Adam PIŁAT[1]

AGH University of Science and Technology, Department of Automatics (1)

The Programmable Analog Controller. Static and Dynamic Configuration, as exemplified for Active Magnetic Levitation

Abstract. *This work presents the static and dynamic configuration of the controller, which is embedded into the hardware layer of an industrial programmable analog controller. Signal processing is carried out in the hardware layer by an analog signal processor, which constitutes the basic component of the newly-developed controller. The static and dynamic configuration for an active magnetic levitations system is used as an illustration.*

Streszczenie. *Praca przedstawia statyczną i dynamiczną konfigurację parametrów układu regulacji zrealizowanego w warstwie sprzętowej programowanego analogowego sterownika przemysłowego. Bezpośrednie wykonywanie formuły regulatora jest realizowane przez procesor analogowy stanowiący element sterownika. Uruchomienie i rekonfiguracja regulatora zostały przedstawione dla systemu aktywnego magnetycznego zawieszenia. (**Statyczna i dynamiczna konfiguracja programowanego sterownika analogowego na przykładzie aktywnej lewitacji magnetycznej**).*

Keywords: dpASP, FPAA, real-time control, Active Magnetic Levitation.
Słowa kluczowe: dpASP, FPAA, sterowanie w czasie rzeczywistym, aktywna lewitacja magnetyczna.

Introduction

This work is an extended version of material presented at the SENE 2011 conference [1]. The research into new solutions in the field of real-time control systems was motivated by the control study on active levitation applications, namely suspensions and bearings. The first successful research into levitation control supported by Field Programmable Analog Arrays (FPAA) [2] provided the drive towards programmable analog solutions as an alternative to the familiar digital control systems [2]. Ongoing study has shown this technology to have some potential. Nowadays, control systems are based on microcontrollers, digital signal processors, Field Programmable Gate Arrays (FPGAs) and multi-purpose computers. In every case, signal conditioning circuits, analog to digital converters, digital to analog converters and/or pulse width modulators are used to close the control loop. Parallel signal processing is guaranteed solely by synchronous multiprocessor platforms or FPGA devices. Apart from this, the controller program is executed sequentially. The complexity of signal processing requires the application of real-time system and the appropriate software. All of these factors mean that the control execution time increases on account of the number of calculations and peripherals handled. In recent years, a slow growth in the field of programmable analog devices has been perceptible. Some are available on the market, but their capacity is still insufficient for carrying out complex signal processing tasks. Analysis of the literature has shown that there are numerous practical applications which are conducted with the support of configurable analog devices [3-8]. In many cases, the solutions presented were created on the basis of manual configuration and parameter tuning. New devices have become available on the market in the past few years. Research into hard real-time applications and investigations into the possibilities of reconfiguration were reported in [9]. A set of design support tools for MATLAB has been developed and reported. The discussion on the features and limitations of FPAA-based devices for active levitation systems is given in [10].

The aim of this work is to present the results of the static and dynamic re-configuration of the Programmable Analog Controller. It is important to find an answer to the question of the time slots that need to be reserved for configuration transmission. These experimental results are significant from the control point of view. The complex signal processing architecture to be embedded into the controller can be imagined. Another crucial aspect is knowing how much time is required solely for reconfiguration of the controller downloading new parameters.

Programmable Analog Controller

Under Research Project No. 3585/B/T02/2009/37, financed by the Polish Ministry of Science and Higher Education, the Programmable Analog Controller (PAC) was developed and manufactured [11-13]. The first version of the PAC, with a semi-automatic configuration of its resources, and successful configuration for hard real-time control of active magnetic levitation systems, namely suspension and bearing, has demonstrated some of this technology’s potential. This permits a conclusion to the effect that programmable analog processors can be applied to simple control tasks, but that, with the controller architecture created, complex signal processing can be carried out.

The controller developed is characterized by modular architecture (Fig. 1, 2) and can be used as a universal apparatus for research purposes [11-13]. Its reconfigurable hardware architecture allows its use in a range of applications. Moreover, the software tools developed enable the programmable configuration of its resources. The controller consist of the following modules: I/O, A/D, D/A, Main APU, Power Driver (PD), Power Supply (PS), and Power Supply for Power Driver (PS-PD) [12]. It was created and built to meet industrial standards (Fig. 1).

