Embedded Architecture for Simultaneous Analog Signal Processing

Bertalan Beszédes
Obuda University
Alba Regia Faculty
Székesfehérvár, Hungary
beszedes.bertalan@uni-obuda.hu
https://orcid.org/0000-0002-9350-1802

Stephen M. Kimathi
Biomatics and Applied Artificial Intelligence Institute,
John von Neumann Faculty of Informatics,
Obuda University, Budapest, Hungary
kimathi.stephen.muchai@nik.uni-obuda.hu
https://orcid.org/0009-0001-1478-4438

***Abstract*- The efficient processing of multiple analog signals in real time is a critical challenge in modern electronic systems, particularly in applications requiring high bandwidth, low latency, and low power consumption. This paper presents a comparative study of electronics architectures for simultaneous analog signal processing, with a focus on the conceptual design trade-offs between digital, analog, and hybrid approaches. Several implementation strategies are evaluated, including FPGA-based reconfigurable digital logic, FPAA-based adaptive analog computation, microcontroller-centric designs with integrated peripherals, and mixed architectures combining analog multiplexers, comparators, ADCs, and DACs. The analysis highlights the strengths and limitations of each paradigm. The results underline the importance of selecting architecture according to system-level requirements and suggest future directions for hybrid reconfigurable platforms that combine the programmability of digital logic with the efficiency of analog signal domains.**

Keywords - electronyics architecture, analog signal design, hybrid system design, mixed-signal systems, simultaneous analog processing, high-speed electronics

I. Introduction

The simultaneous real-time processing of analog signals has become a fundamental requirement in modern electronic systems, underpinning applications as diverse as sensor networks, biomedical instrumentation, advanced signal processing platforms, and industrial automation and control. Although digital technologies dominate contemporary system design, information originating from the physical environment is intrinsically analog in nature. As a consequence, the efficient and precise handling of analog signals remains a critical enabler of system performance. The central challenge lies in developing architectures capable of managing multi-channel, concurrent analog signals while simultaneously satisfying demanding constraints on bandwidth, latency, signal integrity, and energy efficiency.

A broad spectrum of architectural paradigms has been proposed to meet these requirements. Field-Programmable Gate Arrays (FPGAs) offer extensive parallelism, reconfigurability, and scalability, making them attractive for high-performance applications; however, they often incur significant power consumption and resource costs [1-3]. Digital Signal Processors (DSPs) provide specialized computational cores optimized for real-time signal processing, delivering high precision and efficiency for numerically intensive tasks such as filtering, spectral analysis, and adaptive algorithms. DSP-based solutions are particularly advantageous in multi-channel systems requiring complex algorithmic processing [4-6]. By contrast, Field-Programmable Analog Arrays (FPAAs) provide configurability directly in the analog domain, thereby enabling portions of signal processing to be performed without conversion to the digital domain. Nevertheless, their applicability is limited by device availability, resolution, and precision constraints [7-9]. Microcontroller-based solutions present highly integrated and cost-efficient platforms, yet their processing capacity is inherently restricted in scenarios involving high-speed, multi-channel data streams [10], [11].

Beyond these approaches, hybrid architectures – which integrate analog multiplexers, comparators, analog-to-digital converters (ADCs), and digital-to-analog converters (DACs) – have emerged as compelling alternatives that seek to exploit the complementary strengths of digital and analog subsystems. Such architectures can be tailored to specific application domains, enabling balanced trade-offs between accuracy, throughput, scalability, and power efficiency [12], [13].

This paper aims to provide a systematic assessment of these conceptual design strategies. We present a comparative analysis of FPGA-, FPAA-, and microcontroller-based solutions alongside hybrid analog–digital configurations. The evaluation emphasizes their relative strengths, inherent limitations, and practical deployment scenarios [14], [15]. On this basis, the discussion highlights design considerations that can inform the selection of the most suitable architecture for a given application domain and outlines prospective research directions in the development of reconfigurable hybrid platforms [16 - 18].

