MECHATRONIC APPROACH TOWARDS FLIGHT FLUTTER TESTING

Grzegorz KARPIEL, Maciej PETKO, Tadeusz UHL

Katedra Robotyki i Mechatroniki
Akademia Górniczo-Hutnicza
al. Mickiewicza 30, 30-059 Kraków
gkarpiel@agh.edu.pl, petko@agh.edu.pl, tuhl@agh.edu.pl

Summary

The paper presents an idea of identification of the flutter phenomena during a flight. The proposed flutter detection algorithm is based on the identification of natural frequencies and modal damping ratio for an airplane structure based on in-flight vibration measurements. The procedure can be realized during a flight using measured actual vibration. The algorithm is based on recursive identification of model parameters and wavelets based signal filtering. The real-time realization is implemented in hardware and tested during a flight. FPGA technology is used for the hardware design. The results of a test of the hardware system prototype are presented.

Keywords: flutter, real-time modal analysis, FPGA.

MECHATRONICZNE PODEJŚCIE DO BADANIA MARGINESU FLATTERU SAMOLOTÓW

Streszczenie

W artykule przedstawiono implementację algorytmu identyfikacji flatteru podczas lotu. Przedstawiony algorytm bazuje na identyfikacji częstotliwości własnych oraz współczynnika tłumienia poprzez pomiar drgań struktury samolotu. Zaprezentowana procedura może być realizowana podczas lotu wykorzystując dostępne sygnały z czujników drgań. Algorytm wykorzystuje transformatę falkową jako filtr częstotliwościowo-czasowy dla izolacji pojedynczych postaci drgań o parametrach zmiennych w czasie. Realizację w czasie rzeczywistym wykonaną w postaci sprzętowej sprawdzono podczas testowego lotu. Do zaprojektowania struktury sprzętowej użyto technologii FPGA. Przedstawiono rezultat działania prototypowego urządzenia.

Słowa kluczowe: flatter, analiza modalna w czasie rzeczywistym, FPGA.

1. INTRODUCTION

Unstable vibrations of airplane can be a reason of a catastrophic failure of an aircraft. Such a case of vibration is commonly defined as a flutter [1]. In the literature [2] [3] many cases of the flutter phenomena are carefully studied. Each aircraft should be tested using numerical simulations and experimental investigation of airplane structure vibrations due to the controlled excitation during a flight. A vibration based experimental flutter test during a flight is very expensive, time consuming and is a critical part of the aircraft certification procedure [4]. The procedure of in-flight flutter testing consists of measurements of structural vibration of an airplane and, based on these measurements, estimation of modal model parameters [5] [6]. Each possible aircraft configuration should be tested separately. There are two cases: first, if an artificial vibration excitation is used during a flight and second, if an ambient excitation (non measured) is a cause of structural vibration. Many papers are focused on the development of operational modal analysis methods, which can be directly applied for modal parameters estimation during a flight [7] [8]. Some of them are realized iteratively in real-time, with less then 1 second interval between estimations [9] These methods, applied for flight flutter testing, shorten testing procedure and dramatically decreasing its costs. During the proposed procedure, an aircraft is excited by an operational excitation (turbulences, maneuvers of the airplane) or using dedicated excitation devices, and its vibration responses are measured. Based on the recorded signals, the modal parameters of the structure are estimated using some identification method. These parameters, in particular modal damping and natural frequencies, are useful for flutter boundaries determination. The described procedure should be applied at each test point to determine the Flight Clearance Envelope (FCE) [4]. There are many different modal parameters identification methods that could be used for in-flight flutter testing. They can be categorized to experimental and operational modal analysis methods [8]. The second ones can be applied for an experiment scenario, in which the excitation force is not measured and. implemented in the proposed flutterometer design. The flutterometer is an electronic device that permits estimations of modal parameters during a flight, based on the structure response, focused on tracking of the flutter margin changes with variations of flight conditions. Both time domain and frequency domain methods match requirements of the on-line modal parameters identification [9]. A wide range of methods can be considered for final implementation; many of them have been tested using simulated and real measurement data. The time-domain method, based on a simple recursive RLS algorithm is applied in the proposed solution as efficient enough and relatively easy to implement. The major assumption applied in the procedure is linearity of the isolated system modes that are, however, nonstationary due to changes of flight parameters and variations of aerodynamic feedback. Signal processing procedures should satisfy these assumptions. The wavelet based signal preprocessing is employed in the solution.