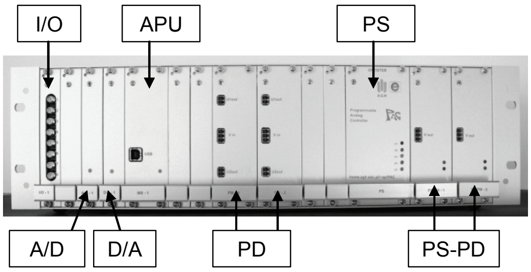

Fig. 1. PAC configured with I/O, A/D, D/A, APU, PD, and Power Supply Modules PS and PS-PD

The PAC core is based on digitally programmable Analog Signal Processors (dpASP). The Analog Processing Unit (APU) Main Module consists of the analog processors, the analog bus by which they are routed, and a digital control unit with communication interfaces [13] (Fig. 3). It is important to point out that this apparatus can also function without analog processors. It can be configured as an oscilloscope or digital controller, in which case, the A/D and D/A modules can be used, and the control loop can be closed by the embedded microcontroller or by the host computer when interfaced using the I/O Module.

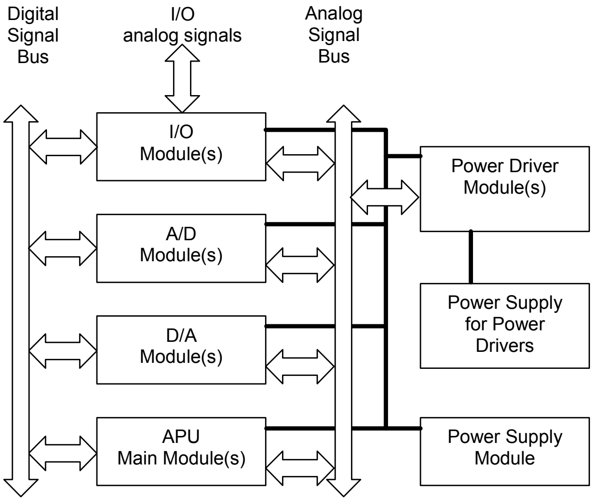

Fig. 2. PAC architecture

The controller features a multi-level architecture that allows the static and dynamic configuration of the embedded analog processors to be executed. The microcontroller applied in the hardware layer is responsible for configuration of the analog processor, digital module service, and host controller communication. The PAC-host computer communication is carried out via the USB interface. In the hardware layer, the analog processors are configured with the support of the SPI interface. The PAC and host computer software applications enable the performance of a number of configuration tasks and the handling of the APU Main Module and the other peripheral digital PAC Modules, such as, for example, data acquisition by the A/D Module.

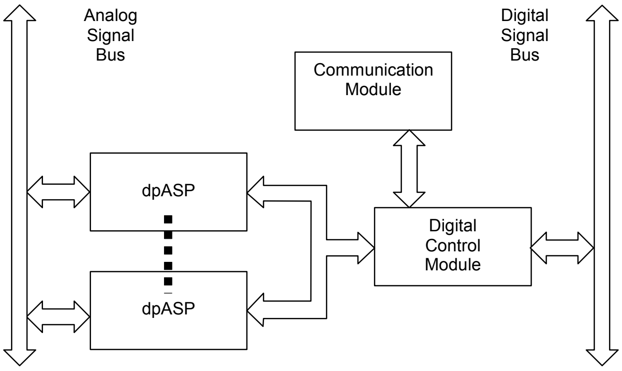

Fig. 3. PAC Main Module architecture

Programming the controller

The signal routing and signal processing method embedded into programmable analog processors can be configured manually. The dpASP devices are configured with the support of a dedicated EDA tool. The design of the dpASP configuration requires a knowledge base as regards the PAC controller, its architecture, dpASP resources and the available Component Analog Modules (CAMs) used to configure the signal processing path. To support the analog processor configuration, a custom method was designed, It allows the dpASP configuration to be generated semi-automatically [14]. With this method, the CAMs are placed and touted together, the I/O ports are attached and CAM properties such as driving frequency and gain are defined. On the basis of the CAM modules available in the EDA library and limited analog processor resources, it is possible to set up a wide range of sampled analog signal processing. The data bit-stream generated as a result of the configuration stage is used to program one or more analog processors, if chained [16]. Such a configuration is downloaded to the controller with a core processor ID in order to facilitate the unique identification of the target devices. The method allows the configuration to be changed either completely or partially while the controller is operating. It also permits the modification of selected properties in the signal processing paths. The method of semi-automatic configuration developed as part of the research allows the project to be verified and validated before downloading to the controller.