II. Parallel signal processing architectures

A. FPGA based architecture

True parallel data processing can be achieved using FPGA technology. The manufacturing process of FPGAs is optimized for digital logic, whereas the technology used for analog circuits (such as precision ADCs) is different. In the vast majority of FPGAs, either there is no ADC at all, or only one or two low-speed built-in ADCs are available (e.g., Xilinx Zynq-7010/7020 or Intel Altera MAX 10). These are insufficient for n×10 channels and simultaneous sampling; they are suitable only for auxiliary functions.

To achieve true parallelism, each sensor must be connected to a dedicated ADC, eliminating multiplexing delay. For precise synchronization, a common sampling clock can be distributed to all ADCs – see Fig. 1. The FPGA can perform simultaneous (pre)processing of the digitized signals, such as filtering, FFT, or fault detection.

Entry-level FPGAs (such as the Lattice iCE40, Gowin GW1N, or Xilinx Spartan-7) are available for around 5 – 130 USD, but they require separate ADCs. An 8-channel ADC capable of simultaneous sampling (e.g., AD7606-8, LTC1859) costs approximately USD 35–60 depending on the type. For example, 40 sensors would require 5 such ICs, costing at least USD 175. Solutions with fewer channels can use multiple smaller, cheaper ADCs that can be triggered simultaneously (e.g., ADS8320).

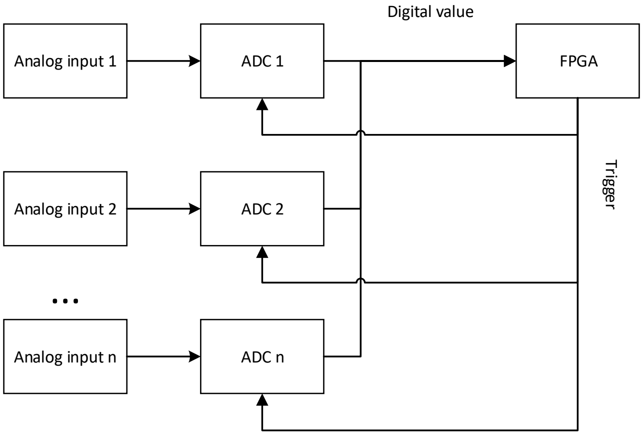


Figure 1: Parallel ADCs for simultaneous signal processing

B. DSP based architecture

Digital Signal Processors are specialized microprocessors designed specifically for high-speed numerical computation and real-time signal processing tasks. Unlike general-purpose MCUs, which are optimized for control logic and peripheral management, DSPs are architected to execute complex mathematical operations – such as multiplication engines, Direct Memory Access controllers, and hardware accelerators for Fast Fourier Transform or digital filtering operations. – extremely efficiently. These components enable deterministic, low-latency data processing with minimal CPU overhead.

DSPs often integrate high-performance ADC peripherals capable of a few channels of simultaneous sampling, synchronized triggering, and direct data streaming to memory or processing units. This makes them well suited for applications that demand precise timing and fast response, such as motor control, audio and image processing, vibration analysis, and real-time communications.

Compared to MCU and DSP analog to digital conversion capabilities, DSP offer higher sampling speeds, lower jitter, and tighter integration with signal-processing pipelines – see the details in Table 1. However, they tend to be more expensive and power-hungry, and they require more specialized programming skills. In both cases, an architecture similar to FPGA is required.

	DSP ADC Peripheral	**MCU ADC Peripheral**
Sampling speed	Very high (up to 10–100 MS/s)	Moderate (typically 0.1–2 MS/s)
Resolution	12–16 bit, stable even at high speed	Typically 10–12 bit, sometimes 16 bit but at lower speed
Parallel channels	Multiple true simultaneous sampling channels	Usually multiplexed, not truly parallel
Data transfer	Direct DMA access, pipelined data path	Often CPU-driven readout, slower data path
Synchronization	Supports multiple ADCs and external triggering	Limited, often only software-based
Signal-processing integration	ADC tightly coupled with MAC/FFT units for real-time processing	ADC acts only as a data source; processing done by CPU
Determinism / jitter	Hardware-level deterministic timing	Affected by interrupt handling, higher jitter
Target applications	Signal and image processing, control, communications	General-purpose measurement and control tasks

Table 1: Comparison of DSP and MCU analog-to-digital converter

C. FPAA based architecture

The FPAA offers an alternative for parallel processing of multiple analog signals and for creating dynamic signal-processing topologies. However, FPAAs are typically more of a complementary or specialized solution alongside MCU/DSP/FPGA systems rather than a full replacement for them. They are also well-suited for analog pre-processing to reduce power consumption.

