

Conference proceedings
Sborník odborného semináře

Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2026
New Methods and Practices in the Instrumentation, Automatic Control and Informatics 2026

27. - 29. 5. 2026 , Kácov

ISBN 978-80-01-07602-6

OBSAH

Nathan Ng'ambi
TUNING OF FOPID CONTROLLER USING GENETIC ALGORITHM AND PARTICLE SWAM OPTIMIZATION 3

Maksym Hrytsenko
ANALYSIS OF ONLINE AND OFFLINE SPEECH RECOGNITION APPROACHES FOR REAL-TIME SYSTEMS 14

Volodymyr Dudavskyi
COMPARATIVE ANALYSIS OF RETRIEVAL-AUGMENTED GENERATION PIPELINES FOR STRUCTURED PROCEDURAL DOCUMENTATION 20

Jan Rollo, Vladimir Hlavac
DETECTION OF AI-GENERATED IMAGES: FREQUENCY SPECTRUM ANALYSIS 30

Uliana Finaeva
STUDIE PROVEDITELNOSTI SPOJOVÁNÍ SKLA A KOVU POMOCÍ INDIA PRO MONTÁŽ OPTICKÝCH PRVKŮ 41

Patrik Brejla
NÁVRH A IMPLEMENTACE WEBOVÉ APLIKACE PRO MONITORING PRŮMYSLOVÝCH ZAŘÍZENÍ 46

Daniel Čech
IDENTIFIKACE A ŘÍZENÍ HYDRAULICKÉHO SUBSYSTÉMU MECHATRONICKÉHO DÁVKOVAČE KAPALIN 52

Pavel Hron
MODERNÍ TRENDY V PREDIKTIVNÍM ŘÍZENÍ (MPC) A ENERGETICKÉM MANAGEMENTU: OD FYZIKÁLNÍCH MODELŮ K AI A DIGITÁLNÍM DVOJČATŮM 57

Ondřej Pecina
PIPELINE PRO AUTOMATIZOVANÉ NASAZENÍ MODELŮ Z PYTORCH NA FPGA KRIA KV260 S VITIS AI 3.5 60

Matěj Mansfeld
REŠERŠE MODELŮ KOROZIVITY ATMOSFÉRY A MODELŮ STROJOVÉHO UČENÍ PRO ÚČELY PREDIKCE KOROZE HLINÍKOVÝCH KONSTRUKCÍ 66

Oto Vrána
VÝPOČETNĚ ÚSPORNÁ METODA PROSTOROVÉHO SROVNÁVÁNÍ CT SNÍMKŮ NOSNÍCH DUTIN 72

TUNING OF FOPID CONTROLLER USING GENETIC ALGORITHM AND PARTICLE SWAM OPTIMIZATION

***Nathan Ng'ambi*[1]**

1 Department of Instrumentation and Control Engineering, Czech Technical University in Prague, Technicka 4, Prague, Czech Republic

ngambnat@cvut.cz

Abstract: This article presents a comparative study on the tuning of Fractional order PID (FOPID) controller using Genetic Algorithms (GA) and Particle Swam Algorithm (PSO). The objective it to optimize all the five parameters i.e. proportional gain (K_p), integral gain (K_i) derivative gain (K_d), integral order (λ) and derivative order (μ) for various linear systems including 2^{nd}, 4^{th} and 6^{th} order dynamics with some time delay. Performance evaluation is based on time-domain criteria such as rise time,settling time,overshoot, steady state error and integral error indices. Simulation results show that Particle Swarm Optimization converges faster and provides superior performance for higher order and delayed systems, while Genetic Algorithm performs reliably for low order systems but exhibits limitations as system complxity increases. The aricle highlights the strengths and weaknesses for both algorithms and discusses practical consideration for FOPID tunning.

Keywords: FOPID Controller, Fractional Order Control, Metaheuristic, Genetic Algorithm, Particle Swarm Optimization, Time Delay Systems, Controller Optimization

1 Introduction

Fractional Order PID controllers provide (FOPID or $PI^{\lambda}D^{\mu}$) provides an alternative to PID controller by introducing two additional tuning parameters i.e. λ and μ which enhance control flexibility and robustness [1]. In modern control systems where high performance and stability margins are critical, these fractional orders offer improved tuning precision.

Metaheuristic optmization methods such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) have been widely used to automatically tune controller parameters [2] [3]. This article investigates the application of GA and PSO in tuning all the five parameters of FOPID controllers with a focus on comparing their performance across different dynamic systems

2 Background and Literature Review

Background PID controllers are widely used in industrial process control due to their simplicity and ease of implementation. However, they often struggle with complex dynamic systems involving nonlinearity, time delay or higher-order dynamics. To address these limitations, Fractional Order PID (FOPID) controllers have been proposed, which extend conventional PID by incorporating two additional parameters the inegral order λ and derivative order $\boldsymbol{\mu}$. This generalisation allows finer tuning and improved robustness by shaping system responses more flexibly in both time and frequency domain [1].

FOPID controllersaregoverned by the transfer function:

$$C(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu} \quad (1)$$

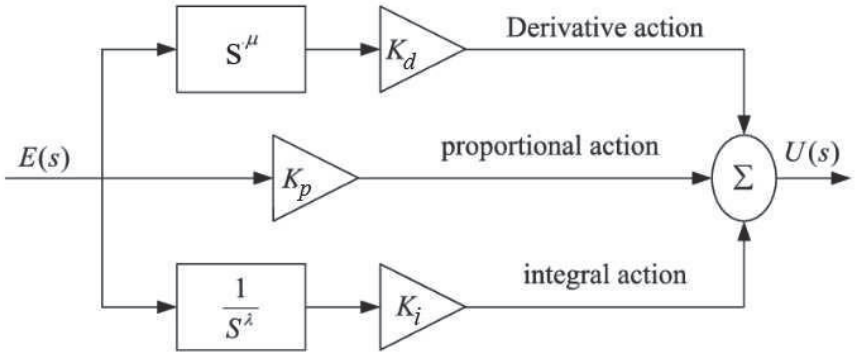


Figure 1. Structure of a fractional-order PID ($PI^{\lambda}D^{\mu}$) controller. Adapted from Podlubny, 1999. Image accessed via ResearchGate.

Where $K_p, K_i$and K_d represent the proportional, integral and derivative gains and λ and μ are the non-integer orders of integration and differentiation. The fractional orders give more freedom in shaping the system's response both in the time and frequency domains [1].Despite their advantages, FOPID controllers are difficult to tune due to their five-dimensional parameter space. Analytical methods or classical tuning rules (such as Ziegler-Nichols) are often insufficient or inapplicable. As a result, researchers have adopted metaheuristic optimisation techniques such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) for automatic parameter tuning.

Previous studies have shown that both GA and PSO can effectively tune FOPID parameters. For Instance, Valerio and da Costa [2] explored heuristic rules and optimisation methods to tune fractional controllers, and Monje et al. [1] provided a systematic framework for FOPID implementation in control applications. Chen et al.[5] emphasized the need for inelligent optimization in fractional control due to lack of universal analytical methods.

In this study, these findings are extended by conducting a detailed comparative analysis of GA and PSO on tuning FOPID controllers across systems of varying complexity. The evaluation focuses on time domain performance metrics such as rise time, overshoot, settling behaviour, steady state error and integral error indices, providing a comprehensive assessment of controller performance. Methodology.

2.1 Genetic Algorithm

This is an evolutionary optimization technique inspired by the principles of natural selection and genetics. It operates on a population of candidate solutions and iteratively improves them through selection, crossover and mutation operators. GA is particularly effective for solving complex, non-linear and multi-modal optimisation problems, where gradient-based methods fail or become trapped in local minima [2]. Due to its population- based search and stochastic nature, GA is well suited for tuning all five parameters of the FOPID controller simultaneously.

2.2 Particle Swarm Optimization (PSO)

This is a population-based stochastic optimisation method inspired by the social behaviour of bird flocking and fish schooling. In PSO, each particle represents a potential solution and updates its position in the search space by combining its own best experience with the global best experince of the swarm. PSO is known for its fast convergence and simple implementation, making it attractive for continous parameter optimization problems such as controller tuning [3].

In this article, both GA and PSO are applied to tune the five parameters of the FOPID controller:

- Proportional gain $\boldsymbol{K_p}$
- Integral gain $\boldsymbol{K_i}$
- Derivative gain $\boldsymbol{K_d}$
- Fractional Inegral Order $\boldsymbol{\lambda}$
- Fractional Derivative Order $\boldsymbol{\mu}$

The tuning objective is formulated as a minimisation problem, where a performance-based cost function combines key time-domain indices including integral of Absolute Error (IAE), overshoot, rise time, settling time and steady state error. This formulation allows a fair and quantitative comparison between GA and PSO tuned controllers [4].

2.3 Test Systems and Their Transfer Functions

To evaluate the performance of GA and PSO under increasing levels of complexity, a set of plant models with different dynamic characteristics was selected.

(i) Second-Order System

$$G(s) = \frac{1}{(0.5s+1)(s+1)} \tag{2}$$

This system represents a simple stable plant and serves as a baseline case for evaluating tuning accuracy and covergence behaviour of the algorithms.

(ii) Second-Order System with Time Delay

$$G(s) = \frac{e^{-0.5s}}{(0.5s+1)(s+1)} \tag{3}$$

The inclusion of time delay introduces phase lag and tests the ability of GA and PSO to maintain stability and robustness in delayed systems.

(iii) Fourth-Order Oscillatory System

$$G(s) = \frac{1}{s^4+2s^3+3s^2+2s+1} \quad (4)$$

This system exhibits oscillatory behaviour and higher order dynamics providing a more challenging tuning scenario where the benefits of fractional orders $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ become more pronounced.

(iv) Fourth-Order System with Time Delay

$$G(s) = \frac{e^{-0.4s}}{s^4+2s^3+3s^2+2s+1} \quad (5)$$

This model combines high order dynamics with time delay, significantly increasing tuning difficulty and allowing deeper evaluation of algorithm robustness.

(v) Sixth-Order Oscillatory System

$$G(s) = \frac{1}{s^6+3s^5+5s^4+4s^3+3s^2+2s+1} \quad (6)$$

The sixth-order system represents a highly complex and oscillatory plant, used to access the limitations of GA and PSO when applied to high-order dynamics without model reduction.

These systems were selected to systematically investigate how GA and PSO scale with increasing system complexity, oscillatory behaviour, and with time delay effects, which are common challenges in practical control applications [1],[4].

The primary steps include:

- Defining the plant model ($2^{nd}, 4^{th}, 6^{th}$ order)
- Implementing GA and PSO algorithm for parameter tuning
- Designing a performance-based cost function using IAE, overshoot, settling time and steady-state error
- Extracting and comparing performance metrics

2.4 Algorithm Parameters

Genetic Algorithm (GA) parameters

- Population: 50
- Crossover: scattered (probability= 0.8)
- Mutation: Gaussian (probability= 0.2)
- Selection: Stochastic Uniform

Paricle Swarm Optimization (PSO) Parameters

- Number of particles: 30
- Inertia weight: 0.7 (with damping)
- Cognitive coefficient c_1: 1.5
- Social coeeficient c_2: 1.5

3 Simulation Results and Analysis

The performance of GA and PSO is assessed using time domain metrics such as rise time, settling time,overshoot, steady state error and integral of absolute error (IAE) as well as frequency domain metrics like phase margin and gain margin. The goal is to observe both optimization techniques behave under varying plant complexities and delays.

4.1 2^{nd} Order System

Both GA and PSO achieved excellent performance for the second order system. The GA tuned controller produced a smooth response with zero overshoot and good steady-state accuracy. PSO achieved a slightly faster rise time and lower integral error, reflecting faster convergence. For simple low-order dynamics, both algorithms are effective, with PSO showing a minor advantage in speed.

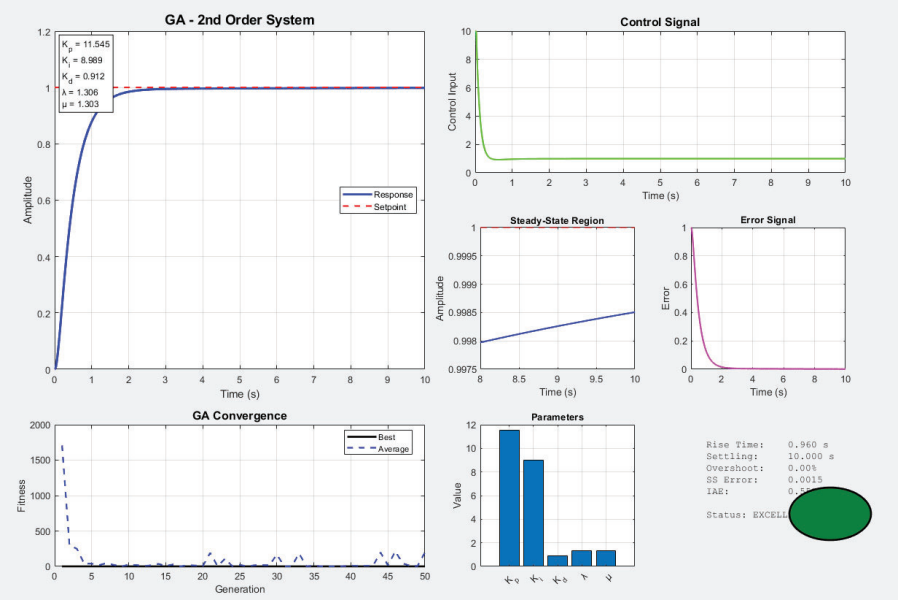


Figure 2 Step Response for 2^{nd} *Order System-GA*

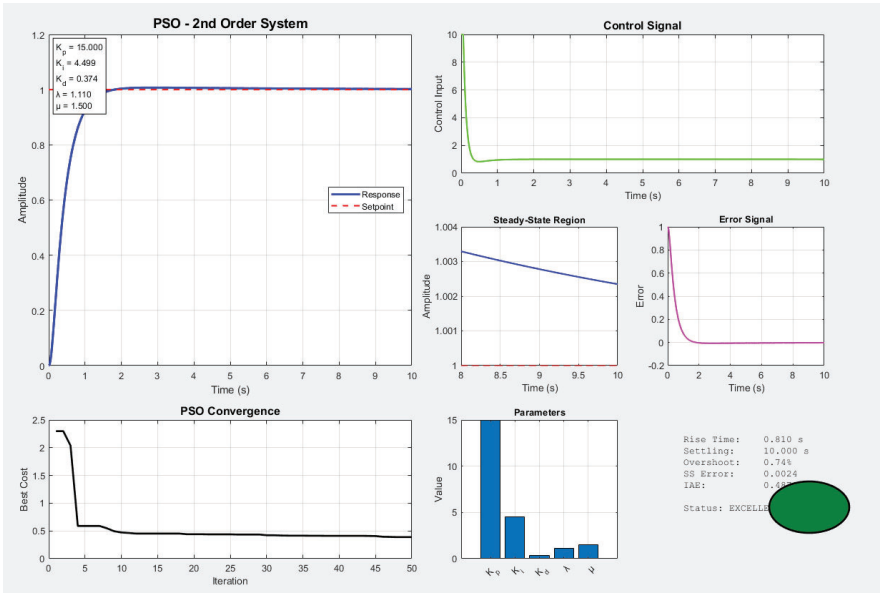


Figure 3 Step Response for 2nd Order System-PSO

4.2 2nd Order System with Time Delay

When time delay was introduced, GA maintained stable performance with moderate overshoot and acceptable steady-accuracy. PSO showed increased sensitivity to the delay to the delay, resulting in higher steady state error and increased integral error. Although PSO converged quickly, its long term accuracy degraded compared to GA in this case.