Active Magnetic Levitation test-rig

The MLS2EM system [15] (Fig. 4) was chosen to carry out the research experiments because it requires a real-time feedback in order to stabilize the levitated object. The MLS2EM is connected to the PAC via the I/O Module using port No. 1, configured as input, and port No. 8, configured as output. The displacement sensor and control signal were connected to the listed ports, respectively. In this architecture, the internal MLS2EM power actuator was used to supply the electromagnet. The PAC was thus operating purely as a controller. The displacement signal is measured with respect to the electromagnet surface and, available in a voltage form, is processed by the control structure embedded in the analog processor. The control signal generated steers the coil current power actuator to drive the electromagnet coil.

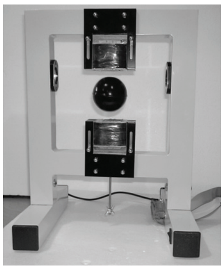

Fig. 4. Laboratory test-rig: Active Magnetic Suspension System MLS2EM [14]

The *PACCtrlGenPD* method developed during the research creates an analog processor configuration, sets the driving clock frequency, adjusts the parameters for the CAMs being used and verifies the control loop parameters at the design stage (Fig. 5). The controller in the PD structure [16] is configured using half-sample CAMs and ZOH CAMs in order to synchronize signal flow (see Fig. 5). This circuit carries out the discrete time control formula (1). The ZOH block is used to hold the input signal value through one sample period of the driving clock.

$$\begin{aligned} e(k) &= w(k)\text{-}x_1(k) \\ u(k) &= K_0\cdot u_0+K_P\cdot e(k)+K_D\cdot(e(k)\text{-}e(k\text{-}1)) \end{aligned} \tag{1}$$

On the basis of the CAM properties feedback, the controller parameters are verified at the design stage. The differences between the parameters requested and those implemented allows the properties of the signal processing that has been carried out to be checked and the closed system dynamics to be validated. In order to observe control and object displacement, the PAC A/D Module was used and configured to perform measurements with a sampling frequency of 2kHz. Both signals were acquired in parallel and synchronously, with the same sampling rate. The data collected is presented in the time diagrams for both investigations (see Figs. 8 and 11).

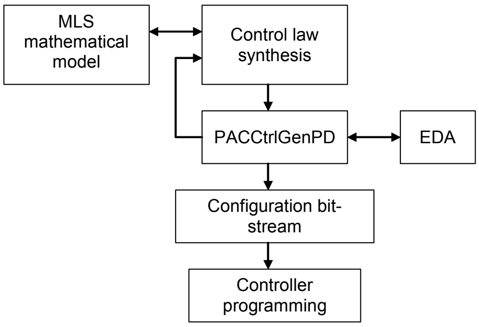

Fig. 5. Semi-automatic Controller synthesis for AML

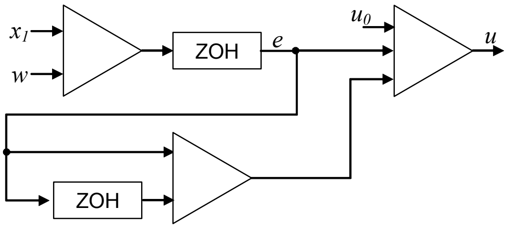

Fig. 6. PD controller structure embedded in the analog processor

Hardware configuration steps

Because the PAC is a solution for embedded control, the configuration and data exchange are carried out with the support of the communication interface. In the version of the PAC under consideration, the USB interface is used. An APU Main Module microprocessor is installed and is responsible for the direct configuration of the analog processors. The following tasks are executed sequentially in order to configure the PAC resources either statically or dynamically:

- A – configuration file read,
- B – data transfer to microcontroller,
- C – processing data stream,
- D – configuration of analog processor,
- E – configuration acknowledgement,
- F – PAC acknowledgement,
- G – End of configuration procedure.