In an FPAA, instead of logic gates, analog blocks (amplifiers, filters, comparators, integrators, switches, etc.) can be programmatically interconnected, allowing part of the signal processing to be carried out in the analog domain even before the ADC. Table 2 shows the different roles of using FPAAs in an embedded system analog signal path.

Anadigm and Okika are virtually dominant in the market, with prices ranging roughly from $100 to several hundred USD per chip. A peculiarity is that each device can handle only a few analog channels (4–8 inputs), so implementing 40 channels, as in the previous example, would require multiple ICs. Analog input expansion is also limited when using analog multiplexers – see Fig. 2. Development requires a dedicated software environment, which may be either closed-source or open-source depending on the manufacturer.

Solution	**Role of FPAA**	**Advantage**
FPGA + ADC	analog signal filtering, amplification	energy-efficient signal chain
DSP + ADC	analog signal filtering, amplification	energy-efficient signal chain, lower DSP load
MCU + MUX + ADC	analog signal filtering, amplification	energy-efficient signal chain, reduced MCU load
MCU + Comparator + DAC	FPAA as comparator + programmable thresholds	digitally controlled analog trigger logic

Table 2: Roles and advantages of FPAAs in a signal path

C. Multi-core microcontroller architecture

Multicore microcontrollers, such as the Parallax Propeller (priced around $10–20 per IC or $40–100 per development board), offer an alternative approach to simultaneous analog signal processing.

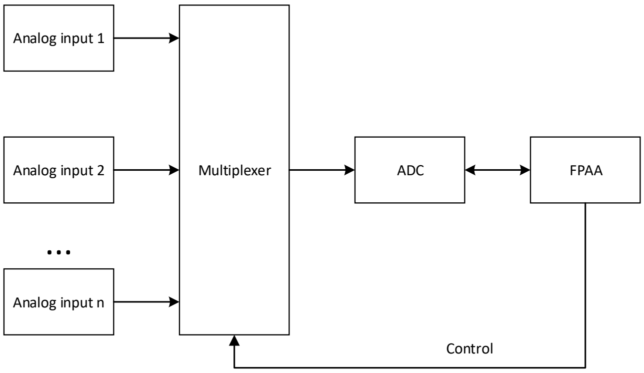


Figure 2: External multiplexer and ADC architecture

Traditionally, analog signal processing operates in a sequential manner: a single microcontroller reads multiple inputs in a time-multiplexed fashion and performs the required computations one after another. This approach is adequate for simple tasks, but as soon as multiple high-frequency or real-time analog signals need to be processed, sequential operation can lead to timing uncertainty, CPU overload, and complex interrupt management.

A multicore architecture addresses this issue at the hardware level. The Propeller microcontroller features eight independent, identical processor cores, all connected through a shared hub memory. Each core can independently handle peripherals, execute programs, and even perform A/D conversion. The deterministic memory access scheme of the hub ensures predictable timing, which is essential for maintaining stability in analog signal processing.

In such an architecture, different signal-processing tasks can be assigned to dedicated cores. For example, one core can perform continuous sampling, while another handles preprocessing operations such as filtering, averaging, or Fourier analysis. A third core may transmit processed data over a serial or wireless interface, while a fourth is responsible for updating display information.

Since each core operates entirely independently, the system can simultaneously monitor and process multiple analog channels in real time without relying on interrupt-driven scheduling or facing timing race conditions. The deterministic timing of this architecture enhances system reliability, which is a critical factor in control and diagnostic applications. Furthermore, programming becomes more straightforward: developers can distribute tasks across truly independent cores instead of managing software-based threads.