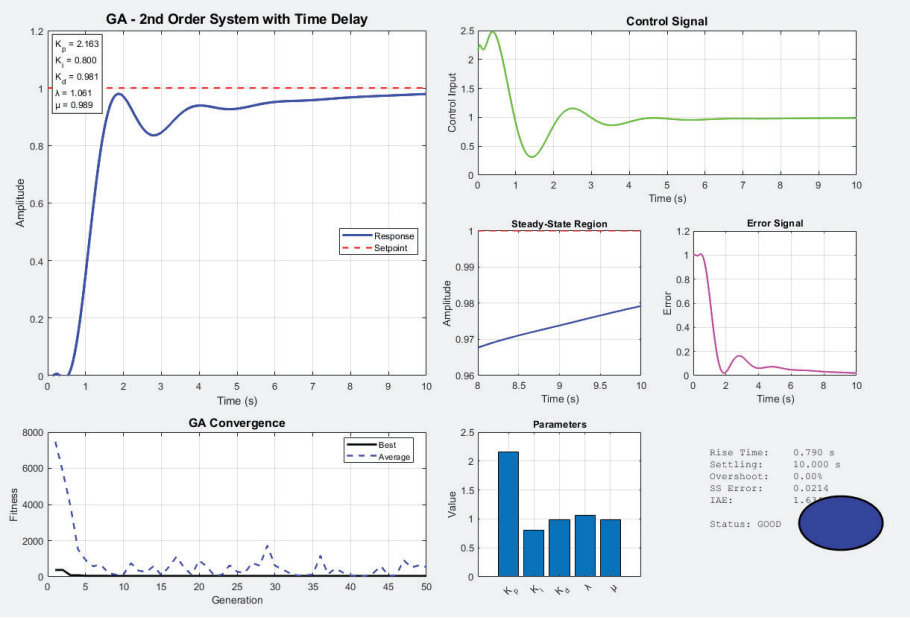


Figure 4 Step Response for 2nd Order with Time Delay-GA

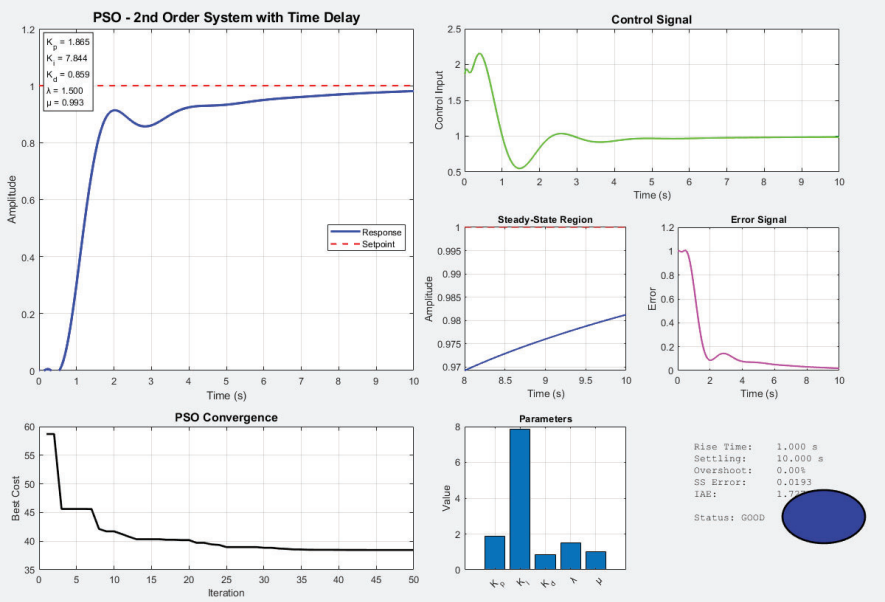


Figure 5 Step Response for 2nd Order with Time Delay-PSO

4.3 4th Order Oscillatory System

For the Fourth-Order oscillatory without delay, PSO outperformed GA, the PSO-tuned controller achieved faster rise time, better damping, and lower steady-state error. GA produced a slower response with more persistent oscillations and longer settling time. Convergence plots indicated more efficient optimization by PSO for this system.

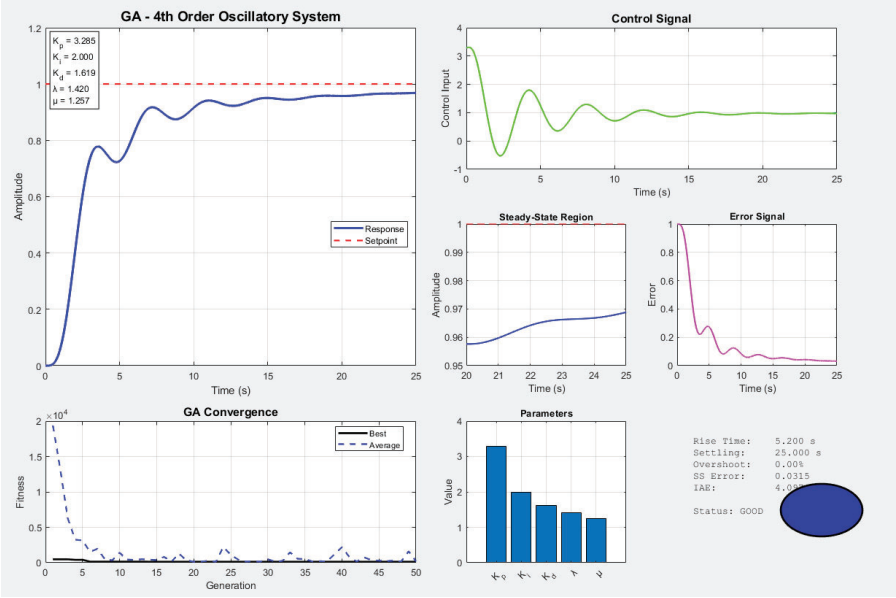


Figure 6 Step Response of 4th Order Oscillatory System-GA

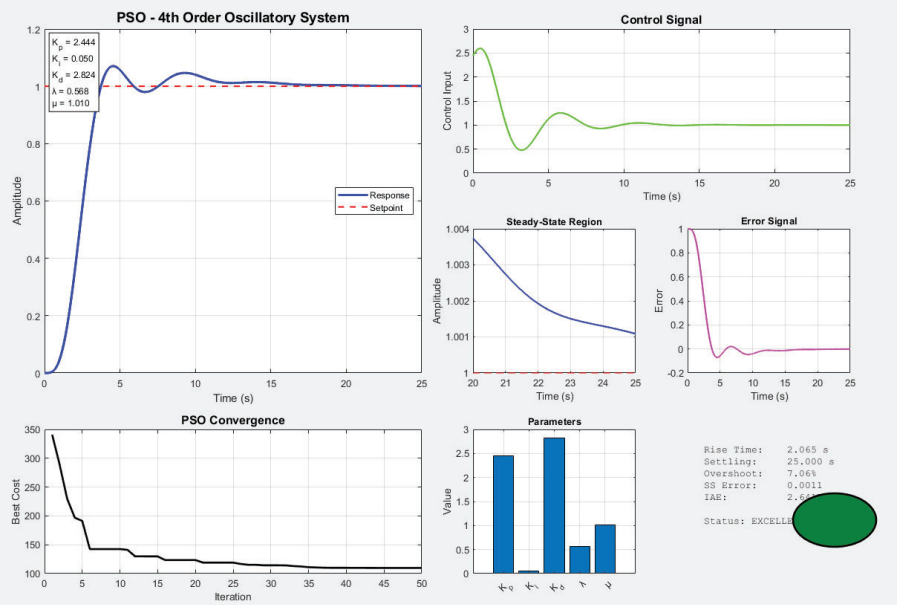


Figure 7 Step Response of 4th Order Oscillatory System-PSO

4.4 4^{th} Order System with Time Delay

The presence of both high order and time delay incresed tuning difficulty. PSO maintained acceptable stability and achieved a better compromise between overshoot and steady state accuracy. GA produced a conservative controller with nrgligible overshoot but large steady-state error. Indicating insufficient integral action. Overall, PSO demonstrated superior robustness for delayed $\mathbf{4^{th}}$ order dynamics.

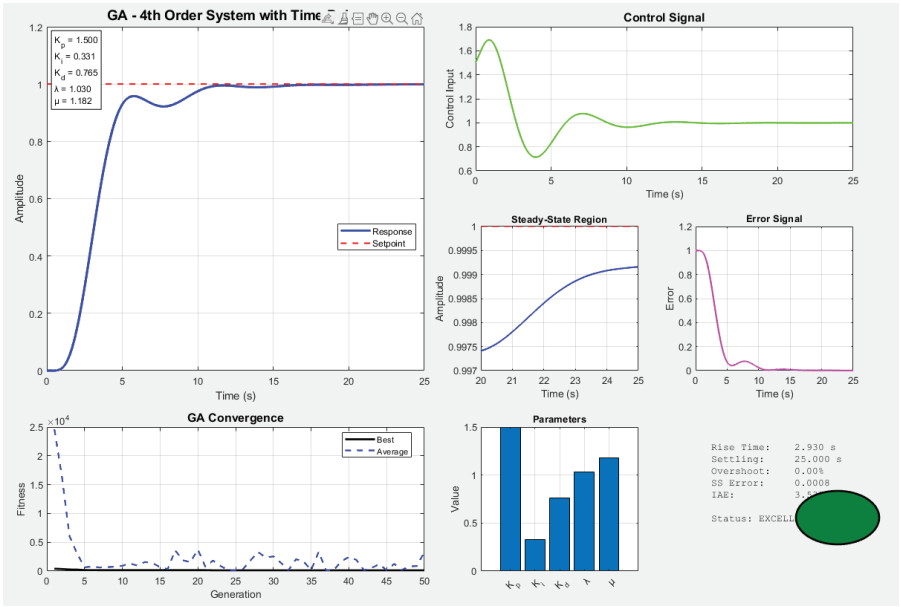


Figure 8 Step Response of 4th Order System with delay-GA

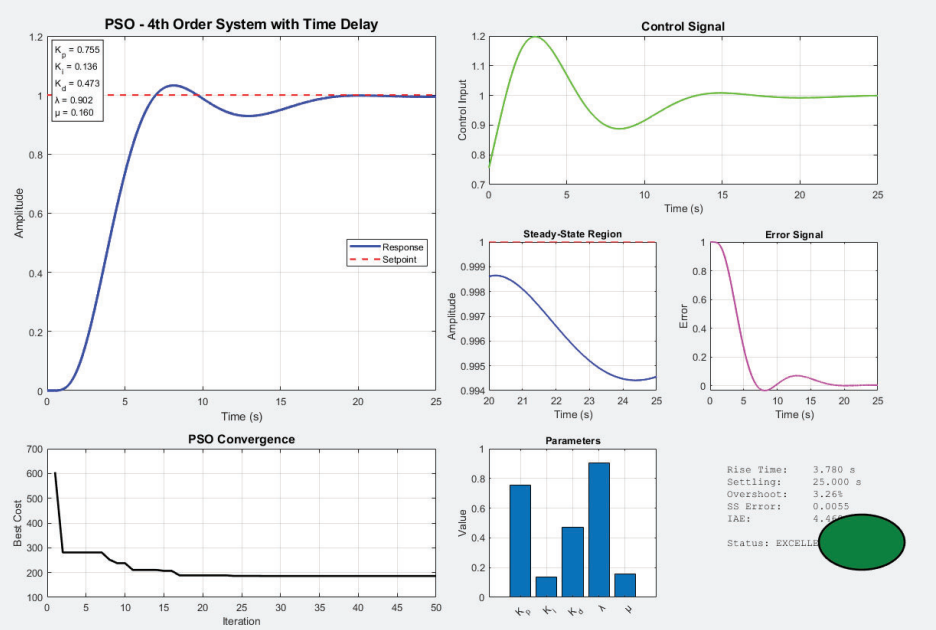


Figure 9 Step Response of 4th Order System with delay -PSO

4.5 6^{th} Order Oscillatory System

For the 6^{th} oscillatory system, PSO did better than GA. The PSO tunned controller achieved a bounded response with controlled overshoot and near zero steady state error. In contrast, the GA-tuned controller failed to track the reference acculately and exhibited poor steady state behaviour. These results highlight PSO's superior capability in hanling highly complex oscillatory systems.

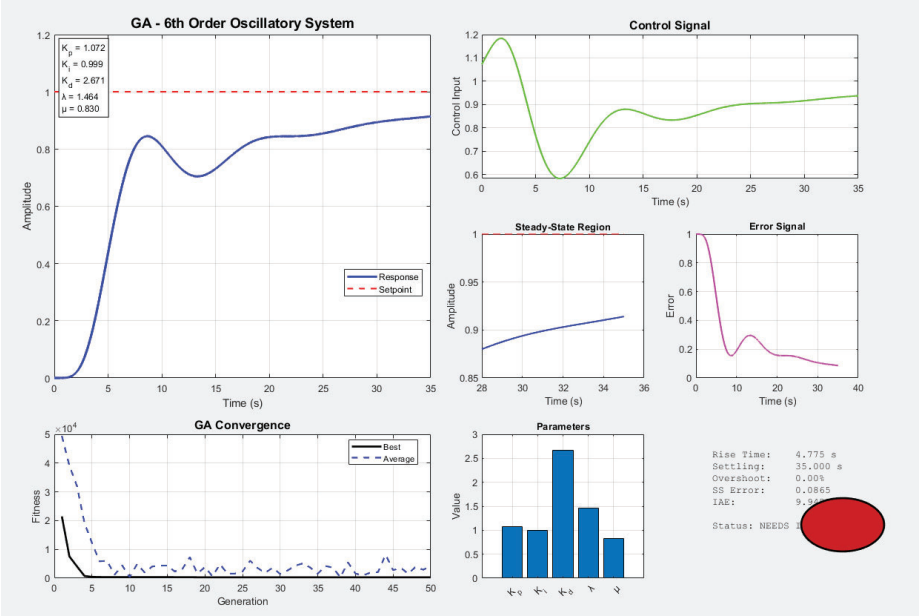


Figure 10 Step Response of 6th Order Oscillatory System-GA

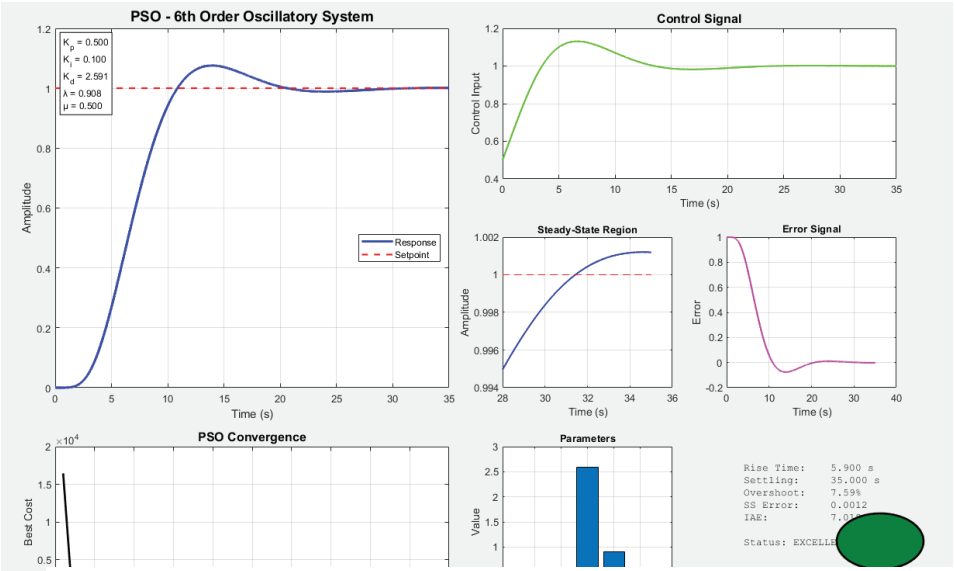


Figure 11 Step Response of 6th Order Oscillatory System-PSO

5 Conclusion

This study compared Genetic Algorithm and Particle Swarm Optimization for tuning Fractional Order PID controllers across systems of different orders and with time delay. The results show that both algorithms successfully optimize FOPID parameters and achieve stable closed loop control. Particle Swarm Optimization consistently provided faster convergence and superior performance for higher order and oscillatory system. Genetic Algorithms performed well for low order and delayed systems, where improved steady state accuracy was observed in some cases. As system complexity increased PSO demonstrated better scalability and robustness than GA. These findings confirm that optimization based tuning is well suited for FOPID controllers and that algorithm selection should depend on system dynamics and control objectives, future work will focus on hybrid optimization approaches and experimental validation.