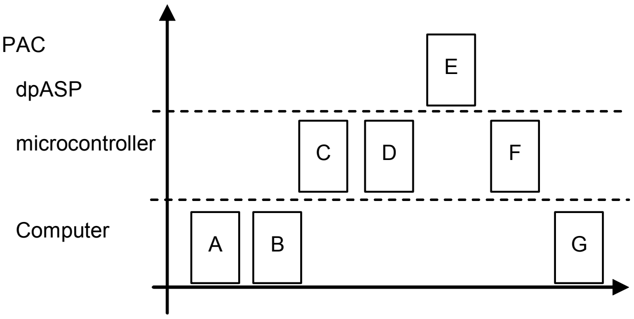

Fig. 7. Configuration tasks time-line

Method for investigating real-time control

The PAC operates as an embedded controller and the investigation of the real-time operation therefore requires the use of the A/D module. The A/D module is configured with a sampling frequency and the buffer length for the collection of the required set of samples. The PAC's functionality allows the acquisition task to be started on request. Therefore, in order to acquire the displacement of the levitated object and the control signal generated at the configuration stage, the A/D module is triggered and the configuration procedure is started. Finally, the collected data collected is transmitted to the host computer for further analysis. The static and dynamic configuration reports consist of diagrams of the signal time measured.

Method for the investigation of the configuration and reconfiguration procedure

For the purposes of this research, the investigation was carried out by the host computer because the configuration and reconfiguration algorithm are executed directly from the MATLAB environment. The functionality of the USB communication interface was also studied. To investigate the configuration and reconfiguration cycle time, 100 iterations were carried out. The configuration procedure consist of four steps, corresponding to the configuration of the selected analog processor at the Main APU Module. These four steps are marked as scenarios A, B, C, and D. They correspond to the number of analog processors configured: 1, 1-2, 1-3, and 1-4 respectively. The time is acquired at the beginning of the task A and at the end of task G (see Fig. 7). In order to achieve comparable results, the same configuration data was sent to all the processors. The time being measured includes the USB transmission management carried out by the Win7 system. The serial communication, with a transmission rate of 115200bps, was set. The configuration data consisted of 232 bytes and 24 bytes for the static and dynamic configuration, respectively. The time measured was then analyzed to calculate the average execution time t_m and standard deviation σ for the configuration scenarios under consideration. The results have been summarized in table form and the appropriate histograms plotted.

Static configuration of the PAC

The static configuration must be carried out when the controller is started and when the structure of the signal being processed is changed. In the application under consideration, the static configuration is downloaded to the PAC from the host computer. The analog processor is configured and reset automatically at the end of the configuration routine. To illustrate the static configuration, the PD controller was developed as described in the section devoted to the magnetic levitation system. First, the ferromagnetic sphere was located at the bottom of the admissible position region, before the configuration procedure was started. The configuration procedure was then executed. The processor start-up was observed after the configuration stage. It is clearly visible in Fig. 6, at the 0.12s time point, when the control signal is set to zero. The control action is calculated and the control signal increases to pull up the sphere. Finally, the object levitates at the desired position of 10mm and the control tends to a steady value of 0.4V. The static configuration was investigated using the proposed method. The results are summarized in Table 1 and the histogram plots are given in Fig. 9

Table 1. Statistics for the static configuration

Proc. No.	Scenario A Time [ms]	Scenario B Time [ms]	Scenario C Time [ms]	Scenario D Time [ms]
1	t_m = 51.01 σ = 3.62	t_m = 101.53 σ = 6.06	t_m = 152.73 σ = 7.32	t_m = 203.80 σ = 7.64
2	X			
3	X	X		
4	X	X	X	