Of course, there are limitations. The Propeller 1 (P8X32A) cores are relatively simple, so complex digital signal processing (DSP) operations can only be performed with limited efficiency. In addition, the built-in analog-to-digital conversion relies on software-based methods, which cannot match the resolution or speed of dedicated ADC-equipped hardware. Nevertheless, the flexibility of the multicore design allows the integration of external ADCs, whose data can be processed in parallel by separate cores.

The newer Propeller 2 (P2X8C4M64P) introduces a pipelined architecture capable of handling more advanced 32-bit mathematical operations, further expanding the range of real-time analog signal processing applications. Expanding the number of ADC input ports also requires the use of external hardware components.

III. Sequential signal processing architectures

Analog signal processing architectures can be broadly categorized into parallel and sequential (or serial) approaches, each with distinct operational principles and trade-offs. In sequential analog architectures, the signal is processed in a step-by-step manner. Each processing stage completes its operation on the signal before passing it to the next stage. This approach is conceptually simple and requires fewer hardware resources, since only one processing element is active at a time. Sequential architectures are typically implemented using cascaded analog components, such as filters, amplifiers, or modulators, where the output of one stage serves as the input to the next. While these architectures are hardware-efficient and easier to design, they are inherently limited in processing speed, as each operation must wait for the completion of the previous one, and they may accumulate noise and distortion over multiple stages.

D. Microcontroller based architecture with analog multiplexer

A microcontroller has a limited number of analog inputs, and expanding them becomes necessary due to the large number of measurement and control signals. The analog signals to be measured are connected through an analog multiplexer (e.g. CD74HCx4067) to either the pin associated with the microcontroller's analog-to-digital converter or to an external ADC hardware module. For high-speed and high-resolution ADC modules, this provides an effective time-division multiplexing solution.

When multiple external or internal ADC peripherals are used, it is advisable to always measure the same analog signals with the respective ADC modules in order to avoid measurement errors resulting from differences between the converter peripherals. The measurement of short-duration analog signals can be aided by sample-and-hold circuits. It is worth to mention there are ADCs with programmable analog input multiplexers and on-board sample and hold circuitry – for example the MCP3208.

E. Microcontroller based architecture with comparators

To avoid constant polling and data processing, it is recommended to design a system that performs sampling and intervention only when a significant signal change occurs. During the microcontroller's sleep state or instruction execution, it may fail to detect non-nominal variations in the analog signal.

An external hybrid circuit designed as an interrupt request system can continuously monitor the output voltage level and, when a voltage deviation beyond the allowed threshold occurs, issue an interrupt request to the microcontroller. The condition for triggering an interrupt is the deviation of the monitored input from its reference range or connection.

$$INT = \bigvee_{m=1}^{n} r_{min_n} < a_n < r_{max_n}$$

where a_n is the measured circuit voltage, r_{min_n} and r_{max_n} are the minimum and maximum threshold values of the monitored circuit voltage, and n is the number of analog inputs.

$$MUX\ output = \sum Analog\ MUX\ (A_{MUX}, a_n)$$

When the interrupt condition is met, the signal line to be examined further can be selected based on the expression given below.

Fig. 3. illustrates the explanatory operational diagram. Alternatively, a built-in comparator (as part of the microcontroller's internal peripherals) can also be used—especially when the number of analog input signals is low. In this case, the reference voltage value can be set in software as a register value. The internal comparator peripheral can likewise generate an interrupt request to the microcontroller. For external components LMV7219 or MAX9060 can be a solution for the comparator, comparators with on-board DACs are rear, but the MCUs built in DAC periphery or an external DAC IC also can be used (e.g. MCP4728 is a 4 channel DAC with I^2C interface). Fore price considerations a reference voltage source and digital potentiometers can be used to generate the comparators reference voltage (e.g. MCP4011, AD5242, MAX5400).