6 References

[1] I. Podlubny, "Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers," IEEE Transactions on Automatic Control, vol. 44, no. 1, pp. 208–214, 1999. ISSN: 0018-9286. DOI: 10.1109/9.739144.

[2] D. Valério and J. S. da Costa, "Tuning of fractional PID controllers with Ziegler–Nichols-type rules," Signal Processing, vol. 86, no. 10, pp.2771–2784, 2006. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2006.02.020.

[3] Y. Q. Chen, I. Petráš, and D. Xue, "Fractional order control – a tutorial," Proceedings of the American Control Conference, St. Louis, USA, pp. 1397–1411, 2009. ISSN: 0743-1619. DOI: 10.1109/ACC.2009.5160719.

[4] C. A. Monje, B. M. Vinagre, V. Feliu, and Y. Q. Chen, Fractional-Order Systems and Controls: Fundamentals and Applications, Springer, London, 2010. ISBN: 978-1-84996-334-3. DOI: 10.1007/978-1-84996-335-0.

[5] Y. Q. Chen and K. L. Moore, "Discretization schemes for fractional-order differentiators and integrators," IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 49, no. 3, pp. 363–367, 2002. ISSN: 1057-7122. DOI: 10.1109/81.989173.

ANALYSIS OF ONLINE AND OFFLINE SPEECH RECOGNITION APPROACHES FOR REAL-TIME SYSTEMS

Maksym Hrytsenko [1]

[1] *Czech Technical University in Prague, Faculty of Mechanical Engineering, Prague, Czech Republic*

Abstract: Speech recognition has become an important component of modern real-time human–computer interaction systems. This paper presents an experimental comparison of online and offline speech recognition approaches using multiple speech recognition systems representing different technological solutions. The evaluation focuses on recognition accuracy, processing latency, and robustness against different English accents. The experiments were performed using prepared voice command datasets and speakers with different language backgrounds. The obtained results demonstrate the trade-offs between cloud-based and local speech recognition systems and highlight their suitability for different real-time application scenarios.
Keywords: speech recognition, online systems, offline systems, latency, real-time recognition

1 Introduction

Speech recognition is currently used in many modern human–computer interaction systems, including voice assistants, accessibility tools, and real-time control applications. Recent progress in machine learning and neural networks has significantly improved the quality and usability of speech recognition systems. At the same time, both online and offline approaches remain widely used because they offer different advantages depending on the target application.

When developing systems intended for real-time interaction, selecting a suitable speech recognition method is an important task. Individual solutions differ in recognition accuracy, processing speed, hardware requirements, and dependency on internet connectivity. These differences become especially important in applications where low latency and stable response are required.

This work focuses on the experimental comparison of selected online and offline speech recognition systems. The paper describes the testing methodology, evaluation criteria, and experimental results obtained during testing with speakers using different English accents. The goal of the analysis is to compare the behavior of individual approaches in real-time scenarios and evaluate their suitability for practical real-time applications.

2 Speech Recognition Approaches

Automatic speech recognition systems can generally be divided into online and offline approaches depending on the method used for audio processing and computation. These approaches differ in latency, computational requirements, accuracy, and dependency on network connectivity [1, 2].

Modern speech recognition systems are commonly based on deep learning methods and neural architectures such as recurrent neural networks and transformer models [3, 4]. The choice of a specific approach depends primarily on the requirements of the target application, especially responsiveness, reliability, and computational constraints.

2.1 Online Speech Recognition

Online speech recognition systems process audio data using remote server infrastructure. In this approach, the recorded audio is transmitted through a network connection to cloud-based services where the speech signal is analyzed using large machine learning models [1].

The main advantage of online systems is their typically higher recognition accuracy. Since cloud platforms can utilize significantly more computational resources, they are capable of running larger and more complex neural models than local systems. Online services also benefit from continuous updates and improvements performed by service providers.

However, this approach introduces several limitations. The processing pipeline depends on stable internet connectivity and requires audio transmission to external servers, which may increase latency and raise privacy concerns. In real-time applications, additional network delay can negatively affect user interaction and overall system responsiveness.

Typical examples of online speech recognition systems include Google Speech Recognition and cloud-based Whisper API services[5, 6].

2.2 Offline Speech Recognition

Offline speech recognition systems perform all computations locally on the user device without requiring internet connectivity. Audio processing, feature extraction, and transcription generation are executed directly on local hardware [2].

The main advantage of offline systems is their independence from network infrastructure. This improves system autonomy and privacy because no audio data is transmitted externally. Offline approaches are therefore suitable for applications operating in environments with unstable internet access or increased privacy requirements.

Another important property of offline systems is the possibility of real-time streaming recognition. Audio data can be processed continuously during speech input, allowing the system to generate transcription results with minimal delay [5].

The main limitation of offline systems lies in hardware constraints. Local devices usually cannot provide computational resources comparable to cloud infrastructure, which may reduce recognition accuracy or require smaller optimized models. In addition, large neural models may significantly increase CPU and memory usage.

Examples of offline approaches include Vosk and locally executed Whisper models.

2.3 Selected Technologies

Several speech recognition systems were selected for experimental evaluation in order to represent different technological approaches and practical use cases.

Vosk was selected as a lightweight offline solution optimized for real-time speech recognition and streaming audio processing [7]. The system can operate on devices with limited computational resources and does not require internet connectivity. Due to its low latency and offline operation, Vosk represents a suitable approach for interactive real-time systems.

Whisper was selected as a modern deep learning based speech recognition model developed by OpenAI [6]. Unlike traditional systems, Whisper uses transformer architecture and end-to-end neural processing. The model is capable of achieving high recognition accuracy, especially for more complex speech patterns and noisy audio conditions.

Google Speech Recognition was selected as a representative of commercial cloud-based speech recognition services [5]. The system utilizes remote server infrastructure and large-scale machine learning models to provide highly accurate transcription results.

The selected technologies therefore represent three significantly different approaches to speech recognition: lightweight offline streaming recognition, modern large neural models, and commercial cloud-based services. This selection enables meaningful comparison of their properties in real-time speech recognition scenarios.

3 Experimental Methodology

This section describes the implementation of the testing environment, the experimental methodology, and the evaluation criteria used for the comparison of speech recognition systems.

3.1 Implementation

The testing environment was implemented in the Python programming language due to its extensive ecosystem of speech processing libraries and tools. The implementation included modules for audio capturing, preprocessing of recorded audio, communication with speech recognition libraries, and collection of transcription results.

The testing pipeline was designed to provide identical conditions for all evaluated systems. Audio input was captured using the same hardware configuration and processed through individual speech recognition

implementations. The obtained transcriptions were then stored and compared with the expected phrases for further evaluation.

Several speech recognition libraries and APIs representing different approaches were integrated into the testing environment. The implementation included both online and offline systems in order to compare their behavior under the same experimental conditions.

Modern speech recognition systems are commonly based on neural or hybrid architectures combining statistical methods with deep learning approaches [2, 3, 4]. Real-time speech recognition systems process audio continuously during speech input, while batch approaches process the complete audio recording after the recording has finished [1].

3.2 Testing Methodology

To ensure objective comparison between individual speech recognition systems, a predefined testing dataset consisting of voice commands and common English phrases was prepared. The selected phrases represented commands of different lengths and complexity levels, ranging from short control commands to longer natural language sentences.

The testing dataset included the following phrases:

- open browser
- close window
- scroll down
- play music
- open file
- open the browser and search for information
- close the window after finishing the task
- scroll down to the bottom of the page
- artificial intelligence system for voice control

The selected phrases were intentionally designed to include both short and long expressions. Short commands were used to evaluate recognition speed and responsiveness in real-time interaction scenarios, while longer phrases were intended to test transcription stability and sentence-level recognition accuracy.

The experiments were performed on a system equipped with an AMD Ryzen 5 4600H processor, 16 GB RAM, and an NVIDIA GeForce GTX 1650 graphics card. Voice recordings were captured using an iPhone microphone. The testing process was conducted in a controlled environment using identical hardware configuration and microphone settings for all evaluated systems. Each phrase was repeated multiple times for every tested speech recognition approach.

Particular attention was given to accent variability and pronunciation differences. The experiments involved three speakers with different language backgrounds:

- a Ukrainian speaker communicating in English,
- a Czech speaker communicating in English,
- a native English speaker from Ireland.

The inclusion of multiple accents was intended to simulate more realistic usage conditions and evaluate the robustness of individual systems against pronunciation variability.

3.3 Evaluation Criteria

The evaluated systems were compared using multiple criteria relevant for real-time speech recognition applications. The evaluation methodology focused primarily on recognition accuracy and processing speed.

Recognition accuracy was defined as a discrete value according to the following conditions:

$$Accuracy = \begin{cases} 1.0, & \text{if the phrase was recognized correctly} \\ 0.5, & \text{if the phrase was recognized partially correctly} \\ 0.0, & \text{if the phrase was recognized incorrectly} \end{cases}$$

This approach allowed simplified comparison between systems while still reflecting partially successful transcriptions.

Processing speed was normalized according to the measured response latency using the following equation:

$$Speed = \frac{1}{1+t}$$

where t represents the speech processing time (latency) measured in seconds. This normalization maps the resulting value into the interval $(0,1)$, where lower latency corresponds to a higher score coefficient.

To obtain a final evaluation value for each tested phrase, a weighted scoring method combining recognition accuracy and processing speed was introduced:

$$Score = 0.7 \cdot Accuracy + 0.3 \cdot Speed$$

The weighted evaluation prioritized recognition correctness while still considering system responsiveness, which is essential for real-time applications [5, 1].

4 Results

This section presents the results obtained during experimental evaluation of the selected speech recognition systems. The analysis focused primarily on recognition accuracy, response latency, and the influence of speaker accent on the final performance.

Table 1 shows an example of detailed testing results obtained using the Vosk speech recognition system for the Ukrainian speaker communicating in English. The table includes the original spoken phrase, recognized transcription, processing latency, and the final weighted score calculated according to the evaluation methodology described in the previous section.

Tab. 1: Detailed testing results for Vosk (Ukrainian speaker)

No.	Spoken Phrase	Recognized Text	Time (s)	Score
1	open browser	browser	0.91	0.51
2	close window	gold swindle	0.91	0.16
3	scroll down	scroll down	0.97	0.85
4	play music	play music	0.88	0.86
5	open file	open file	1.48	0.82
6	open the browser and search for information	open the browser and of the for information	2.45	0.44
7	close the window after finishing the task	cause the window after finishing a did ask	2.99	0.08
8	scroll down to the bottom of the page	scroll down to the bottom of the beach	2.48	0.44
9	artificial intelligence system for voice control	artificial intelligence system for voice control	3.42	0.77

The detailed results demonstrate that short commands were generally recognized with higher accuracy and lower latency compared to longer natural language phrases. More complex sentences frequently introduced transcription errors or partially recognized outputs, especially for offline systems.

To provide a clearer comparison between evaluated approaches, average scores for all tested speakers were calculated. The results are summarized in Table 2.

Tab. 2: Average evaluation results for individual speakers

System	Ukrainian Speaker	Czech Speaker	Irish Speaker
Vosk	0.55	0.53	0.47
Whisper	0.84	0.82	0.76
Google Speech Recognition	0.72	0.70	0.64

The obtained results indicate that online systems generally achieved higher recognition accuracy than offline approaches. However, offline systems demonstrated more stable latency and independence from network conditions.

The experiments also showed measurable influence of speaker accent on recognition performance. The Irish speaker achieved slightly lower average scores compared to the Ukrainian and Czech speakers. This behavior is likely related to accent variability and pronunciation patterns that are less represented in commonly used training datasets.

5 Discussion

The experimental results revealed noticeable differences between online and offline speech recognition approaches.

Online systems achieved higher recognition accuracy, especially for longer and more complex phrases. This behavior is mainly related to the use of large cloud-based neural models trained on extensive speech datasets. However, the experiments also showed that online approaches introduce additional latency caused by network communication, which may negatively affect responsiveness in real-time applications.

Offline systems provided more stable response times and full independence from internet connectivity. Lightweight solutions such as Vosk were capable of continuous real-time speech processing even on limited hardware configurations. At the same time, offline systems showed lower robustness when processing longer phrases or less common pronunciation patterns.

The experiments also demonstrated that speaker accent has a measurable influence on recognition quality. While all evaluated systems handled moderate pronunciation differences relatively well, stronger regional accents reduced transcription accuracy. This effect was most visible for the Irish speaker, whose pronunciation differs more significantly from the speech patterns commonly represented in training datasets.

From a practical perspective, the results suggest that online systems are more suitable for applications prioritizing transcription accuracy, while offline systems remain advantageous for privacy-sensitive or low-latency real-time applications.

The experiments were performed on a relatively limited dataset and a small number of speakers. In addition, the testing process was conducted in a controlled environment using identical hardware configuration and microphone settings. Therefore, the obtained results should be interpreted primarily as a comparative experimental analysis rather than a universal benchmark for all speech recognition scenarios.

6 Conclusion

This paper presented an experimental comparison of online and offline speech recognition approaches for real-time applications.

Several speech recognition systems representing different technological approaches were evaluated using a unified testing methodology. The analysis included recognition accuracy, processing latency, and robustness against accent variability.

The obtained results showed that online systems achieved higher transcription accuracy, while offline approaches provided lower dependency on external infrastructure and more predictable real-time behavior.

Future work may include evaluation on larger multilingual datasets and comparison under real-time noisy environments.

Literature

[1] C. Deuerlein, M. Langer, J. Sener, and J. Franke. Human–robot interaction using cloud-based speech recognition systems. *Procedia CIRP*, 97:541–546, 2020.

[2] M. Gales and S. Young. The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3):195–304, 2017.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[4] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[5] N. Anggraini, A. Kurniawan, L. K. Wardhani, and N. Hakiem. Speech recognition application for the speech impaired using android-based google cloud speech api. *TELKOMNIKA*, 16(6):2733–2739, 2018.

[6] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2104.01384*, 2021.

[7] Alpha Cephei. Vosk speech recognition toolkit. https://alphacephei.com/vosk/, 2024.

COMPARATIVE ANALYSIS OF RETRIEVAL-AUGMENTED GENERATION PIPELINES FOR STRUCTURED PROCEDURAL DOCUMENTATION

Volodymyr Dudavskyi [1]

[1] *Czech Technical University in Prague, Faculty of Mechanical Engineering, Prague, Czech Republic*

Abstract: Technical documentation for industrial machines often contains structured procedures, section titles, numbered steps, warnings, configuration values, and alarm recovery instructions. This paper presents a comparative evaluation of four Retrieval-Augmented Generation pipelines for question answering over structured procedural documentation. The compared approaches are Flat Vector RAG, Hierarchical RAG, Reranked RAG, and a local GraphRAG-style pipeline. All approaches use the same documentation dataset and the same final answer generation prompt; the main difference is the retrieval strategy used before generation. The evaluation uses ten realistic operator-style questions in English, Czech, and mixed Czech-English. The questions include exact machine identifiers, machine aliases, missing machine identifiers, alarm codes, exact-value requests, and multi-section procedural requests. The results show that Hierarchical RAG achieved the strongest retrieval performance on this dataset, with correct expected source retrieval for all ten questions. Flat Vector RAG and Reranked RAG were faster but less robust for Czech wording and missing machine identifiers. The local GraphRAG-style pipeline often retrieved related context, but its top-ranked source selection was weak for exact procedural lookup. The result is interpreted as a limitation of the simplified local graph retrieval implementation and of graph-neighborhood retrieval for narrow procedural questions, not as a general evaluation of the Microsoft GraphRAG framework.
Keywords: Retrieval-Augmented Generation, technical documentation, procedural documentation, vector search, reranking, hierarchical retrieval, graph-based retrieval

1 Introduction

Technical service and operating documentation is an important part of industrial machine operation. It usually contains instructions for machine setup, normal operation, calibration, alarm recovery, maintenance, and final inspection. In many cases this documentation is stored as PDF, Markdown, or other text-based files and is organized into manuals, sections, and numbered procedures. Although such structure is suitable for human reading, it can still be inefficient when an operator needs to quickly find one exact instruction during machine operation.