2. IMPLEMENTATION OF SIGNAL PROCESSING ALGORITHMS IN FPGA

2.1. Hardware – software partitioning

The flutter monitoring algorithm consists of five steps: data read from analog-to-digital converters (ADC), convolution of a stored wavelet with signals from sensors [10], signals reconstruction, on-line recursive least square (RLS) routine for estimation of parameters of ARMA models [9], and determination of damping coefficients for flutter detection (Fig. 1) [11]. The result of calculation is stored in memory and transmitted to PC by USB interface. The wavelets are generated during system initialization by the processor and stored in buffers in custom hardware. Floating-point operations of software part are accelerated by hardware custom instructions created in the Nios II ALU, namely floating-point addition, multiplication, reciprocal and square root.

The normally most time-consuming part of the algorithm – convolution and signal reconstruction is accelerated by a custom hardware accelerator and executed in fixed-point arithmetic [14]. It is possible because of restricted resolution of input data and fixed size of both the wavelet table and the input buffer. Signal reconstruction is carried out simultaneously with convolution. The hardware accelerator (Fig. 2) is controlled by a single master control unit responsible for signal acquisition from DACs, storing of the samples in a circular buffers, generation of addresses and control signals for memories containing wavelet and signal buffers, and synchronizing data feed with actual calculations performed by multiple identical convolution and signal reconstruction units.

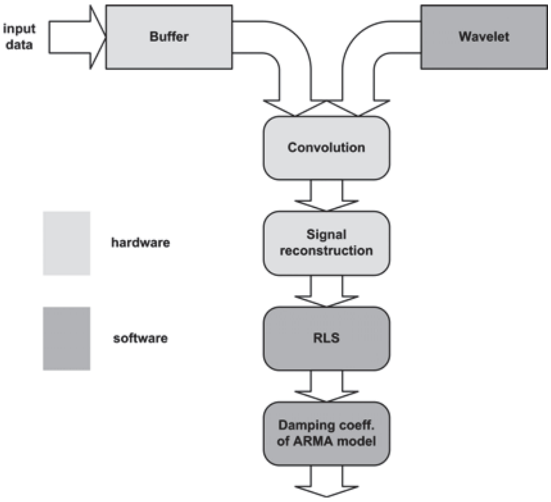

Fig. 1. Hardware-software partitioning of the flutter monitoring algorithm

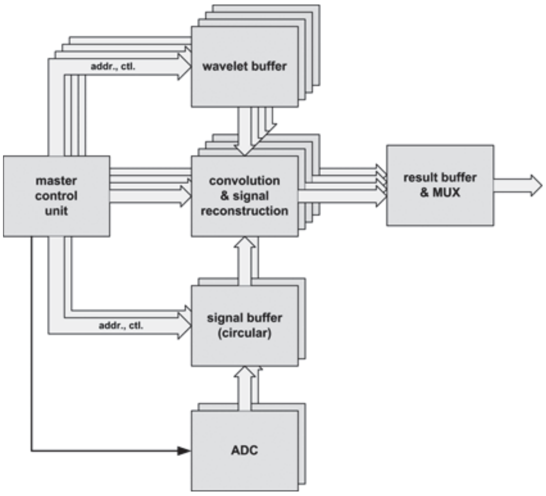

Fig. 2. Architecture of the custom hardware accelerator for multi-channel wavelet transformation

Such an arrangement allows for easy scaling i.e. simultaneously performing variable number of wavelet transformations of each of variable number of input signals without consuming more time, with variable wavelet length in each channel (1024 points maximum) and with signal time window width of 512 samples. In present version, the flutterometer is equipped with hardware accelerator capable of separation of three vibration modes in each of two input signals (six channels altogether).

2.2. Hardware architecture

The processor and the hardware accelerator work in master-slave arrangement with the processor as master. Completion of calculations by the hardware accelerator is registered in status bits and generates a DMA request to transfer the results to software. RLS algorithm and damping coefficients calculations are performed for each wavelet-filtered signal in floating-point arithmetic by the processor supported by customs instructions. Flutter appearance can be then determined using damping coefficients thresholds table indexed by the actual flight conditions. Fig. 3 shows overall hardware architecture of the flutterometer.