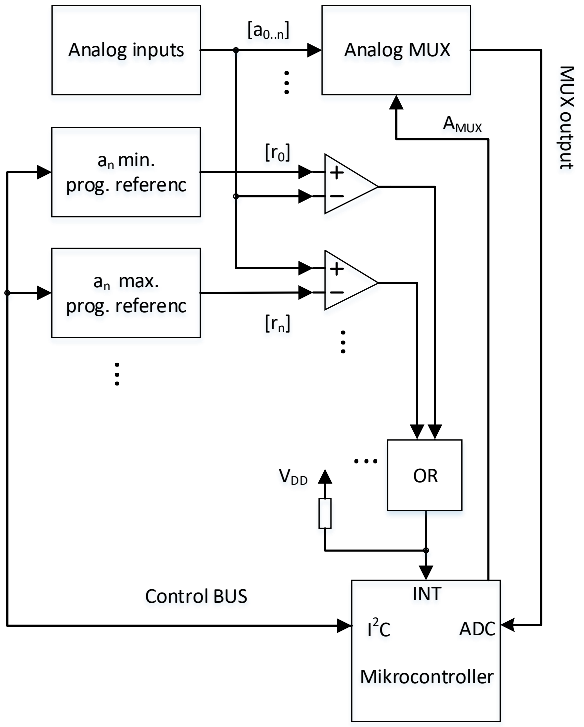


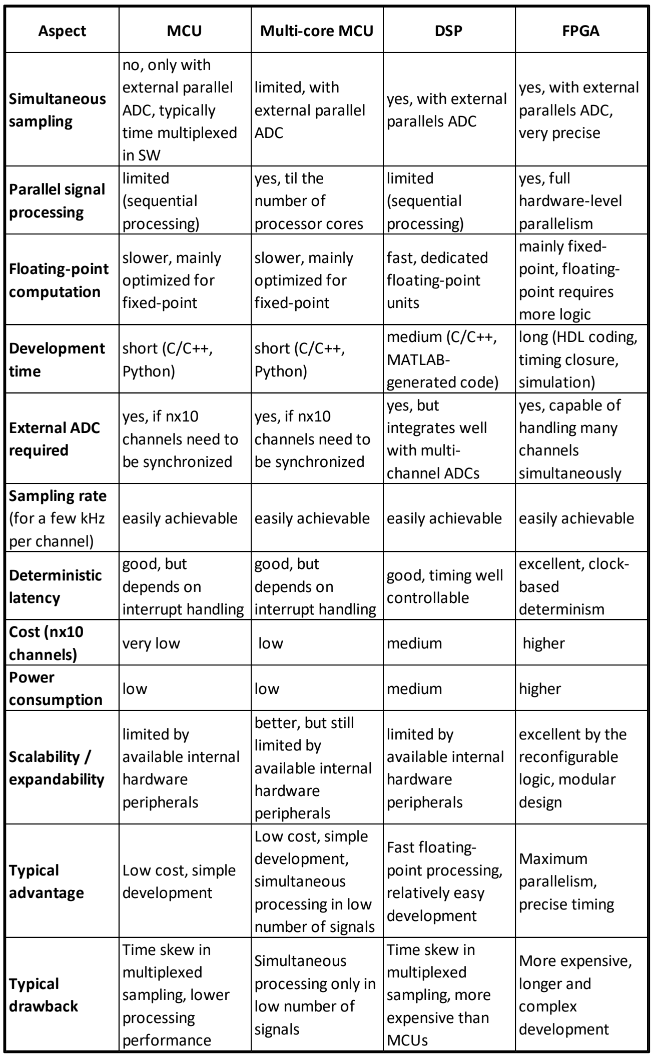
Figure 3: Interrupt base analog reference

The choice between sequential and parallel architectures depends on design priorities. Sequential designs are favored when hardware simplicity, low power consumption, or cost is critical, whereas parallel designs are selected when speed and real-time performance are essential. Table 3 provides a summary overview of the options, features and drowbacks.