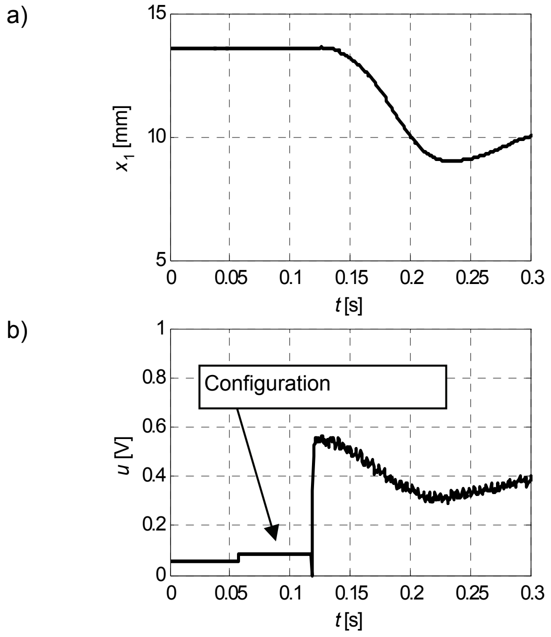

Fig. 8. Static configuration of the control law for an active magnetic suspension system: a) object position, b) control signal.

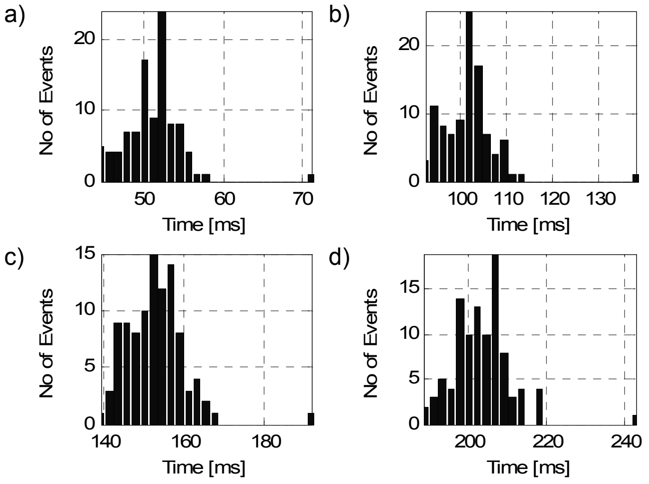

Fig. 9. Histograms of the static configuration download executed for a) single, b) two, c) three, d) four processors.

The histogram of the time events for a single processor configuration, extracted from all the configuration scenarios, is presented in Fig. 10. A distribution similar to that for Scenario A can be seen, as can the fact that the static configuration of a single processor takes about 50ms, including the USB communication.

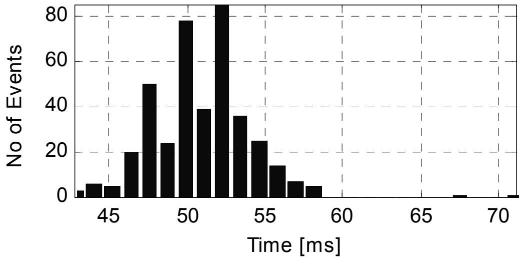

Fig. 10. Histogram of 400 static configuration programming events

In the case of industrial applications, it is assumed that the static configuration will be available on the APU Main Module in the microcontroller memory. Such a configuration can be loaded at controller startup or on demand. In other cases, when the configuration has to be downloaded from the host, the transmission properties must be considered.

Dynamic reconfiguration of the PAC

One of the most important features of the controller developed under this research project is the potential for dynamic reconfiguration. The hardware and software solutions implemented use a unique programmability feature of programmable analog processors. The configuration parameters can be updated in on-line mode during real-time signal processing. This dynamic reconfiguration is a key point in control field applications because it allows the controller parameter to be moderated while operating. This special feature means that the PAC can be used as an adaptive controller. The dynamic configuration procedure is totally different from the static configuration because only part of the configuration is modified. Moreover, additional software is required for the configuration upgrade. On the basis of the static configuration and the program code generated for the analog processor, the appropriate calculations are performed and a new configuration is obtained. This procedure was carried out in the form of the Windows Visual application [2] and, for this research, in the form of a C-mex file to be executed in MATLAB. As a result of the execution of the reconfiguration functions, only the processor configuration upgrade is generated. In the case of the PD controller under consideration, such a configuration stream occupies a mere 24 bytes.