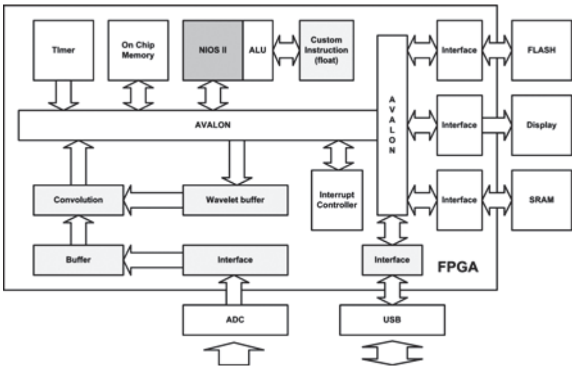

Fig. 3. Internal architecture of the flutterometer FPGA

All calculations are shared between software (dark gray blocks) and hardware (light gray blocks) [15], as shown on Fig. 3, in system created in a Stratix FPGA chip. Software means that parts of the algorithm are written in C and next compiled for Nios II soft-processor [13, 14]. Hardware fragments are realized in the logic of the FPGA and the Avalon Bus is used for data exchange.

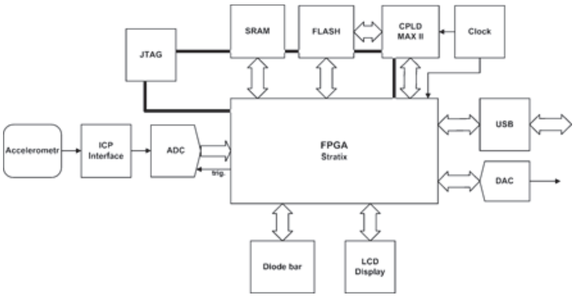

Fig. 4. Block diagram of the flutterometer hardware

The hardware of the flutterometer is based on a Stratix FPGA chip (Fig. 4) that is responsible for algorithm realization. FPGA programming is performed from flash memory by MAX II CPLD chip. Flash can store several configurations for FPGA and can be accessed directly by Nios II microprocessor implemented in FPGA. The latter allows for reading program code and long-term data storage by microprocessor. For temporary data storage, the system is equipped with fast static memory. FPGA, CPLD and both memories are placed in a single JTAG chain for programming and debugging.

The inputs to the flutterometer are signals from ICP piezoelectric accelerometers. After amplification and antialising filtering, they are digitized by analog-to-digital converter triggered by FPGA with programmable frequency in the range 10-200 Hz.

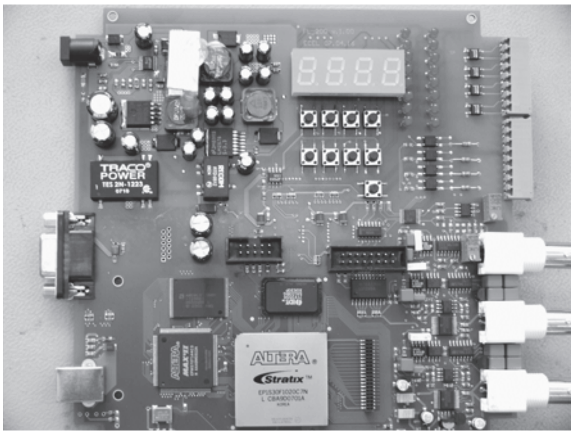

Fig. 5. Prototype of the flutterometer

The system has five outputs:

- analog output, generated by digital-to-analog converter, to enable external recording of changes of damping coefficient;
- serial digital interface for sending results of calculations to external devices;
- 8-segment, 4-digit display and LCD providing the current value of damping coefficient;
- multicolour diode bar indicating current flutter margin;
- digital, optoisolated alarm outputs signalling improper (too small) flutter margin.

The prototype of flutterometr is presented on Fig. 5. All components are placed on single PCB. LCD display is outside of the board and is designed for showing status information.

The whole application occupies 32% of logic elements, 29% DSP blocks and 34% of memory in EP1S30F1020C7 FPGA chip, leaving enough resources for future modifications of algorithm.

2.3. Experimental verification

In the first step of verification a triangle-wave input signal from a generator was used (Fig. 6). The frequency of the signal was 7.48Hz. Because the amplitude of the signal was constant, damping ratio was near zero. The frequency was identified correctly by the RLS algorithm to 7.48Hz.

On the next step, the shape of the input signal was replaced by square-wave. Additionally the amplitude of the signal was variable. For decreasing amplitude, the identified damping ratio was positive and vice-versa (Fig. 7).

Next, the device was tested on signals recorded during a flight of a military jet aircraft when flutter actually appeared [16].

Finally, the performance of the flutterometer has been verified during flight of a "Skytruck" aircraft.