CONCLUSION

This study has provided a comprehensive analysis of electronic architectures for simultaneous analog signal processing, including FPGA-, DSP-, FPAA-, microcontroller-based, and hybrid solutions. The comparative evaluation demonstrates that each paradigm exhibits distinct advantages and limitations: FPGAs excel in high-speed, parallel processing but are constrained by power consumption and area requirements; DSPs offer high-precision, numerically intensive real-time processing suitable for complex filtering, spectral analysis, and adaptive algorithms; FPAAs provide analog-domain configurability with low-latency and energy-efficient operation, though their precision and availability are limited; microcontrollers offer cost-effective, integrated platforms but are restricted in throughput for multi-channel real-time applications; and hybrid analog–digital architectures enable a flexible trade-off among accuracy, speed, and energy efficiency.

Overall, the literature indicates that no single universal solution exists. The optimal choice of architecture depends on the specific requirements of the target application, including bandwidth, latency, signal fidelity, and power constraints. Emerging trends toward hybrid and reconfigurable analog–digital systems suggest that such combinations represent the most promising direction for future developments in parallel, real-time analog signal processing. By leveraging the complementary strengths of analog, digital, and DSP-based domains, these platforms can deliver scalable, efficient, and high-performance solutions tailored to the evolving needs of modern electronic systems.



Aspect	MCU	Multi-core MCU	DSP	FPGA
Simultaneous sampling	no, only with external parallel ADC, typically time multiplexed in SW	limited, with external parallel ADC	yes, with external parallels ADC	yes, with external parallels ADC, very precise
Parallel signal processing	limited (sequential processing)	yes, til the number of processor cores	limited (sequential processing)	yes, full hardware-level parallelism
Floating-point computation	slower, mainly optimized for fixed-point	slower, mainly optimized for fixed-point	fast, dedicated floating-point units	mainly fixed-point, floating-point requires more logic
Development time	short (C/C++, Python)	short (C/C++, Python)	medium (C/C++, MATLAB-generated code)	long (HDL coding, timing closure, simulation)
External ADC required	yes, if nx10 channels need to be synchronized	yes, if nx10 channels need to be synchronized	yes, but integrates well with multi-channel ADCs	yes, capable of handling many channels simultaneously
Sampling rate (for a few kHz per channel)	easily achievable	easily achievable	easily achievable	easily achievable
Deterministic latency	good, but depends on interrupt handling	good, but depends on interrupt handling	good, timing well controllable	excellent, clock-based determinism
Cost (nx10 channels)	very low	low	medium	higher
Power consumption	low	low	medium	higher
Scalability / expandability	limited by available internal hardware peripherals	better, but still limited by available internal hardware peripherals	limited by available internal hardware peripherals	excellent by the reconfigurable logic, modular design
Typical advantage	Low cost, simple development	Low cost, simple development, simultaneous processing in low number of signals	Fast floating-point processing, relatively easy development	Maximum parallelism, precise timing
Typical drawback	Time skew in multiplexed sampling, lower processing performance	Simultaneous processing only in low number of signals	Time skew in multiplexed sampling, more expensive than MCUs	More expensive, longer and complex development

Table 3: Comparison of embedded cogic units based on simultaneous signal processing capabilities

ACKNOWLEDGMENT

The authors would like to thank all the faculty staff and member of Obuda University Alba Regia Faculty, that provide help and assistance throughout the project completion.

REFERENCES

[1] Mbihi, J., Otam, U. S., Moffo, B. L., Ngounou, C. E. G. A novel FPGA-based multi-channel signal acquisition system using parallel duty-cycle modulation and application to biologic signals: design and simulation. Journal of Electrical Engineering, Electronics, Control and Computer Science, 7(2), 13-20. 2020.

[2] Hu, W., Choi, Y., Hong, K. J., Kang, J., Jung, J. H., Huh, Y. S., ... & Chung, Y. Free-running ADC-and FPGA-based signal processing method for brain PET using GAPD arrays. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 664(1), 370-375. 2012.

[3] Anttila, L., Lampu, V., Hassani, S. A., Campo, P. P., Korpi, D., Turunen, M., ... & Valkama, M. Full-duplexing with SDR devices: Algorithms, FPGA implementation, and real-time results. IEEE Transactions on Wireless Communications, 20(4), 2205-2220. 2020.