Large Language Models (LLMs) can produce fluent natural language answers, but their direct use is not sufficient for technical documentation. The model may answer from its internal knowledge instead of the real machine manual. In technical applications this can lead to unsupported steps, incorrect values, or missing safety instructions. For this reason, the answer must be grounded in the relevant source section of the documentation.

Retrieval-Augmented Generation (RAG) addresses this problem by combining a language model with an external knowledge base. In a typical RAG system, documents are first divided into smaller text units, embedded into vectors, stored in a retrieval index, and later retrieved according to the user question. The retrieved context is then passed to the language model, which generates the final answer based on the selected source material [1, 2]. This makes RAG suitable for machine manuals and service instructions because the system can answer from a selected document context instead of relying only on the model parameters.

The aim of this work is to design and evaluate several RAG pipelines on structured procedural documentation. The evaluated approaches are Flat Vector RAG, Hierarchical RAG, Reranked RAG, and a local GraphRAG-style pipeline. The comparison focuses on how the retrieval strategy influences the final generated answer when all

approaches use the same source documents and the same answer generation step. The tested questions are designed to be realistic and include exact machine identifiers, Czech wording, mixed language questions, machine names written as words, and ambiguous procedural requests.

The main contribution of this work is a unified experimental comparison of RAG pipelines under the same conditions. The report describes the repository architecture, data preparation process, selected approaches, evaluation criteria, and observed retrieval behavior. Special attention is given to realistic operator questions, where the machine identifier may be missing or expressed as a Czech or English alias. The results are then discussed with respect to retrieval precision, answer grounding, language robustness, and the suitability of graph-based retrieval for exact procedural lookup.

2 Background

2.1 Retrieval-Augmented Generation

Retrieval-Augmented Generation is a method that extends a generative language model with an external retrieval component. Lewis et al. introduced RAG as a model that combines parametric memory in a pre-trained sequence-to-sequence model with non-parametric memory represented by a dense vector index [1]. The retrieval component selects relevant passages from an external corpus and the generator uses them as context for answer generation.

In recent work, RAG is commonly divided into several families such as naive RAG, advanced RAG, and modular RAG [2]. Naive RAG usually consists of document chunking, vector search, top-k retrieval, and answer generation. Advanced and modular variants add components such as query rewriting, reranking, metadata filtering, routing, graph-based retrieval, or multi-step retrieval. These extensions are important because simple top-k vector retrieval may return incomplete or noisy context when the document collection contains many similar sections.

For technical documentation, the retrieval component is especially important. The system should not only find text that is semantically related to the question, but should also select the correct machine, manual, and section. The final answer should preserve exact procedure steps, alarm codes, values, component names, and source references. This requirement makes structured procedural documentation a useful test case for comparing different RAG architectures.

2.2 Dense Retrieval and Vector Search

Dense retrieval represents text as numerical vectors and compares them in a continuous embedding space. Dense Passage Retrieval showed that open-domain question answering can be implemented using dense representations learned by a dual-encoder framework and that dense retrieval can outperform traditional sparse retrieval in several settings [3]. Sentence-BERT introduced sentence embeddings that can be compared using cosine similarity and are suitable for semantic similarity search [4].

In this project, dense retrieval is used to represent each procedural section or chunk as an embedding vector. A user question is embedded in the same vector space and compared with stored chunks. The most similar chunks are retrieved and used as context for the final answer. For efficient similarity search, FAISS-like vector indexing is suitable because it was designed for fast nearest-neighbor search over large vector collections [5].

The advantage of dense retrieval is its ability to match semantically similar text even when the question and document do not use exactly the same words. This is useful for operator-style questions, where the user may describe a symptom rather than a manual title. The limitation is that vector similarity alone may confuse sections with similar vocabulary, especially in structured manuals where many sections contain words such as *HMI*, *alarm*, *reset*, *check*, or *recipe*.

2.3 Reranking

Reranking is a common improvement over basic vector retrieval. The system first retrieves a larger set of candidate chunks using a fast retriever and then applies a second model to rank these candidates more accurately. Nogueira and Cho showed that a BERT-based model can be used for query-based passage reranking and can improve passage ranking performance on benchmark datasets [6].

In a RAG pipeline, reranking is useful when the first retrieval stage returns several partially relevant chunks. Instead of passing all candidates to the generator, the reranker selects the most relevant subset. This can reduce context noise and improve answer grounding. However, reranking cannot recover a correct chunk if it was not included in the candidate set retrieved by the first-stage retriever. Therefore, candidate size and first-stage retrieval quality remain important.

2.4 Hierarchical Retrieval

Hierarchical retrieval uses the natural structure of a document collection. Instead of searching all chunks globally, the system first selects a higher-level unit such as machine, manual, chapter, or section, and only then retrieves detailed text from this selected part. This approach is related to recent hierarchical retrieval methods such as RAPTOR, where documents are recursively clustered and summarized into a tree structure that can be used during retrieval [7].

For structured procedural documentation, hierarchy is already present in the data. Machine folders contain manuals, manuals contain sections, and sections contain numbered procedure steps. This structure can be used directly by a routing component. For example, the system may first identify the machine as `02-110`, then select the manual about label verification, and finally retrieve the section about resetting a barcode alarm. The advantage is a reduced search space and better control over the selected source. The weakness is that a routing error at the beginning can be fatal because the system may retrieve only from the wrong manual or section.

2.5 GraphRAG

GraphRAG extends RAG by building a graph representation of the document collection. Instead of treating chunks as independent text blocks, the system extracts entities, relations, and higher-level communities. Edge et al. proposed GraphRAG for query-focused summarization, where a graph is created from private text corpora and community summaries are used to answer both local and global questions [8].

The motivation of GraphRAG is strongest when the question requires relationships between entities, information distributed across many documents, or a higher-level summary of a corpus. In contrast, the present work evaluates mostly local procedural questions, where the desired answer is usually contained in one exact section. This distinction is important for interpretation of the results. The implemented method in this repository is a local GraphRAG-style prototype: it builds an entity graph and community summaries, but it does not reproduce the full Microsoft GraphRAG pipeline. Therefore, its results should be interpreted as the performance of this local graph-based retrieval implementation, not as a general benchmark of the original GraphRAG system.

3 Compared RAG Pipelines

Four RAG pipelines are selected for comparison. All of them use the same documentation dataset and the same final answer generator. The difference is the retrieval and context selection method.

3.1 Flat Vector RAG

Flat Vector RAG is used as the baseline approach. In this method, all sections or chunks from all manuals are embedded and stored in one vector index. When a question is asked, the query is embedded and the top-k most similar chunks are retrieved from the complete collection.

The pipeline can be summarized as follows:

$$\text{Question} \rightarrow \text{query embedding} \rightarrow \text{global vector search} \rightarrow \text{top-k chunks} \rightarrow \text{answer generation}.$$

This approach is simple and easy to implement. It is also a useful baseline because many practical RAG systems start from this architecture. The main weakness is that all chunks compete globally. If two machines contain similar procedures, the system may retrieve a section from the wrong machine. The answer generator may also receive mixed context from different manuals.

3.2 Hierarchical RAG

Hierarchical RAG uses document structure during retrieval. The system first routes the question to a machine and then to a manual section. After the section is selected, the corresponding chunk is passed to the answer generator.

The pipeline can be summarized as follows:

$$\text{Question} \rightarrow \text{machine routing} \rightarrow \text{manual/section routing} \rightarrow \text{selected section} \rightarrow \text{answer generation}.$$

This method is suitable for structured procedural documentation because the documents are already organized by machine, manual, and section. It is expected to perform well when the question contains a machine ID or a clear machine alias. It may also handle Czech aliases such as *stodesítka* or *sto dvanáctka* if these aliases are included in the metadata catalog. The main weakness is dependence on the router. If the selected section is wrong, the system has limited ability to recover.

3.3 Reranked RAG

Reranked RAG adds a reranking stage after the first vector retrieval. The first-stage retriever returns a larger candidate set, for example top-n chunks. Then a reranker scores each query–chunk pair and selects the final top-k chunks for answer generation.

The pipeline can be summarized as follows:

Question → vector candidate retrieval → reranking → final top-k chunks → answer generation.

This method is expected to improve over Flat Vector RAG when the vector index retrieves several similar chunks. It can reduce irrelevant context and help the generator focus on the most relevant procedure. Its limitation is that the correct chunk must already be present in the candidate set. If the first-stage vector retrieval misses the correct section, the reranker cannot select it.

3.4 Local GraphRAG-style RAG

The implemented graph-based method builds an additional graph index from the documentation. During indexing, entities and relations are extracted from chunks. The graph connects machines, manuals, sections, technical components, alarm codes, recipe names, and other extracted entities. Community-level information is also summarized and stored for retrieval.

The pipeline can be summarized as follows:

Chunks → entity and relation extraction → graph construction → community summaries → graph retrieval → answer generation.

During question answering, the system matches query terms to graph nodes, expands the graph neighborhood, ranks the related chunks, adds community summaries, and passes the resulting context to the same answer generator. This design is useful for examining whether graph neighborhoods can support procedural question answering. However, it is a simplified local GraphRAG-style implementation. It relies on rule-based query entity extraction and token-overlap ranking at query time. It therefore should not be presented as a complete reproduction of the Microsoft GraphRAG framework. This methodological limitation is considered explicitly in the discussion.

4 Documentation Dataset

4.1 Dataset Structure

The experimental dataset is organized as structured procedural documentation. The root folder contains machine folders. Each machine folder contains several Markdown instruction manuals. Each manual begins with metadata and then contains numbered sections with step-by-step instructions.

The current dataset contains two machine groups:

- `02-110`: Carton Packaging Cell CP-110,
- `02-112`: Shrink Wrap Packaging Cell SW-112.

Each machine contains several instruction manuals. For example, machine `02-110` contains manuals related to carton packaging, label printer setup, carton erector calibration, tape sealer changeover, and PLC/HMI alarm reset. Machine `02-112` contains manuals related to tray packaging, film unwinder changeover, shrink tunnel temperature setup, vision inspection, and PLC alarm recovery.

Each manual contains three sections. A section usually corresponds to one complete procedure, such as a setup procedure, alarm recovery procedure, calibration procedure, or final check. Since the sections are short and procedural, the default chunking strategy creates one chunk per section. This preserves the complete procedure and avoids splitting warnings or expected results away from the related steps.

4.2 Metadata

Metadata is used to preserve the document hierarchy and support routing. Each chunk contains at least the following fields:

- `machine_id`,

- machine_name,
- instruction_id,
- instruction_title,
- section_title,
- chunk_id,
- source_path,
- steps_count.

For Hierarchical RAG, machine aliases are also important. The aliases allow the system to map operator wording to the correct machine. Examples include *stodesítka*, *sto desítka*, and *one hundred ten* for machine 02-110, and *sto dvanáctka* or *one hundred twelve* for machine 02-112.

5 System Architecture

5.1 Common Processing Pipeline

All approaches use the same initial processing pipeline. First, Markdown files are loaded from the documentation folder. The YAML-like metadata is parsed and the document body is extracted. Then the document is split into sections. Each section is converted into one chunk with hierarchical metadata. The chunks are saved as JSON and embedded using a local embedding model. The embeddings are stored in a FAISS index.

The common processing pipeline is:

$$\text{Documentation} \rightarrow \text{loading} \rightarrow \text{section parsing} \rightarrow \text{chunking} \rightarrow \text{embedding} \rightarrow \text{vector index}.$$

The advantage of this design is that the four compared methods use the same input data. Therefore, differences in results can be attributed mainly to the retrieval strategy and not to different preprocessing.

5.2 Local and Configurable Language Model Interface

The repository is designed to support a local language model interface. This follows the requirement that technical documentation may contain internal information and should not necessarily be sent to external services. A local model can be executed through an Ollama-compatible interface. For practical testing, the same interface can also be configured to use an OpenAI-compatible provider when local inference is unavailable.

The final answer generator uses the same prompt for all approaches. The prompt instructs the model to answer only from the provided context and to avoid inventing steps, values, alarms, or component names. This is important for fair comparison because the generation stage should not be different between retrieval methods.

5.3 Repository Modules

The repository is divided into modules according to responsibility:

- cli: command-line scripts for building indexes and running individual approaches,
- config: project paths, model settings, and prompt templates,
- data: structured procedural documentation and test questions,
- src/ingestion: Markdown loading, section parsing, chunk creation, and metadata catalogs,
- src/embeddings: local embedding model wrapper,
- src/vector_store: FAISS index and chunk storage,
- src/rag: implementations of the four RAG approaches,
- src/router: machine, manual, and section routing,
- src/graphrag: graph extraction, graph construction, community processing, and graph retrieval,

- `src/reranker`: reranker model wrapper,
- `src/evaluation`: result logging and manual evaluation helpers.

Each method is tested separately from the command line. This makes debugging easier and prevents one long test run from hiding method-specific problems. The produced outputs are saved as JSON files and later used for manual analysis.

6 Experimental Methodology

6.1 Testing Strategy

The experiment evaluates the four RAG approaches on the same set of ten questions. The methods are not executed as one combined automatic comparison. Instead, each method is called separately with one question at a time. This allows inspection of terminal output, retrieved sources, routing traces, graph context, and final generated answers.

The typical testing command sequence for one question is:

- run Flat Vector RAG,
- run Hierarchical RAG,
- run Reranked RAG,
- run the local GraphRAG-style pipeline.

Each run stores the final answer, retrieved sources, method name, question, and available metadata. The saved outputs are then compared manually according to the criteria below.

6.2 Evaluation Questions

The question set is designed to be realistic and not too easy. It includes English, Czech, and mixed Czech-English questions. Some questions contain exact machine identifiers, while others use aliases or do not mention the machine directly. This is important because real operators often do not ask questions in the same wording as the manual.

Table 1 shows the evaluation questions and expected sources.

Tab. 1: Evaluation questions

ID	Question	Expected source
Q01	How do I reset the label verification alarm on machine 02-110?	02-110, INS-02-110-05, Section 2 – Resetting Label Verification Alarm
Q02	Na stodesítce nejde přečíst čárový kód na štítku. Co mám udělat?	02-110, INS-02-110-05, Section 2 – Resetting Label Verification Alarm
Q03	On one hundred ten, the tape is stuck in the sealer. How do I recover it?	02-110, INS-02-110-04, Section 3 – Tape Jam Recovery
Q04	Co mám zkontrolovat před potvrzením hotového kartonu?	02-110, INS-02-110-01, Section 3 – Final Check After Carton Packaging
Q05	How do I calibrate the vacuum arm for the carton erector on 02-110?	02-110, INS-02-110-03, Section 1 – Calibration for Type A Carton
Q06	Fólie se přetrhla na sto dvanáctce, jak obnovit provoz?	02-112, INS-02-112-05, Section 1 – Film Break Alarm Recovery
Q07	The shrink tunnel is too hot and shows SHT-09. What is the safe restart procedure?	02-112, INS-02-112-05, Section 2 – Shrink Tunnel Overtemperature Alarm
Q08	Jak nastavit fólii pro deep tray na 02-112? Potřebuju šířku, tah a čas čelisti.	02-112, INS-02-112-02, Section 2 – Film Setup for Type D Deep Tray
Q09	How do I check that the camera rejects packages with missing label or damaged film?	02-112, INS-02-112-04, Section 3 – Reject Verification Test
Q10	Na shrink lince chci připravit deep tray a potom zkontrolovat balení po smrštění. Jaké jsou hlavní kroky?	02-112, INS-02-112-01, Sections 2 and 3

The questions are divided into several difficulty categories. Q01 and Q06 are direct procedural questions. Q02, Q03, and Q06 test machine aliases and language variation. Q04 and Q09 do not directly mention the machine and therefore test whether the system can infer the correct context from the procedure. Q07 and Q08 require exact alarm or value handling. Q05 and Q10 are more difficult because they can require more than one relevant section.