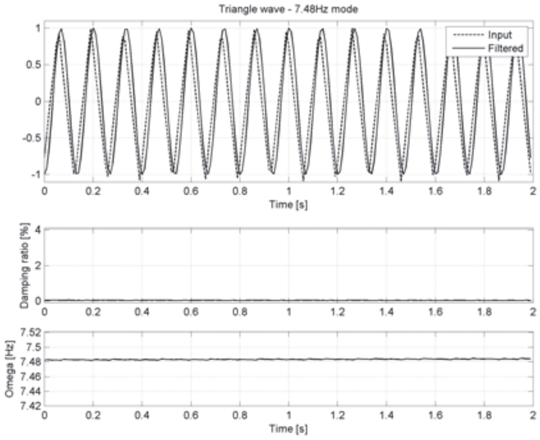

Fig. 6. Laboratory verification: triangle-wave input signal; frequency ca 7.48Hz, sample time 0.01s

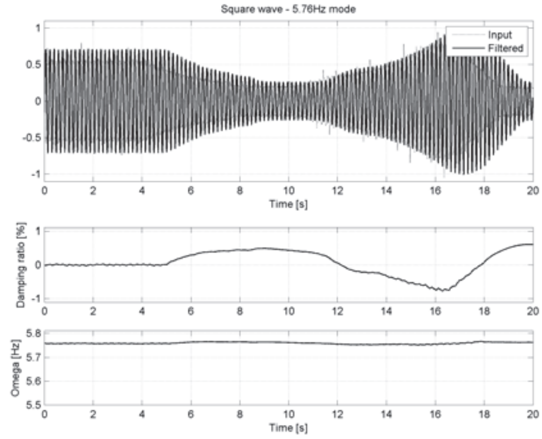

Fig. 7. Laboratory verification: square-wave input signal; frequency ca 5.76Hz, sample time 0.01s

The M28 "SkyTruck" (Fig. 8) is a twin-engined high-wing cantilever monoplane of all-metal structure, with twin vertical tails and a robust tricycle non-retractable landing gear, featuring a steerable nose wheel to provide for operation from short, unprepared runways where hot or high altitude conditions may exist. The M28 is dedicated for passenger or cargo transportation.

Fig. 8. M28 "SkyTruck" aircraft – civil version

During the flutterometer verification experiment, two signals have been used (Fig. 9-a). The first accelerometer was mounted on the right vertical tail. The second signal came from an accelerometer which was placed on the right plane. Both accelerometers measured vibrations in the vertical direction. Results of the calculations were transmitted to a PC (Fig. 9-b) by the USB interface, and stored on hard disk.

The signals were sampled with frequency of 100Hz during the test flight performed at constant speed of about 172KTS (320km/h). For each channel three modes were estimated. The frequency of the modes and appropriate wavelet parameters had been identified using data from previous flights of the same plane [17]. Results for modes of frequency: 5.2Hz, 7.4Hz and 12.5Hz in the vertical tail signal are presented on the Fig. 10.

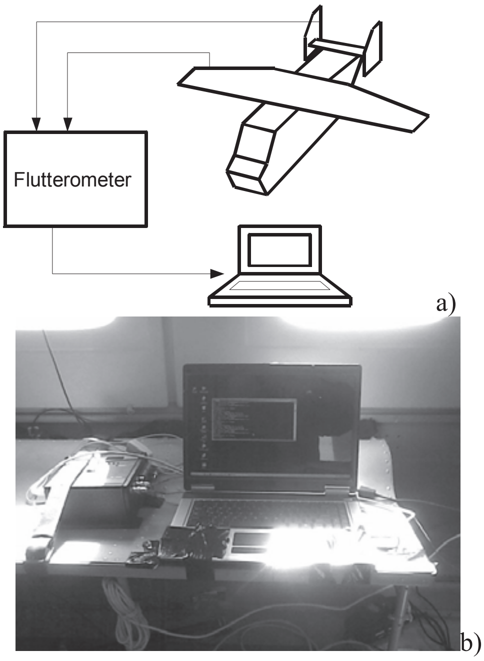

Fig. 9. In-flight experiment: a) diagram of connections, b) picture of equipment; the flutterometer on the left side of the table

3. CONCLUSIONS

The hardware platform for implementation is based on a modern FPGA chip. Implementation of the flutter monitoring algorithm is proposed with Hardware-Software Co-design approach, i.e. a part of it is realized by hardware and the remaining part by software running on Nios II soft-processor contained in the FPGA. The flutterometer is an example of System-on-Chip, which allows for high level of integration and flexibility – it can be altered, e.g. to optimize for different algorithm, or to add some functionality, by reprogramming the FPGA without modifications of the PCB. The methodology used, lowered costs and the duration time of the development process.