To illustrate the dynamic reconfiguration stage, the sphere was suspended in a stable position of x_{10} = 7.5mm, and was then moved to a position of x_{20} = 10mm by means of the modification of the steady control value (see Fig. 11). The configuration download and investigation procedure were identical, as was the case for the static configuration. In this instance, the results are totally different. First, the reconfiguration time is much shorter, which is directly connected with the configuration stream length. Second, the change of the analog processor parameters takes one sample clock period, which is a mere 1.34μs, because the analog processor is sampled at a frequency of 720kHz. Finally, the analog processor processes the signal permanently; there is no processor reset. The sole sample period reconfiguration time is relatively short in comparison to the system dynamics under consideration. Moreover, because, there is no processor reset, the control output is permanently valid (compare Fig. 8 and Fig. 11). The continuity in the control signal is a guarantee of the stable operation of the AML system.

The PD controller is calculated to stabilize the levitated sphere at the desired positions. When the steady control value is changed, the controller reacts with the lowest control output signal to implement a free fall motion, and is then applied to keep the ferromagnetic object levitated (see Fig. 11). The reconfiguration procedure was investigated using the method proposed here. The results of the experiments are summarized in Table 2.

Table 2. Statistics of the dynamic configuration

Proc. No.	Scenario A Time [ms]	Scenario B Time [ms]	Scenario C Time [ms]	Scenario D Time [ms]
1	t_m = 36.38 σ = 10.31	t_m = 72.78 σ = 15.96	t_m = 110.31 σ = 20.73	t_m = 148.57 σ = 24.61
2	X			
3	X	X		
4	X	X	X	

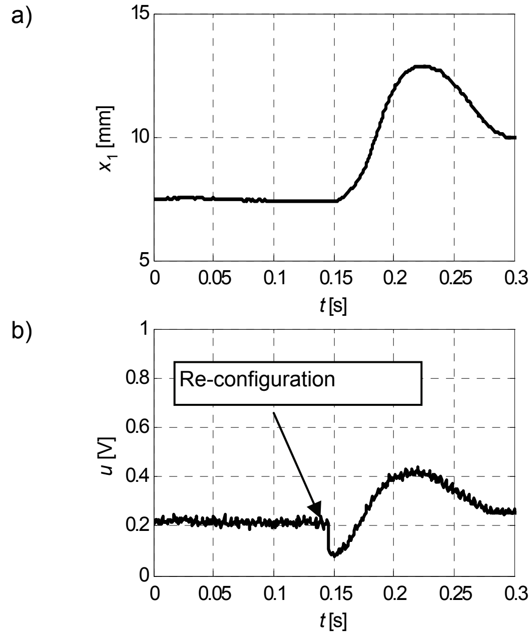

Fig. 11. Dynamic reconfiguration in the real-time regime: a) object position, b) control signal.

The dynamic configuration cycle is shorter than in the case of static configuration. This is owing to the relatively short data frame. Unexpectedly, a high value is registered fir the configuration time. This is owing to the USB protocol specification. The re-configuration histograms for the scenarios under consideration are given in Fig. 12. Fig. 13 shows the histogram of all single processor configuration; the time was calculated on the basis of the time registered for the inter configuration events.

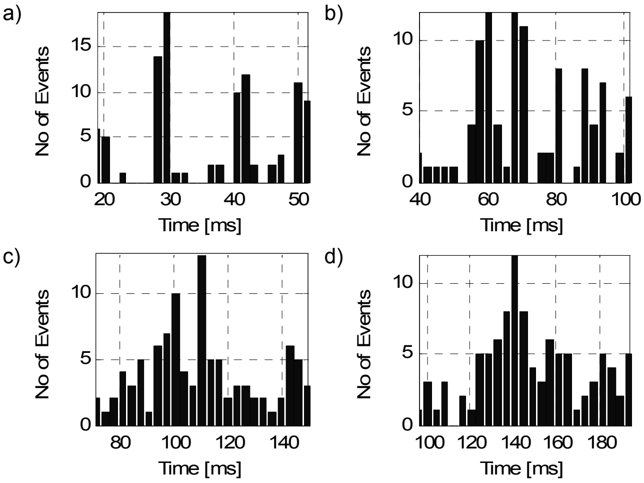

Fig. 12. Histograms of the dynamic reconfiguration procedure executed in Win7 using a USB interface