6.3 Evaluation Criteria

The evaluation will be performed using five criteria. Each criterion is scored from 0 to 2 points.

Tab. 2: Evaluation criteria

Criterion	Description	Score
Source retrieval	Whether the expected section or acceptable alternative section was retrieved	0–2
Answer correctness	Whether the generated answer follows the source procedure correctly	0–2
Completeness	Whether important steps, values, and warnings are included	0–2
Grounding	Whether the answer is supported by the retrieved context	0–2
Hallucination control	Whether the answer avoids unsupported steps, values, or components	0–2

The maximum score for one question is therefore 10 points. Since ten questions are used, the maximum score for one approach is 100 points. The final score is calculated as:

$$Score_{method} = \sum_{q=1}^{10} \left(S_{retrieval} + S_{correctness} + S_{complete} + S_{grounding} + S_{hallucination} \right)$$

where each partial score is assigned manually after inspecting the answer and retrieved sources.

6.4 Expected Weaknesses of the Approaches

Before running the experiment, the expected weaknesses of the methods can be identified from their architecture.

Flat Vector RAG is expected to be simple and robust for direct questions, but it may retrieve chunks from the wrong machine when the machine is not specified. It may also mix context from multiple similar procedures.

Hierarchical RAG is expected to work well when the machine and section can be routed correctly. Its main weakness is that a routing error is difficult to recover from. Ambiguous questions may require clarification rather than direct answer generation.

Reranked RAG is expected to reduce irrelevant context compared with Flat Vector RAG. Its main limitation is dependence on the candidate set from the first retrieval stage. If the correct chunk is not retrieved as a candidate, the reranker cannot select it.

The local GraphRAG-style pipeline is expected to help with entity-based and broader questions. However, it has higher indexing cost and may lose exact procedural detail when community summaries or broad graph neighborhoods are used instead of exact sections. Since the query-time implementation uses rule-based entity matching and simple graph-neighborhood ranking, it is expected to be weaker for Czech aliases and precise step-by-step questions than methods that explicitly route through the document hierarchy.

7 Results

The experiment was executed on the ten evaluation questions from Table 1. Each question was submitted to all four approaches. The runtime reported in this section includes the complete query-time method call, including retrieval, routing or reranking when used, and final answer generation. It does not include offline vector index construction or offline graph construction.

Table 3 summarizes the retrieval behavior. Top-1 accuracy means that the first returned source was the expected section or an accepted alternative. Recall means that the expected section appeared anywhere in the

returned context. The column “insufficient answers” counts cases where the generated answer explicitly stated that the provided documentation did not contain enough information.

Tab. 3: Overall retrieval and answer behavior

Method	Top-1 correct source	Correct source anywhere	Insufficient answers	Average runt.[s]
Flat Vector RAG	6/10	7/10	2/10	14.67
Hierarchical RAG	10/10	10/10	2/10	31.47
Reranked RAG	5/10	7/10	2/10	14.30
Local GraphRAG-style RAG	1/10	7/10	3/10	14.90

The strongest retrieval result was obtained by Hierarchical RAG. It selected the expected source or an accepted alternative for all ten questions. This is mainly because the method first routes the question to a machine and then to a manual section, instead of searching all chunks globally. For example, in Q02 the question does not contain the literal identifier `02-110`; it uses the Czech alias *stodesítce*. The hierarchical router mapped this alias to machine `02-110` and then selected the label verification alarm section. In Q04, the question also does not contain a machine identifier, but the term *karton* indicates the carton packaging cell. The router therefore selected machine `02-110` and the section *Final Check After Carton Packaging*.

Flat Vector RAG performed well on direct English questions and on questions with strong technical keywords. It returned the correct top source for Q01, Q03, Q05, Q07, Q09, and Q10. However, it failed on Czech and ambiguous questions. In Q02 it retrieved vision inspection sections from machine `02-112` instead of the label verification alarm section from `02-110`. In Q04 it selected a shrink wrapping final check from machine `02-112`, which is semantically similar to the phrase “final check”, but belongs to the wrong machine and procedure.

Reranked RAG improved the ranking for some direct questions, but it did not solve the main weakness of the first-stage vector retrieval. It achieved correct top-1 retrieval for five questions and included the correct source somewhere in the context for seven questions. The results show the expected limitation of reranking: if the correct section is missing from the candidate set or is weakly represented because of language mismatch, the reranker cannot reliably recover it.

The local GraphRAG-style pipeline showed the weakest top-1 precision. It selected the correct top source only for Q09. However, the correct source was present somewhere in the retrieved context for seven questions. This means that the graph index often found related context, but the query-time ranking did not place the exact procedural section first. For example, in Q01 the correct label verification section was retrieved, but a more general PLC/HMI restart section was ranked above it. In Q07 the correct shrink tunnel overtemperature section appeared in the returned context, but the top source was a packaging section. These results suggest a ranking and context-selection problem rather than a complete absence of useful graph structure.

Tab. 4: Top-1 retrieval correctness by question

Question	Flat Vector RAG	Hierarchical RAG	Reranked RAG	Local GraphRAG-style RAG
Q01	correct	correct	correct	incorrect
Q02	incorrect	correct	incorrect	incorrect
Q03	correct	correct	correct	incorrect
Q04	incorrect	correct	incorrect	incorrect
Q05	correct	correct	correct	incorrect
Q06	incorrect	correct	incorrect	incorrect
Q07	correct	correct	correct	incorrect
Q08	incorrect	correct	incorrect	incorrect
Q09	correct	correct	correct	correct
Q10	correct	correct	incorrect	incorrect

8 Discussion

The results confirm that the best RAG architecture depends on the structure of the documentation and on the type of user question. The documentation used in this work is highly structured: it contains machines, manuals, sections, and short procedural chunks. For this type of corpus, explicitly using the hierarchy is more effective than

relying only on global semantic similarity. Hierarchical RAG was therefore the most reliable retrieval method in the experiment.

The success of Hierarchical RAG on Q02 and Q04 is especially important. In both cases the machine identifier is not written as a number. In Q02 the Czech expression *stodesítce* is interpreted as an alias for machine `02-110`. In Q04 the machine is inferred from the procedure domain: *hotový karton* indicates the carton packaging cell. This behavior is not possible with pure vector search unless the embedding model and the indexed text represent the alias or domain context very clearly. The result shows the value of a separate routing step for industrial documentation.

At the same time, Hierarchical RAG was not perfect as a complete question-answering system. In Q06 it retrieved the correct source, but the final answer still stated that the documentation did not contain enough information. In Q10 the selected source was an accepted alternative, but the question asked for both preparation of a deep tray and final checking after shrink wrapping. A single selected section was therefore not sufficient for a complete answer. These cases show that correct retrieval does not automatically guarantee correct generation. They also show that hierarchical retrieval should support multi-section routing when the question contains several procedural intents.

Flat Vector RAG and Reranked RAG have the advantage of simpler implementation and lower query-time latency. They are useful baselines and work well for exact English questions with strong keywords, such as alarm codes or component names. Their weakness is robustness. When the question is in Czech, when a machine alias is used, or when the machine is omitted, the methods can select semantically similar but procedurally wrong sections. Reranking improves ordering only inside the candidate set produced by the first retrieval step. It cannot select a correct section that was not retrieved as a candidate.

The interpretation of the graph-based result requires care. The method implemented here should be described as a local GraphRAG-style prototype, not as a full reproduction of Microsoft GraphRAG. The graph index contains useful information: machines, instructions, chunks, technical entities, alarm codes, recipe names, and communities. However, query-time retrieval uses rule-based entity matching, graph-neighborhood expansion, and token-overlap ranking. This is much simpler than a full query-focused graph retrieval pipeline. As a result, the system often retrieved related neighborhoods but did not rank the exact procedural section first.

Several graph artifacts explain the observed behavior. Some communities are useful, for example a community that connects carton packaging final check with label verification on machine `02-110`. Other communities are too broad for exact procedural lookup. For instance, alarm-related communities can connect different machines and different alarm procedures because they share general entities such as *PLC*, *HMI*, *Reset Alarm*, or *restart*. Generic entities such as *check*, *press*, *confirm*, *machine*, and *section* can also create noisy graph bridges. This explains why the local GraphRAG-style method often had good recall but poor top-1 precision.

This result does not contradict the motivation of GraphRAG in the literature. The original GraphRAG work focuses on query-focused summarization and on moving from local to global understanding over a corpus [8]. Such methods are valuable when a question asks for themes, relations, or information distributed across many documents. The present experiment is different: most questions require one exact procedural section and ordered steps. In this setting, the document hierarchy already provides a strong retrieval structure, while broad graph neighborhoods and community summaries may introduce noise.

A stronger graph-based system for this task would need several improvements. First, the query should be processed by an LLM-based entity and intent linker that can map Czech expressions and machine aliases to graph nodes. Second, graph expansion should be limited for procedural lookup, for example to one hop unless the question explicitly asks for a broader relation. Third, generic entities should be filtered or heavily down-weighted. Fourth, graph-based scores should be combined with metadata and vector similarity, with strong boosts for machine matches, alarm code matches, and section title matches. Finally, exact chunks should be placed before community summaries in the answer context, because procedural questions depend on exact steps and values.

9 Conclusion

This paper compared four retrieval strategies for question answering over structured procedural documentation. The experiment used the same documentation dataset, the same chunk structure, and the same final answer generator for all methods. The main variable was the retrieval strategy.

Hierarchical RAG achieved the best retrieval performance in this experiment. It correctly selected the expected source or an accepted alternative for all ten questions. The result shows that structured routing through machine, manual, and section metadata is highly suitable for procedural documentation. This is especially useful when operators use Czech wording, machine aliases, or incomplete questions without explicit machine identifiers.

Flat Vector RAG and Reranked RAG were simpler and faster, but less robust for multilingual and ambiguous questions. Reranked RAG improved some direct cases, but remained dependent on the initial vector candidate

set. The local GraphRAG-style pipeline retrieved useful related context in several cases, but its top-ranked source precision was weak. This should be interpreted as a limitation of the simplified local graph retrieval implementation and of broad graph-neighborhood retrieval for exact procedural lookup, not as a general negative result for GraphRAG as proposed in the literature.

The main practical conclusion is that, for structured procedural documentation, the most effective approach is not necessarily the most complex one. The best retrieval method is the one that matches the structure of the source material and the type of user question. For the tested dataset, hierarchy-aware retrieval was the most reliable solution. Future work should improve multi-section routing for complex procedural questions and develop a stronger graph-based retriever with semantic query-to-entity linking and weighted graph ranking.

Literature

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

[2] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[3] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781, 2020.

[4] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, 2019.

[5] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

[6] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

[7] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In *Proceedings of the International Conference on Learning Representations*, 2024.

[8] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

DETECTION OF AI-GENERATED IMAGES: FREQUENCY SPECTRUM ANALYSIS

Jan Rollo[1], Vladimir Hlavac[1]

1Department of Instrumentation and Control, Czech Technical University in Prague, Technicka 4, Prague 6, jan.rollo@fs.cvut.cz

Abstract: With the massive rise of synthetic content generation using Deep Neural Networks (DNNs) [2], distinguishing real photographs from artificially created ones is becoming increasingly difficult. This paper explores the feasibility of detecting AI-generated images using frequency spectrum analysis, specifically the Radially Averaged Power Spectral Density (RAPSD) method. The aim of the experiment was to verify whether spectral data alone contains sufficient information for successful classification without the need for visual content analysis. A 1D Convolutional Neural Network (ResNet1D) model was designed and trained using log-transformed RAPSD data and linear regression residuals (Square Error) as inputs. The results suggest that while frequency analysis reveals specific artifacts of generative models, robust detection requires combining this approach with other methods and utilizing high-resolution data.

Keywords: Deepfake detection, frequency analysis, RAPSD, FFT, Convolutional Neural Networks, ResNet1D.

1. Introduction

With the advent of advanced generative models (e.g., Midjourney, Stable Diffusion), the visual boundary between reality and synthetic content is blurring. This reality poses significant risks regarding the spread of disinformation and the verification of digital media authenticity. Traditional detection methods often focus on visual artifacts in the spatial (pixel) domain. This work, however, focuses on an alternative approach—analyzing the image in the frequency domain.

The fundamental hypothesis is based on the physics of imaging. Real photographs of natural scenes tend to exhibit a power spectrum that falls off approximately linearly on a log-log scale, a phenomenon known as the "1/f law." The premise of this study is that artificial neural networks, while generating visually convincing content, fail to faithfully reproduce this natural spectral characteristic. When generating an image from latent space (e.g., through up-sampling processes), periodic patterns or high-frequency noise often emerge, leaving an invisible "fingerprint" on the image.

The goal of this work was to ignore the semantic content of the image and focus purely on the mathematical analysis of frequency characteristics to distinguish between real and synthetic photographs.

2. Methodology

Instead of training computationally expensive 2D Convolutional Neural Networks (CNNs) [3] on full-resolution images, a method of drastic data reduction was chosen. The processing pipeline consisted of the following steps:

- FFT [1] (Fast Fourier Transform):
  Converting the image from the spatial domain to the frequency domain.
- RAPSD (Radially Averaged Power Spectral Density):
  A 1D profile was created from the 2D image spectrum by averaging energy values along concentric circles from the center outward. This resulted in a curve representing energy distribution across different frequencies.
- Log-Log Transformation:
  Data was converted to a logarithmic scale, where the profile of real photographs should approximate a straight line.
- SE (Square Error) Residuals:
  A linear regression of the data (ideal "1/f" line) was calculated, and the deviations (residuals) of the actual curve from this model were stored.
- Neural Network Training:
  The processed 1D vectors (Log-RAPSD and SE residuals) were used to train a 1D Convolutional Neural Network (ResNet1D) [3] to classify images as either Real or Fake.
- Evaluation**:**
  The trained model was tested on a separate dataset to evaluate its accuracy and ability to generalize to unseen data.

Through this process, each image was reduced to a 1D vector for the R, G, B, and Gray channels. This reduction significantly lowered computational complexity and enabled efficient training even on consumer-grade hardware.

2.1 Neural Network Model

Given the nature of the input data (1D signals similar to audio waveforms), a 1D Convolutional Neural Network [3] was selected for classification, specifically a custom ResNet1D architecture. The model was designed to learn and recognize the specific shapes of spectral curves:

Real Class: Typically characterized by a smooth, linear decay of energy across frequencies.

Fake Class: Characterized by an irregular spectrum, often exhibiting sudden drops or distinct peaks (spikes) at high frequencies caused by up-sampling artifacts.

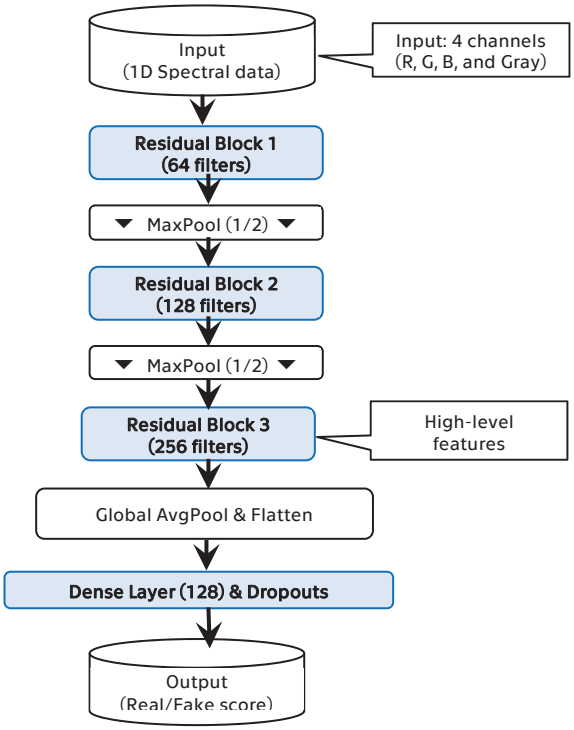


Figure 1 - Architecture scheme of the ResNet1D model [5] for 1D spectral data classification

3. Results and Discussion

One of the key findings of the experiment was that anomalies in synthetic image data are often visible even upon visual inspection of the RAPSD graphs. When analyzing the Log-Log RAPSD plots, significant peaks were observed at specific frequencies in a large portion of the artificial images.

