Systems, Preprint*, **2**, 489-494, Darmstadt
21. ZION L., 1994, Predicting fatigue loads using regression diagnostics, *Proceedings of The American Helicopter Society Annual Forum*, Washington D.C.

22. APEX 20K Programmable Logic Device Family. Altera Corp., 1999

23. A|RT Builder User's and Reference Documentation, Frontier Design Inc., Danville, 2000

24. A|RT Library User's and Reference Manual, Frontier Design Inc., Danville, 2000

25. Fixed-Point Blockset User's Guide, The MathWorks Inc., Natic, 1999

26. FPGA Compiler II / FPGA Express VHDL Reference Manual, Synopsys, Inc., 1999

27. Neural Network Toolbox User's Guide, The MathWorks Inc., Natic, 1999

28. Nonlinear Control Design Blockset User's Guide, The MathWorks Inc., Natic, 1997

Inteligentny sensor do pomiaru obciążeń w czasie pracy

Streszczenie

Ważną techniką wykorzystywaną w monitorowaniu stanu technicznego maszyn i lokalizacji uszkodzeń jest pomiar obciążeń w czasie ich pracy. Jednak bezpośredni pomiar jest zwykle trudy albo nawet niemożliwy. Artykuł dotyczy realizacji idei opracowania "inteligentnego sensora" estymującego obciążenie na podstawie odpowiedzi struktury mechanicznej. W opisanym przykładzie, realizowana sprzętowo sieć neuronowa została zastosowana do otrzymania przebiegu obciążenia na podstawie drgań określonych punktów struktury. Artykuł omawia także problemy związane z prototypowaniem i implementacją algorytmów przetwarzania sygnałów, ze szczególnym uwzględnieniem sytuacji, gdy część sprzętowa oparta jest na układach ASIC/FPGA, dla której opracowano metodologię implementacji, zweryfikowaną poprzez praktyczne zastosowanie do problemu inteligentnego sensora. Pokazano szczegóły tej procedury wraz z zastosowanymi narzędziami i wyniki osiągnięte w trakcie jej realizacji. Przeanalizowano wyniki działania urządzenia podczas eksperymentu i przedstawiono wnioski.

Manuscript received March 18, 2002; accepted for print April 9, 2002