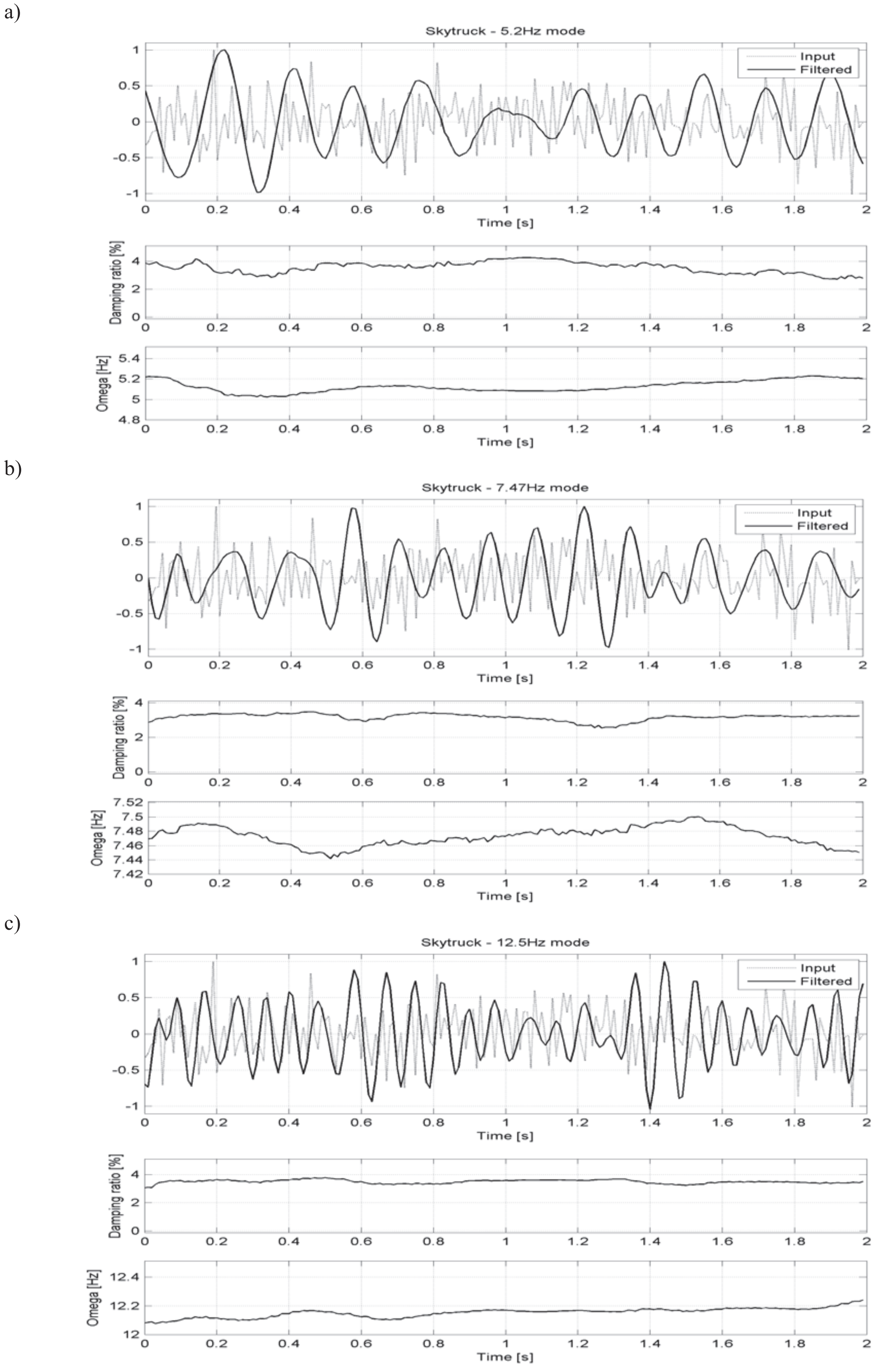

Fig. 10. “SkyTruck” in-flight experiment: V_{IAS}=172 KTS, H_{ST}=10000ft. Right vertical tail: a) 5.2Hz mode, b) 7.46Hz mode c) 12.5Hz mode

ACKNOWLEDGMENT

The project is financed by the Polish Ministry of Scientific Research in the frame of the EUREKA E!3341 FLITE 2 project.