These spectral peaks often corresponded to specific visual or generative artifacts, such as:

- Unnatural blurring and smudging in texture-heavy areas.
- Added synthetic elements, such as digital filters or effects overlaying the image.
- Checkerboard artifacts, which are a typical byproduct of image generation using GANs (Generative Adversarial Networks) due to imperfect up-convolution operations.

Experiments were conducted on a publicly available dataset from KK (Deepfake and Real Images) [4]. However, the analysis revealed that even the "Real" dataset contained images with anomalous spectral characteristics. Upon inspection, these were often photographs with heavily applied filters (e.g., from social media platforms like Instagram) or images artificially upscaled from lower resolutions. These modifications disrupt the natural frequency characteristic, necessitating the exclusion of such data from the training set to avoid confusing the model.

Another factor affecting accuracy was the depth of field. Real photographs taken with high-quality optics and a shallow depth of field (blurred background) exhibited higher SE (Square Error) values compared to fully focused images, which initially led to false positives.

3.1 Examples of Intermediate Results

- 2D FFT (Fast Fourier Transform) [1]:

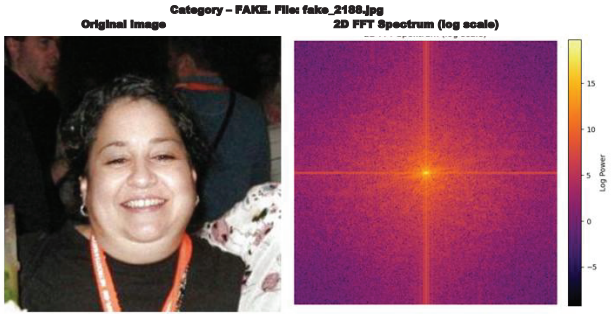


Figure 2 -2D Fast Fourier Transform (FFT) [1] amplitude spectrum of a fake photograph from the dataset [4]. Category – FAKE. Left – original image, right – 2D FFT Spectrum in log-log scale

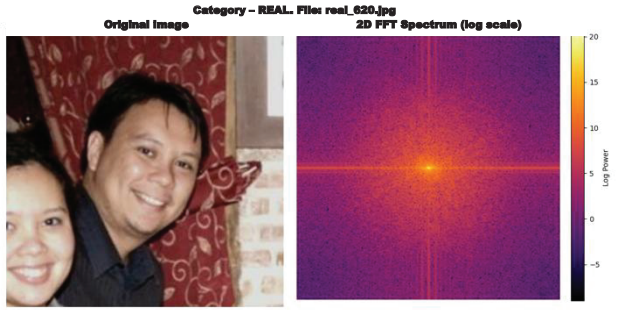


Figure 3 - 2D Fast Fourier Transform (FFT) [1] amplitude spectrum of a real photograph from the dataset [4]. Category – REAL. Left – original image, right – 2D FFT Spectrum in log-log scale

- RAPSD (Radially Averaged Power Spectral Density) [2] in Log-Log Transformation:

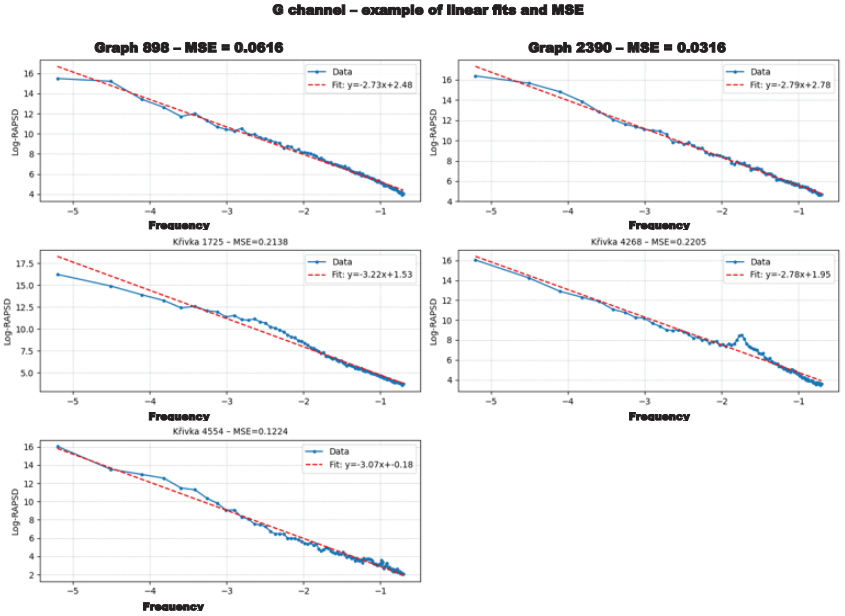


Figure 3 - Comparison of 1D RAPSD [2] profiles from grey color data (Log-Log scale)

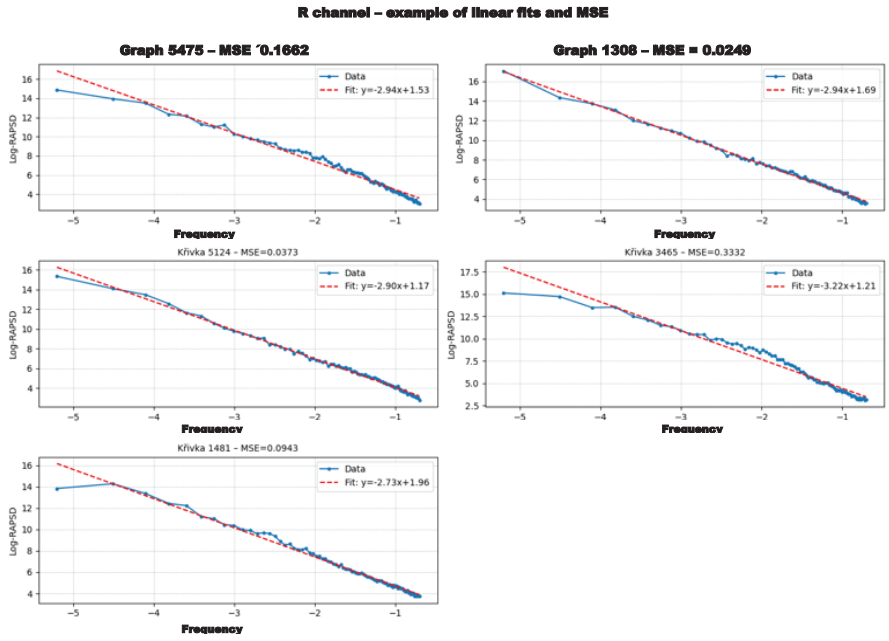


Figure 4 - Comparison of 1D RAPSD [2] profiles from red color data (Log-Log scale)

- SE (Square Error) Residuals:

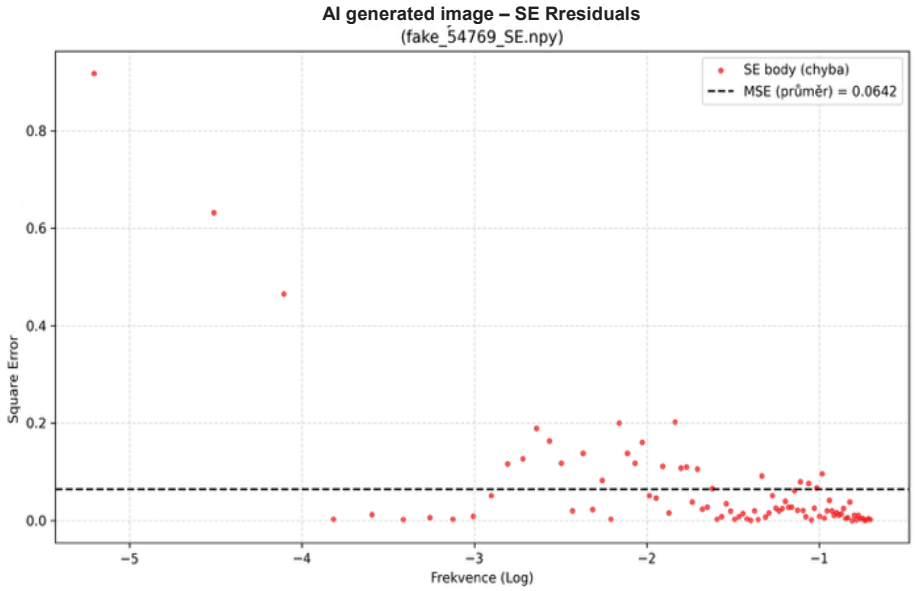


Figure 6 - Visualization of Square Error (SE) residuals on fake data

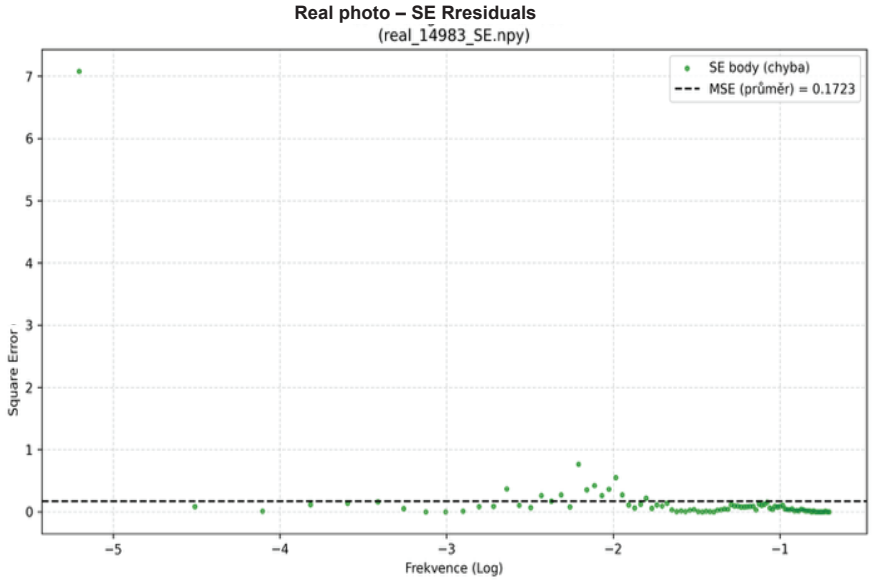


Figure 5 - Visualization of Square Error (SE) residuals on real data

- Neural Network Training:

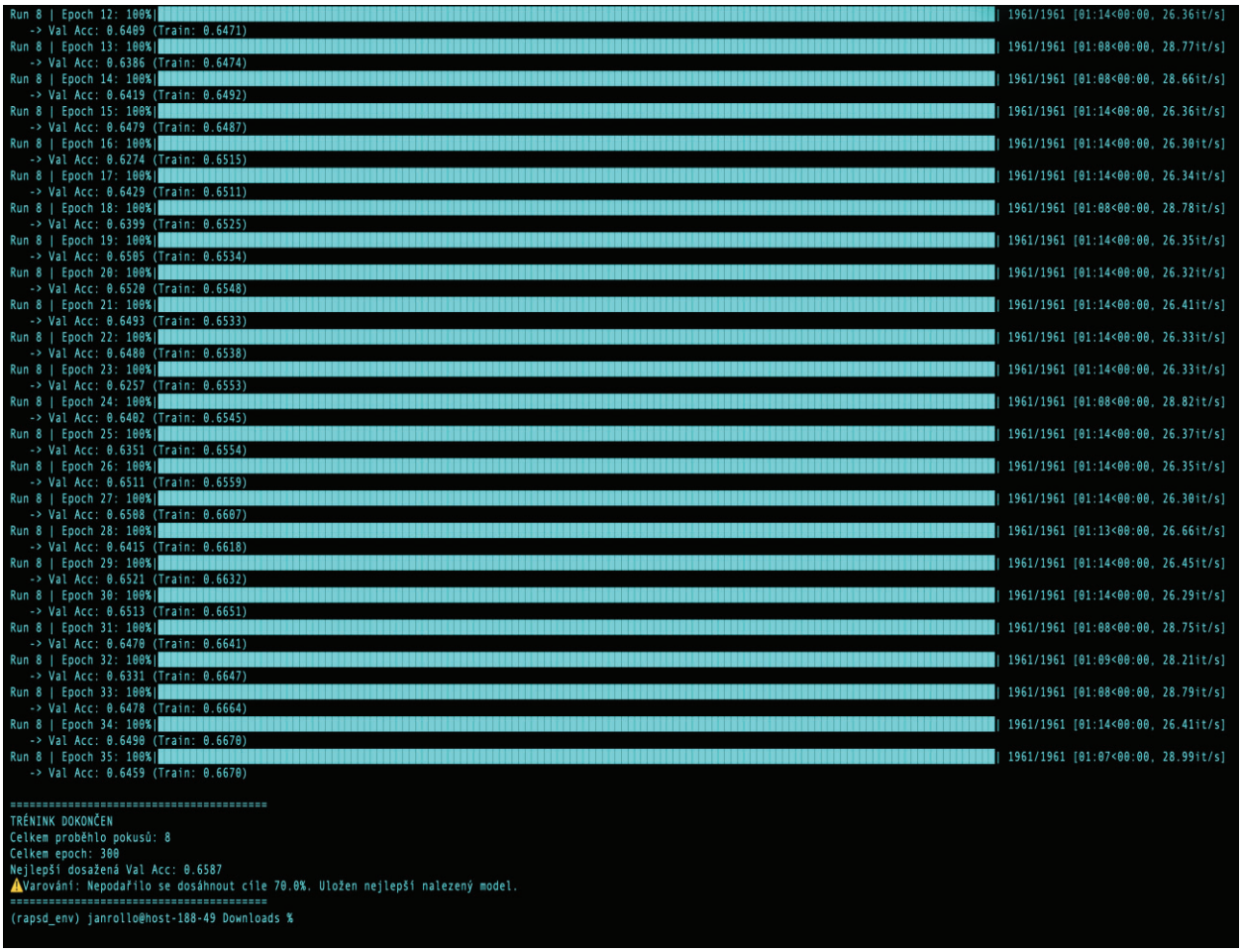


Figure 7 - Training and validation accuracy curves during the learning process. The convergence of both curves around 66%

- Evaluation**:**

Dataset analysis and preprocessing initial training on the uncleaned Kaggle dataset [4] showed strong signs of overfitting. While accuracy on the training set rose to 75%, validation stagnated around 45–55%. The reason for the low success rate was the aforementioned fundamental flaws in the "Real" dataset [4] (revealed via visual inspection, listing Real photos with the highest MSE error, and listing photos not containing a human face).

- Absence of faces: Many images did not contain a human face but only noise, or they were photographs of cartoon drawings of human faces.

- Digital filters: A significant number of "real" photographs contained added digital layers (e.g., snow effects, text overlays, strong Instagram filters). These modifications introduce high-frequency noise and sharp edges into the frequency spectrum, which the model erroneously interpreted as artifacts of generative networks (Fake). Several of these elements appeared in the dataset even after initial cleaning, which may have been one of the reasons for the poorer results.

To resolve most of these issues by preventing the neural network from learning on this invalid data, a two-phase cleaning process was implemented:

1st **Face Detection (MTCNN)**: The MTCNN detector was used to automatically identify and remove images that did not contain a valid human face.

2nd **Listing Real photos with highest MSE or SE**: Both of these methods helped list photographs where the problem appeared most frequently (though this was not a rule for all photographs).