[4] Kester, W. (Ed.). Mixed-signal and DSP design techniques. Newnes. 2003.

[5] Stranneby, D. Digital signal processing and applications. Elsevier. 2004.

[6] Smith, S. Digital signal processing: a practical guide for engineers and scientists. Newnes. 2003.

[7] Györök, G. Spectrum Analyzer with FPAA. In Recent Advances in Intelligent Engineering: Volume Dedicated to Imre J. Rudas' Seventy-Fifth Birthday (pp. 283-290). Cham: Springer Nature Switzerland. 2024.

[8] Györök, G., Makó, M., Shvets, O. Educational universal analog circuit measurement laboratory with FPAA. In 2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES) (pp. 000021-000026). IEEE. 2023.

[9] Györök, G., Makó, M. Novel applications of FPAAs in hybrid feedback circuits. In 2022 IEEE 26th International Conference on Intelligent Engineering Systems (INES) (pp. 000159-000162). IEEE. 2022.

[10] Alexander Baklanov, Svetlana Grigoryeva, György Györök. Control of LED Lighting Equipment with Robustness Elements, Acta Polytechnica Hungarica (1785-8860 1785-8860): 15 3 pp 105-119. 2016.

[11] G. Gyorok and M. Mako, "Configuration of EEG input-unit by electric circuit evolution," *2005 IEEE International Conference on Intelligent Engineering Systems, 2005. INES '05.*, Spain, 2005, pp. 255-258, doi: 10.1109/INES.2005.1555168.

[12] Aizhan Zhaparova, Dimitry Titov, Alexander Y. Baklanov, *Gyorgy* Gyorok. Study of the Effectiveness of Switching-on LED Illumination Devices and the Use of Low Voltage System in Lighting. Acta Polytechnica Hungarica (1785-8860 1785-8860): 12 5 pp 71-80. 2015.

[13] Attila Sáfár, Bertalan Beszédes. Educational Aspects of a Modular Power Management System. In: Orosz, Gábor Tamás (szerk.) AIS 2019 : 14th International Symposium on Applied Informatics and Related Areas organized in the frame of Hungarian Science Festival 2019 by Óbuda University. Székesfehérvár, Magyarország : Óbudai Egyetem (2019) pp. 163-166. , 4 p.

[14] S. Kimathi, and B. Lantos. Robust adaptive formation control of fixed wing UAVs using disturbance estimators. in IEEE 24th International Symposium on Computational Intelligence and Informatics, Budapest, Hungary November 2024.

[15] S. Kimathi, and B. Lantos. Integrated path tracking, and control of a fixed wing UAV based on dual quaternion parameterized dynamics. in 47th MIPRO ICT and Electronics Convention, Opatija, Croatia, May 2024.

[16] Odry, A., Kecskes, I., Pesti, R., Csik, D., Stefanoni, M., Sarosi, J., & Sarcevic, P. (2025). NN-augmented EKF for Robust Orientation Estimation Based on MARG Sensors. International Journal of Control, Automation and Systems, 23(3), 920-934.

[17] Akos Odry, Istvan Kecskes, Richard Pesti, Dominik Csik, Massimo Stefanoni, Imre Kovacs, Edit Laufer, Peter Sarcevic. Feedforward Neural Network for Orientation Estimation Under Magnetic and Acceleration Disturbances. Acta Polytechnica Hungarica Vol. 22, No. 12, 2025. ISSN: 1785-8860. DOI: 10.12700/APH.22.12.2025.12.4

[18] Imre Kovacs, Massimo Stefanoni, Richard Pesti, Dominik Csik, Peter Sarcevic and Ákos Odry. Evaluation of Optimal Covariance Models for EKF-based Wheeled Mobile Robot Localization. Acta Polytechnica Hungarica Vol. 22, No. 12, 2025. ISSN: 1785-8860. DOI: 10.12700/APH.22.12.2025.12.10