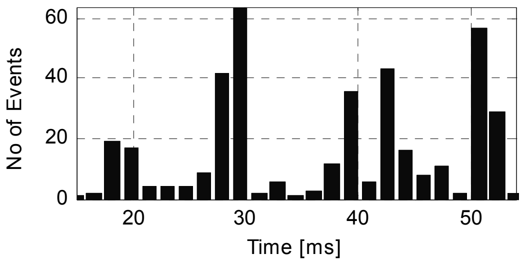

Fig. 13. Histograms of a single processor dynamic reconfiguration procedure executed in Win7 using a USB interface

Discussion about configuration methods

Observing the static and dynamic configurations reveals features and limitations to both of them. The static configuration must be executed at least once on start-up of the controller. The application of this configuration results in the restart of the processor at the end of the configuration cycle. This fact affects the devices being controlled when they are connected and supplied. In this case, which involved the multiple application of the method, proper management of the routing of input and output signals, is required, as is that of the power supply of the devices being steered in order to protect them and guarantee the safe operation of the system being controlled. The availability of controller reconfiguration allows the signal processing formula to be changed while the controller is operating. This allows structurally reconfigurable control systems to be developed. The dynamic configuration is the fastest way to tune up the controller properties and obtain the adaptive control system. The short configuration stream and shortest configuration download allows a real-time adaptive embedded controller to be attained.

Having both configuration features, the multilayer control system architecture has been obtained. The complex method and software tools created for reconfiguration purposes allow the management of controller resources. At present, all these tasks are carried out with the support of MATLAB for the fastest performance of supervisory and optimization control tasks.

It is important to point out that the controller processes signals in the hard real-time regime, and the data exchange conducted in parallel, without interrupting the control task. On the basis of the research carried out, the controller firmware and configuration download software should be changed to compact the data stream in order to speed up the configuration process and minimize USB imperfections. The extension of the communication protocols toward industrial standards should be also considered.

Finally, the static and dynamic reconfiguration task can be moved to the hardware layer for execution by the microcontroller embedded into the APU Main Module. Such a solution is less fuzzy for research purposes but, for target applications, it will eliminate communication problems and enable the controller to work in an autonomic way. In this case, some limits caused by the microcontroller resources will restrict the complexity of the control and optimization methods.

Conclusions

The Programmable Analog Controller is an interesting solution for research and industrial applications. Its main feature is the simplification and minimization of the components necessary to the formation of the signal processing path. Moreover, the version created under this research project allows their resources to be configured. In this case, the complex analog signal processing is available and configurable. Moreover, the signal processing parameters can be modified even in the real-time regime. On the other hand, there are some difficulties and limitations arising from, or caused by, the CAM constraints and the specifics of their operation. As a result, the solution under consideration here is neither as universal nor as easy to manage and configure as digital signal processing solutions. Apart from these inconveniences, the PAC stands as an alternative for practical, hard real-time applications, when rapid signal processing is required. The PAC can be handled by the high level control system designed under INSTEPRO research project [18]. The possibility of static and dynamic configuration opens up the feature of variable structure and adaptive control systems. The execution of a reconfiguration procedure within a range of milliseconds, even from a host computer with a USB interface and Win7 system, demonstrates that adaptive signal processing is attainable. To minimize the configuration and reconfiguration time, if required, embedding the calculation procedure into the microcontroller or modifying the communication interface is recommended and will be considered in further research devoted to upgrading the controller properties.

Given the familiar wish of the FPGA world [19], to wit: "*Furthermore, over the years a number of companies have promoted different flavors of field-programmable analog arrays (FPAAs). Thus, there is more than a chance that predominantly digital FPGAs will start to include areas of truly programmable analog functionality similar to that provided in pure FPAA devices.*" has not, so far, been fulfilled, the Programmable Analog Controller, with its static and dynamic configuration of programmable analog signal processing, satisfies the requirements for reconfigurable real-time controllers.