3rd **Spectral Filtration (Blacklist)**: A "blacklist" of indices was created for images that contained a face but were degraded by digital noise or text, which would confuse the spectral model training.

Training results after cleaning the dataset [4] and removing damaged samples, the model stabilized significantly. The final experiments used a combination of two signals as input:

- Log-RAPSD: The shape of the power spectrum.

- SE (Square Error): Residuals (deviations) from the ideal linear 1/f model.

The ResNet1D model achieved an accuracy of approximately 66% on the validation set. Although this is not an absolutely high value, the key result was the consequent convergence of the training and validation curves. By removing noise, we managed to eliminate model "hallucination," where the model stopped learning specific noise in the data and began learning general spectral characteristics.

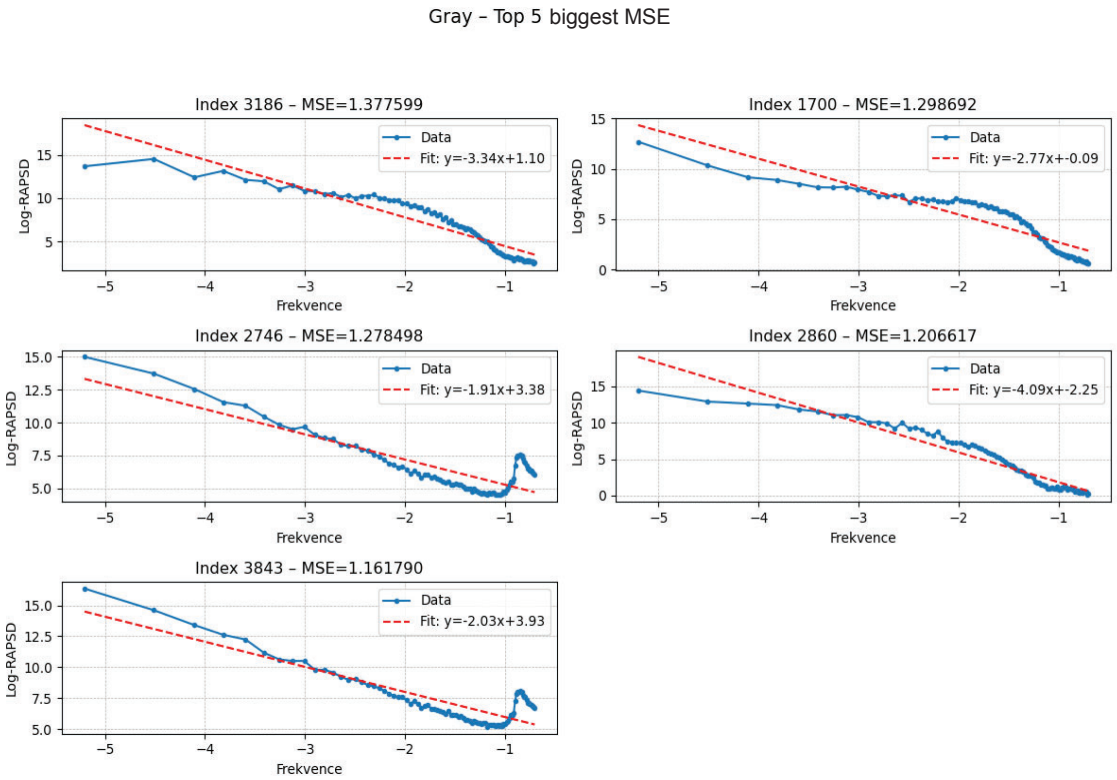


Figure 9 - Log-log graph of biggest MSE in dataset of RAPSD data of Real photos



Figure 10 – Index 2860 [4]

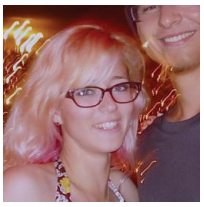


Figure 11 – Index 1700 [4]



Figure 12 – Index 3186 [4]



Figure 13 - Index 2746 [4]



Figure 8 - Index 3843 [4]

3.2 Implementation Specifics on macOS

During development on the Apple Silicon platform (M2 Pro chip), a specific technical issue was identified when training with the PyTorch framework. The training script repeatedly crashed with `RuntimeError` or `BrokenPipeError` messages during data loading.

The cause was identified as the difference in process management (`multiprocessing` start method `spawn` vs. `fork`) on macOS. The standard PyTorch `DataLoader` attempts to parallelize data loading, which conflicts with the macOS restriction on forking processes.

The issue was resolved by setting the parameter `num_workers=0` in the `DataLoader` initialization: `train_loader = DataLoader(dataset, batch_size=64, shuffle=True, num_workers=0)`

While this setting disabled parallel data loading (forcing it to run in the main process), it ensured the stability of the script execution on the local machine.

4. Conclusion

This work examined the feasibility of detecting AI-generated images using frequency spectrum analysis—specifically the Radially Averaged Power Spectral Density (RAPSD) method and Square Error (SE) residuals—utilized within a 1D Convolutional Neural Network (ResNet) architecture [3]. The experiments demonstrated that the frequency domain contains valuable signal data suitable for partially distinguishing synthetic content, particularly in the high-frequency range.

However, a primary finding of this study is the method's high sensitivity to input quality. The initial assumption of a strictly linear spectral decay (the 1/f law) in real photographs proved to be a simplification. Many real images, especially those sourced from social media, exhibit concave spectral characteristics or high-frequency disruptions due to compression, post-processing, and digital filters. Consequently, these modified real photos become mathematically indistinguishable from synthetic images for the spectral model, leading to false positive detections.

To mitigate this, a stricter definition of "real photography" was required. After thoroughly cleaning the dataset of images [4] without valid faces and those with extreme digital noise, the model achieved a stable accuracy of approximately 66%. Furthermore, it was observed that detection success rates correlate with the native resolution of the analyzed images; higher-quality inputs yield more reliable spectral fingerprints.

These results suggest that while the RAPSD/SE analysis is not suitable as a standalone "all-in-one" detector, it represents an efficient and fast tool for revealing specific types of generative artifacts. In future work, this method will be integrated as a supplementary module into more complex detection systems. The combination of spectral analysis with spatial visual analysis is expected to enable robust detection that is resilient to both visually superior deepfakes and filtered real photographs.

5. References

[1] V. Hlaváč, "Fourier Transformation," *Teaching Material – Digital Image Processing*. Prague: Czech Technical University in Prague, CIIRC. [Online]. Available: https://people.ciirc.cvut.cz/hlavac/TeachPresCz/11DigZprObr/12FourierTxCz.pdf

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/

[3] ÚFAL MFF UK, "Deep Learning (NPFL138)," *Online lectures*. Prague: Charles University, Faculty of Mathematics and Physics. [Online]. Available: https://ufal.mff.cuni.cz/courses/npfl138/2324-summer#lectures

[4] M. Karki, "Deepfake and Real Images," *Kaggle Dataset*. [Online]. Available: https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images/data

[5] PlantText, "PlantText UML Editor," *Online tool*. Used for generating the neural network schema. [Online]. Available: https://www.planttext.com

STUDIE PROVEDITELNOSTI SPOJOVÁNÍ SKLA A KOVU POMOCÍ INDIA PRO MONTÁŽ OPTICKÝCH PRVKŮ

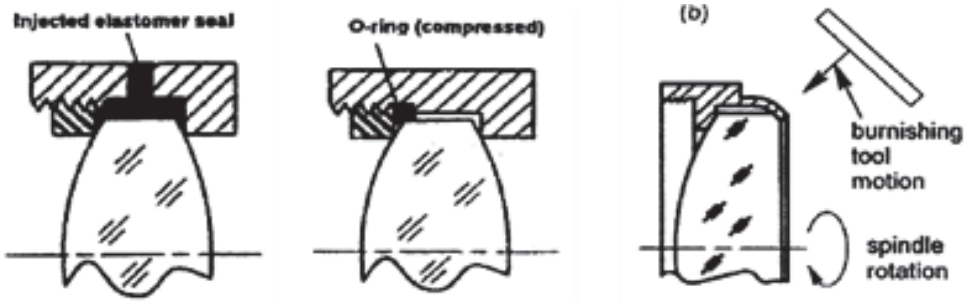
***Uliana Finaeva*[1]**

1 Ústav přístrojové a řídicí techniky, Fakulta strojní, ČVUT v Praze, uliana.finaeva@fs.cvut.cz

Abstrakt: Optické soustavy obsahují řadu optických prvků, které je nutné spolehlivě upevnit v mechanické konstrukci. Běžné metody montáže nemusí být vhodné pro aplikace s vysokými nároky na přesnost, teplotní stabilitu nebo použití ve vakuu. Jednou z možných alternativ je pájené spojení skla a kovu, jehož realizaci však komplikuje zejména špatná smáčivost skla běžnými pájkami a rozdílná teplotní roztažnost spojovaných materiálů. Tato práce se zabývá možností využití india, které se vyznačuje nízkou teplotou tání a vysokou plasticitou. Při experimentálním nanášení india za zvýšené teploty a mechanické aktivaci povrchu se za určitých podmínek podařilo vytvořit soudržné spojení skla s kovem. Další práce bude zaměřena na optimalizaci procesu a charakterizaci mechanických vlastností spoje.
Klíčová slova: pájený spoj, indium, optomechanika

1. Úvod

Optické prvky v optických systémech je nutné vhodným způsobem uložit do mechanické konstrukce tak, aby byla zachována jejich poloha, orientace a požadované optické vlastnosti. Mezi běžné způsoby uložení patří závitové pojistné kroužky, elastické O-kroužky, lepené spoje, pružné elementy nebo mechanické tváření držáku přes okraj optického prvku [1]. V praxi existuje celá řada modifikací těchto základních principů, které umožňují přizpůsobit uložení konkrétní aplikaci [2-4].



Obr. 1 Příklady klasických metod uložení optických prvků [1].

Vyšší nároky na konstrukci vznikají zejména u přesných optických soustav a zařízení pracujících ve specifických podmínkách. Změny teploty a rozdílné koeficienty teplotní roztažnosti optického prvku a jeho držáku mohou způsobovat změny polohy, deformace a vznik mechanického napětí. Konstrukce může být dále zatěžována vibracemi, mechanickými rázy, provozem ve vakuu nebo lokálním ohřevem způsobeným intenzivním optickým zářením. Mechanické napětí přenášené na optický prvek je nežádoucí

zejména proto, že může vést k deformaci optických ploch a u některých materiálů také ke vzniku napěťového dvojlomu. Způsob uložení optického prvku tak může přímo ovlivnit výsledné vlastnosti celé optické soustavy [5].

Specifickou skupinu představují trvalé spoje optických prvků s kovovými držáky. Jejich realizace je komplikována rozdílnými fyzikálními a chemickými vlastnostmi skla a kovu, a především rozdílnou teplotní roztažností obou materiálů. Při ochlazování spoje tak může vznikat zbytkové mechanické napětí, které může vést až k deformaci nebo poškození optického prvku. Pro vytvoření trvalého spoje lze využít několik technologických přístupů. Nejjednodušší je lepení, jehož použití však může být omezeno teplotní stabilitou lepidla, stárnutím, odplyňováním ve vakuu nebo napětím vznikajícím při vytvrzování [6]. Alternativou jsou například difuzní [7], laserové [8] a anodické spojování [9]. Tyto metody mohou vytvářet pevné spoje, často však vyžadují přesně řízené podmínky, speciální technologické vybavení nebo vyšší procesní teploty.

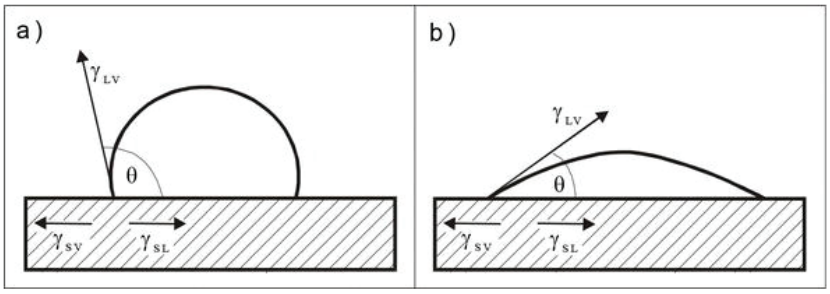
Zajímavou alternativou pro uložení optických prvků je pájení, které umožňuje vytvořit kompaktní kovový spoj bez použití organického lepidla. Jeho výhodou může být také využití ve vakuu nebo za náročnějších teplotních podmínek. Zásadním problémem při přímém pájení skla je však jeho špatná smáčivost běžnými pájkami. V praxi se proto využívá například předchozí metalizace povrchu skla [10], aktivní pájky [11] nebo mechanicky či ultrazvukově aktivované procesy [12]. Výslednou kvalitu spoje ovlivňuje také stav povrchu a přítomnost oxidačních vrstev.

Pro nízkoteplotní pájení je zajímavým materiálem indium. Jeho teplota tání je přibližně 157 °C a zároveň se jedná o velmi měkký a tvárný kov. Plastická deformace india může do určité míry kompenzovat rozměrové změny a mechanické napětí vznikající mezi materiály s rozdílnou teplotní roztažností. Indium a pájky na jeho bázi již byly zkoumány pro spojování skla, křemene a dalších optických materiálů s kovovými držáky [13].

Předkládaná práce se proto zabývá experimentálním ověřením relativně jednoduchého způsobu vytvoření spoje mezi sklem a kovovým držákem pomocí india. Zvolený postup je založen na zahřátí spojovaných součástí, nanesení india a mechanické aktivaci povrchu během jeho aplikace. Předběžné experimenty ukázaly, že za určitých podmínek lze tímto způsobem vytvořit soudržný spoj skla s kovem. V této fázi práce však nebyly kvantitativně stanoveny všechny technologické parametry ani mechanická pevnost vytvořeného spoje. Práce má proto charakter studie proveditelnosti, na kterou může navázat optimalizace technologického postupu a experimentální charakterizace výsledných spojů.

2. Princip spojování skla a kovu pomocí india

Pájení je založeno na vytvoření spoje pomocí přídavného kovového materiálu, který se při procesu roztaví, zatímco základní spojované materiály zůstávají v pevném stavu. Pro vytvoření kvalitního spoje je důležité, aby roztavená pájka vytvořila dostatečný kontakt s povrchem obou spojovaných materiálů. Tento proces je úzce spojen se smáčivostí povrchu. Smáčivost povrchu lze charakterizovat pomocí kontaktního úhlu (viz obr. 1) a adhezní práce, která vyjadřuje energii potřebnou k oddělení kapaliny od povrchu substrátu [14].



Obr. 2 Ilustrační schéma a) nesmáčivého a b) smáčivého materiálů [14].

V případě pájení smáčivost popisuje schopnost roztavené pájky rozprostřít se po čistém povrchu spojovaného materiálu a vytvořit s ním kontakt. Adheze na rozhraní může být zprostředkována fyzikálními interakcemi, například van der Waalsovými silami, případně vznikem chemických vazeb mezi pájkou a substrátem. V případě skla je situace komplikovanější než při pájení dvou kovových součástí, protože sklo není běžnými pájkami snadno smáčitelné a výsledné chování výrazně závisí na chemickém a mechanickém stavu povrchu [14]. Vzhledem k obtížné smáčivosti skleněných povrchů nelze při jejich pájení posuzovat pouze vlastnosti samotné pájky. Významnou roli hraje rovněž způsob aktivace povrchu a použitý technologický postup. Vybrané příklady pájek a technologických postupů používaných při experimentálním spojování skla jsou uvedeny v tab. 1. Kratký seznam příkladů obsahuje metody jejichž pracovní teploty nepřesahuji 300 stupňů, pro zachování jednoduchosti manipulaci vzorky.

Tab. 1. Vybrané příklady pájení skla popsané v odborné literatuře, teploty do 300 °C.