REFERENCES

[1] Kehoe M. W.: *A historical Preview of flight flutter testing*, AGARD Conference proceedings 566, Advanced Aeroservoelastic Testing and Data Analysis, (1995).

[2] Garrick I. E. and Reed W. H.: *Historical development of aircraft flutter*, Journal of Aircraft, v.18, (1981), pp. 897-912.

[3] Brenner M. J., Lind R. C. and Voracek D. F.: *Overview of recent flight flutter testing research at NASA Dryden*, NASA TM/4792, (1997).

[4] Abbasi A. A. and Cooper J. E.: *Current Status and Challenges for Flight Flutter Testing*, Proceedings of ISMA 2006 Conference, Leuven, (2006), pp. 1523- 1546.

[5] Nissim E. and Gilyard G.B.: *Methods for experimental determination of flutter speed by parameter identification*, NASA TP-2923, (1989).

[6] Dimitriadis G. and Cooper J.E: *Flutter prediction flight flutter test data*, AIAA Journal of Aircraft, vol. 38, (2001), pp. 355-367.

[7] Hermans L. and Van H. der Auweraer: *Modal testing and analysis of structures under operational conditions: Industrial applications*, Mechanical Systems and Signal Processing, vol. 13(2), (1999), pp. 193-216.

[8] Uhl T., Lisowski W. and Kurowski P.: *In-operation modal analysis and its application*, AGH, Krakow, (2001).

[9] Bogacz M. and Uhl T.: *Real time modal model identification and its application for damage detection*, Proceedings of ISMA 2004 Conference, Leuven, (2004), pp. 1065-1075.

[10] Klepka A. and Uhl T.: *Application of wavelet transform to identification of modal parameters of nonstationary systems*, Journal of Theoretical and Applied Mechanics, Vol. 43, (2005) ,pp. 277-296.

[11] Klepka A.: *Identification of modal parameters of mechanical structures in nonstationary conditions*, PhD thesis, AGH Univ. of Science and Technology, Krakow, (2005) (in Polish).

[12] Wegrzyn M., Adamski M. A. and Monteiro J. L.: *The application of reconfigurable logic to controller design*, Control Engineering Practice, Vol. 6, 1998, pp. 879-887.

[13] Kambe T., Yamada A. and Yamaguchi M.: *Trend of system level design and approach to C-based design*, Microelectronics Journal, Vol. 33, (2002), pp. 875-880.

[14] Petko M. and Karpiel G.: *Semi-Automatic Implementation of Control Algorithms in ASIC/FPGA*, in 2003 IEEE Conference on Emerging Technologies and Factory Automation: Proceedings, vol. 1,(2003), pp. 427-433.

[15] Edwards M. D., Forrest J. and Whelan A. E.: *Acceleration of software algorithms using hardware/software co-design techniques*, J. of Systems and Architecture, Vol. 42, (1996/97), pp. 697-707.

[16] Uhl T., Petko M.: *Real-time flutter detection from in-flight vibration data*, Key Engineering Materials Vol. 347 (2007) pp. 679-684.

[17] Mendrok K., Lisowski W., Uhl T.: Sas, *Modal analysis of the M28 SkyTruck airplane basing on output only data measured during flight*, Proc. of 10th Int. Congress on Sound and Vibration, Stockholm, (CD) 2003.

dr inż. **Grzegorz KARPIEL** jest adiunktem w Katedrze Robotyki i Mechatroniki, Akademii Górniczo-Hutniczej. Główne jego zainteresowania skupiają się na mechatronice, zagadnieniach dotyczących robotów równoległych oraz projektowaniu sterowników opartych na układach FPGA.

dr inż. **Maciej PETKO -** adiunkt w Katedrze Robotyki i Mechatroniki AGH. Jego zainteresowania skupiają się na mechatronice, robotyce, zagadnieniach prototypowania i implementacji algorytmów przetwarzania sygnałów, głównie w sterowaniu i diagnostyce technicznej.

Prof. dr hab. inż. **Tadeusz UHL,** jest kierownikiem Katedry Robotyki i Mechatroniki, Akademii Górniczo-Hutniczej w Krakowie. W swoich pracach zajmuje się zagadnieniami dynamiki konstrukcji, a zwłaszcza analizą modalną. Jego zainteresowania obejmują także układy aktywnej redukcji drgań, układy sterowania i szeroko pojętą mechatronikę.