Publication has been supported by the European Regional Development Fund Grant no. UDA-POIG.01.03.01-12-171/08/00 within the framework of Project INSTEPRO.

REFERENCES

[1] Piłat A., Statyczna i dynamiczna konfiguracja programowanego sterownika analogowego na przykładzie aktywnej lewitacji magnetycznej, SENE 2011: X krajowa konferencja naukowa: sterowanie w energoelektronice i napędzie elektrycznym. Politechnika Łódzka, Instytut Automatyki. ISBN 978-83-7283-439-3, pp. 1-4

[2] Piłat A., Programmable analog hardware for control systems exampled by magnetic suspension, Computer Methods and Systems, Cracow, Poland 14-16 November 2005, pp. 143–148

[3] Berenson, D., Estevez, N., Lipson, H., 2005. Hardware evolution of analog circuits for in-situ robotic fault-recovery. In: EH '05 Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware, pp. 12–19.

[4] Malcher, A., Pietraszek, S., Przybyła, T., Kidoń, Z., 2009. Real time qrs detector based on field programmable analog array. Journal of Medical Informatics and Technologies 13, 183–188.

[5] Morales, D., Garcia, A., Castillo, E., Carvajal, M., Banqueri, J., Palma, A., 2011. Flexible ecg acquisition system based on analog and digital reconfigurable devices. Sensors and Actuators 165, 261–270.

[6] Stala, R., Grubiel M., Practical realization of state detection of contactless electrical energy transmission system using field programmable analog array fpaa. Electrical Review 2005, (81)78, 91-97.

[7] Visan D.A, Lita I., Jurian M., Cioc I.B., Simulation and Implementation of Adaptive and Matched Filters Using FPAA Technology, IEEE 16th International Symposium for Design and Technology in Electronic Packaging (SIITME), 23-26 Sep 2010, Pitesti, Romania, pp. 177-180

[8] Zhang W., Li Y., A Self-Adapted PID System Based on Intrinsic Evolvable Hardware, , IMECS 2008, 19-21 March, 2008, Hong Kong

[9] Piłat A., Hard and soft real-time in application of programmable analog device (in Polish). Trybus L., Samolej S. (ed.) Methods of manufacture and application of real-time systems, WKŁ, 2010, pp. 343-352.

[10] Piłat A., Features and limitation of the programmable analogue signal processing for levitated devices. 12th International Symposium on Active Magnetic Bearings, Wuhan, China, August 22-25. 2010, pp. 482–489.

[11] P-394 445 UP RP "Konfigurowany komputer analogowy", AGH, zgłoszenie z dn. 05.04.2011r., Piłat A., Klocek J.

[12] Piłat A., Klocek J., Modułowa aparatura do zadań diagnostyki sterowania, Par 2/2011, pp. 707-716

[13] Piłat A., Klocek J., Programmable Analog Computer, (paper under review).

[14] Piłat A. Semi-automatic design and code generation for FPAA devices. Computer Methods and Systems, 26–27 November 2009, Kraków, Poland, pp. 375–378

[15] MLS2EM, Dual electromagnets Active Magnetic Levitation System with current driver, INTECO Ltd. Krakow, Poland

[16] Piłat A., Klocek J., Investigation of chained analog signal processors in Programmable Analog Computer, 11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems, 2012, May 23-25, Brno, Czech Republic, (accepted for presentation)

[17] Piłat A., Stiffness and damping analysis for pole placement method applied to active magnetic suspension (in Polish), Automatyka, 2009, vol. 13, no. 1, pp. 43-54

[18] INSTEPRO, Zintegrowane sterowanie produkcją, www.instepro.agh.edu.pl

[19] Clive C. FPGAs: instant access, 2008, Elsevier Ltd., ISBN 978-0-7506-8974-8,

***Author**: Adam Pilat Ph.D. AGH University of Science and Technology, Department of Automatics, Al. A. Mickiewicza 30, 30-059 Krakow, Poland, E-mail: Adam.Pilat@agh.edu.pl.*