Pájka / spojovací materiál	Teplota uvedená ve zdroji	Spojované materiály	Způsob aktivace / technologie
Sn-28Bi	138,5 °C	SiO_2 / Al 2024	Ultrazvuková asistence [12]
Sn-Ag-Cu (SAC0307)	227 °C	Nerez, K9	CW Laser [15]
Sn-9Zn	240 °C	Borofloat 33 / Cu	Anodická aktivace [16]
In	157 °C	sklo / kov – tato práce	mechanická aktivace kartáčem

Z uvedené tabulky je patrné, že spojování skla pájením lze realizovat různými kombinacemi pájecího materiálu a způsobu aktivace rozhraní. Pro experimentální ověření v této práci bylo zvoleno čisté indium (99,9 %) v kombinaci s jednoduchou mechanickou aktivací povrchu. Jeho teplota tání je přibližně 157 °C a jedná se o velmi měkký a tvárný kov. Nízká teplota tání omezuje tepelné zatížení během samotného spojování a velikost zbytkového napětí vznikajícího při ochlazování materiálů s rozdílnými koeficienty teplotní roztažnosti. (Přestože příprava skleněného povrchu zahrnuje krátkodobý ohřev na vyšší teplotu, vlastní spojení předem indiovaných součástí probíhá v oblasti 150–160 °C). Plasticita india umožňuje částečně kompenzovat vzájemné rozměrové změny spojovaných materiálů a tím omezovat přenos mechanického napětí do optického prvku nejen za vysokých teplot.

Při přípravě skleněného povrchu lze vycházet také z doporučení výrobce Indium Corporation pro spojování nekovových materiálů. Pro sklo, křemen a glazovanou keramiku je doporučeno povrch nejprve zahřát přibližně na 350 °C a následně před aplikací pájky ochladit na nižší teplotu. Takový postup může podpořit následné vytvoření přilnavé vrstvy india na povrchu. Vlivem mechanického a tepelného zatížení může ve skle vznikat dočasné nebo zbytkové napětí. Při nerovnoměrném ohřevu se povrchové vrstvy zahřívají a roztahují rychleji než vnitřní části materiálu, což v důsledku nízké tepelné vodivosti skla vede ke vzniku vnitřního napětí. Při dostatečně pomalém vyrovnání teplot může toto napětí postupně zaniknout, zatímco při rychlém nebo nerovnoměrném ochlazování může část napětí v materiálu zůstat. Zbytkové napětí zvyšuje náchylnost skla k prasknutí při následném tepelném či mechanickém zatížení a lze jej omezit vhodně řízeným žíháním. U přesných optických soustav je proto žádoucí minimalizovat nejen mechanické namáhání optického prvku, ale také teplotní gradienty vznikající během procesu montáže.

Z hlediska tepelného zatížení je však vedle maximální dosažené teploty důležitá také její prostorová rovnoměrnost. V dříve provedených autory experimentech s ohřevem skleněných vzorků proudem horkého vzduchu bylo pozorováno, že vzorky o tloušťce 4-8 mm snášely bez porušení teploty nejméně okolo 300 °C a rovněž teplotní oblast přibližně 150–160 °C odpovídající aplikaci india. K destrukci vzorků docházelo zejména při nerovnoměrném ohřevu, přičemž místo porušení bylo spojeno s lokální oblastí nižší teploty vzdálenější od centra proudu horkého vzduchu. Toto pozorování ukazuje, že při návrhu postupu je třeba omezit nejen absolutní teplotu skla, ale především teplotní gradienty, které mohou

způsobovat vznik tepelně indukovaného mechanického napětí.

Zásadním krokem zkoumané technologie je následná mechanická aktivace povrchu během nanášení roztaveného india. Samotné roztavení india na povrchu skla a kovu nemusí vést k jeho dostatečnému rozprostření a přilnutí. Mechanické roztírání narušuje povrchové vrstvy a podporuje vytvoření kontaktu mezi roztaveným kovem a substrátem. Přesný mechanismus vzniku rozhraní však nebyl v rámci této práce analyticky stanoven.

3. Návrh a experimentální ověření technologického postupu

Před zahájením experimentu byly skleněný vzorek i kovový protikus očištěny vodou se saponátem za použití plastového kartáče. Povrchy byly následně odmaštěny acetonem a ponechány k vyschnutí. Povrch kovového protikusu byl dále mechanicky upraven brusivem SiC o zrnitosti 80. Po broušení byl kovový díl znovu očištěn saponátem, odmaštěn acetonem a vysušen.

Skleněný vzorek byl vložen do stolní tavicí pece určené pro zpracování kovů a během přibližně 30 minut zahřát z pokojové teploty (cca 25 °C) na 350 °C. Po dosažení této teploty byl ponechán k postupnému ochlazení na přibližně 160 °C. Indiová fólie byla před použitím očištěna acetonem a vysušena. Skleněný vzorek byl následně vyjmut z pece a na jeho zahřátý povrch byla přiložena indiová fólie. Po jejím roztavení bylo indium intenzivně mechanicky roztíráno kartáčem z korozivzdorné oceli, dokud nebyla na povrchu vytvořena souvislá vrstva s charakteristickým zrcadlovým vzhledem.

Obdobným způsobem byl zpracován kovový protikus. Po zahřátí byl na jeho připravený povrch nanesen a pomocí kartáče mechanicky rozetřen roztavený indiový materiál. Oba vzorky s vytvořenou vrstvou india byly následně opět vloženy do pece a zahřáty na přibližně 150–160 °C.

Po vyjmutí z pece byly indiované plochy skleněného a kovového vzorku přiloženy k sobě a mechanicky stlačeny. Sestava byla před upnutím obalena textilií, která sloužila jako ochrana skleněného vzorku před přímým kontaktem s čelistmi svěráku. Přítlak byl vytvářen ručním utahováním svěráku až do okamžiku, kdy již nebylo možné jeho rukojeť běžnou ruční silou dále otáčet. Takto upnutá sestava byla ponechána k volnému ochlazení po dobu nejméně jedné hodiny. Po uvolnění ze svěráku bylo pozorováno vytvoření soudržného spoje mezi skleněnou a kovovou částí.

Použitý postup měl v této fázi charakter předběžného experimentálního ověření technologie. Po vychladnutí a vyjmutí sestavy ze svěráku nedošlo při běžné manipulaci k samovolnému oddělení součástí. Přítlačná síla při spojování nebyla měřena a nebyly kvantitativně stanoveny ani skutečná plocha spoje, drsnost povrchů po mechanické úpravě nebo tloušťka vytvořené vrstvy india. Experiment proto dokládá především možnost vytvoření spoje za uvedených podmínek, nikoliv jeho kvantifikovanou pevnost nebo reprodukovatelnost.

4. Závěr

V rámci práce byla experimentálně ověřována možnost vytvoření trvalého spoje mezi sklem a kovovým substrátem pomocí čistého india. Navržený postup využívá tepelnou přípravu spojovaných součástí, mechanickou úpravu kovového povrchu a mechanickou aktivaci roztaveného india pomocí kartáče z korozivzdorné oceli. Tímto způsobem se podařilo vytvořit souvislou vrstvu india na povrchu skla i kovu a následným zahřátím, přitlačením a ochlazením obou částí bylo dosaženo soudržného spoje.

Experiment tak potvrdil základní proveditelnost navrženého způsobu spojování. Pro další optimalizaci technologie se jako významné jeví zejména řízení teplotního režimu, příprava povrchů a způsob mechanického roztírání india. Z hlediska skleněného prvku je nutné věnovat pozornost zejména rovnoměrnosti ohřevu a omezení teplotních gradientů, které mohou vést ke vzniku vnitřního napětí a následnému porušení skla.

V současné fázi však nebyla stanovena pevnost ani skutečná plocha vytvořeného spoje a nebyly přesně řízeny parametry, jako jsou přítlačná síla, drsnost povrchu nebo tloušťka vrstvy india. Výsledky proto nelze považovat za úplnou charakterizaci technologie, ale za úspěšné prvotní ověření jejího

principu. Další práce by měla být zaměřena především na zvýšení reprodukovatelnosti procesu, přesnější řízení technologických parametrů a měření mechanické pevnosti spoje. Pro využití v optomechanických aplikacích bude vhodné rovněž ověřit stabilitu spoje při teplotních změnách a jeho vliv na mechanické napětí a polohu upevněného optického prvku.

Poděkování

Práce byla podpořena grantem SGS24/124/OHK2/3T/12 "Nová technologie montáže skleněných optických prvků"

Literatura

[1] YODER Jr., P.R. Lens Mounting Techniques, Los Angeles: International Society for Optical Engineering, 1983.
[2] Institut national d'optique, Québec (CA). Mounting of an optical element in a barrel using a flexible ring. Inventor: Frédéric Lamontagne, Nichola Desnoyers. US10288095 (B2). Appl. no. 14/861, 574 20190514 Int. Cl. GO2B 7/02; F16B 2/06. United States Patent and Trademark Office.
[3] L3 Communications Sonoma EO Inc. Stress decoupling optic mount. Inventor: Craig Wheeler. US7538959 (B1). Appl. no. 11/382,456 20090526 Int. Cl. G02B 7/02. United States Patent and Trademark Office.
[4] Northrop Grumman Corp. Strain free temperature-compensated optical mounts. Inventor: Edwin G. Haas. US5523893. Appl. no. 08/294,038 19960604 Int. Cl. G02B 7/02. United States Patent and Trademark Office.
[5] YODER, P. R.; VUKOBRATOVICH, D. Opto-Mechanical Systems Design. Volume 1: Design and Analysis of Opto-Mechanical Assemblies. 4th ed. CRC Press, 2015.
[6] DALY, J. et al., Structural adhesives for bonding optics to metals: a study of optomechanical stability, Proc. SPIE 4444, Optomechanical Design and Engineering 2001, null (5 Nov 2001); https://doi.org/10.1117/12.447299
[7] JACOBS, M.et al., Glass-Metal Joining in Nuclear Environment: The State of the Art. In Conference Proceedings: Exploring Structure, Processing, and Applications Across Multiple Materials Systems, 2007.
[8] WANG, Y. et al., Welding of 304 stainless steel and glass using high-repetition frequency femtosecond laser, Mater. Res. Express, 8, 2021, pp. 106523
[9] YANG, L. et al., Enhanced Adhesion of Copper Films on Fused Silica Glass Substrate by Plasma Pre-Treatment. Materials., 16(14), 2023, pp. 5152. https://doi.org/10.3390/ma16145152
[10] RIBES PLEGUEZUELO P.. High-precision optomechanical lens system for space applications assembled by a local soldering technique. In: Optical Engineering, 2016, Volume 55, Number 6, pp. 065101. [9 May 2023]. DOI: 10.1117/1.OE.55.6.065101
[11] RAMIREZ, G. et al., Bonding nature of rare-earth-containing lead-free solders. Appl. Phys. Lett., 80 (3), 2002, pp. 398–400. https://doi.org/10.1063/1.1435075
[12] GAO, X. et al., High-strength joining of silica glass and 2024 aluminum alloy by ultrasonic-assisted soldering with Sn-Bi low temperature solder, Journal of Manufacturing Processes, 101, 2023, pp. 1482-1496, https://doi.org/10.1016/j.jmapro.2023.07.010.
[13] SABININ, V. E., SOLK, S. V., Using indium in optical instrumentation, J. Opt. Technol., 77, 2010, pp. 762-763
[14] KOLENAK, R. Recent Advances in Solderability of Ceramic and Metallic Materials with Application of Active Solders and Power Ultrasound, Recent Progress in Soldering Materials. InTech, 2017. doi: 10.5772/intechopen.69552.
[15] REN, H.et al., Laser soldering of stainless steel and glass assisted with copper coating, Optics & Laser Technology, 176, 2024, pp. 110969, https://doi.org/10.1016/j.optlastec.2024.110969.
[16] HU, L. et al., Glass-Cu joining by anodic bonding and soldering with eutectic Sn-9Zn solder, Journal of Alloys and Compounds, 789, 2019, pp. 558-566, https://doi.org/10.1016/j.jallcom.2019.02.257.

NÁVRH A IMPLEMENTACE WEBOVÉ APLIKACE PRO MONITORING PRŮMYSLOVÝCH ZAŘÍZENÍ

Patrik Brejla

České vysoké učení technické v Praze
Fakulta strojní
patrikbrejla@seznam.cz

Abstrakt: Článek se zabývá návrhem a implementací webové aplikace pro monitoring průmyslových zařízení v průmyslovém prostředí. Popsána je architektura systému, komunikace s průmyslovými zařízeními pomocí technologie OPC UA, využití systému KEPServerEX a implementace backendové části aplikace pomocí frameworku FastAPI. Součástí práce je také návrh databázové vrstvy, autentizace uživatelů a vizualizace dat v reálném čase.
Klíčová slova: FastAPI, OPC UA, KEPServerEX, IIoT, průmyslová automatizace, monitoring zařízení

Abstract: This paper focuses on the design and implementation of a web application for monitoring industrial devices in an industrial environment. The system architecture, communication with industrial devices using OPC UA technology, the use of the KEPServerEX platform, and the implementation of the backend application using the FastAPI framework are described. The paper also includes the design of the database layer, user authentication, and real-time data visualization.
Keywords: FastAPI, OPC UA, KEPServerEX, IIoT, industrial automation, device monitoring

Anotace

Tato práce se věnuje analýze digitalizace výroby ve společnosti Doosan Bobcat EMEA s.r.o. a následnému návrhu i implementaci webové aplikace pro centralizovaný monitoring průmyslových zařízení. Hlavním cílem je sjednotit roztříštěná data z různých systémů a poskytnout uživatelům jednotné rozhraní pro správu, vizualizaci a diagnostiku strojů v reálném čase pomocí standardu OPC UA a moderních webových technologií.

1 Úvod

V současné době dochází v průmyslovém prostředí k výraznému nárůstu digitalizace a automatizace výrobních procesů. S tímto trendem, často označovaným jako Průmysl 4.0 [4], souvisí rostoucí potřeba efektivního sběru, správy a vizualizace dat z výrobních zařízení. Ve velkých podnicích však často nastává situace, kdy jsou data a nástroje pro jejich zpracování roztříštěny napříč různými systémy, což komplikuje jejich využití a snižuje přehlednost celého prostředí.

Tato práce vznikla na základě reálných potřeb společnosti Doosan Bobcat EMEA s.r.o., konkrétně výrobního závodu na Dobříši. Ve společnosti je využíván systém KEPServerEX od společnosti PTC [1], který slouží jako komunikační rozhraní pro připojení širokého spektra průmyslových zařízení, od PLC automatů až po robotická pracoviště. Přestože tento systém umožňuje centralizovaný sběr dat, chyběl jednotný systém poskytující přehled o fyzickém umístění zařízení, jejich síťové konfiguraci a aktuálním provozním stavu.

Cílem práce je tedy návrh a implementace webové aplikace, která tyto nedostatky odstraňuje, snižuje závislost na drahých externích licencích a vytváří základ pro další rozvoj v oblasti průmyslové digitalizace.

2 Analýza současného stavu

Ve společnosti Doosan Bobcat EMEA s.r.o. byly identifikovány hlavní nedostatky v oblasti správy dat. Existující softwarové nástroje pro monitoring jsou často dodávány externími výrobci a jsou omezeny na specifické typy zařízení, například pouze na svařovací roboty. Tyto systémy jsou licenčně zpoplatněny, což zvyšuje náklady při každém požadavku na rozšíření funkcionality.

Pro ostatní zařízení, jako jsou CNC stroje, PLC automaty nebo momentové nástroje, byla evidence vedena nejednotně, nejčastěji v aplikaci Microsoft Excel. Tento způsob správy trpí vysokou chybovostí, neaktuálností dat a absencí propojení na reálná provozní data. Absence jednotné vizualizace, například ve formě interaktivní mapy výrobního prostoru, významně ztěžovala diagnostiku při výpadcích komunikace, kdy bylo nutné manuálně dohledávat IP adresy a umístění datových zásuvek.

3 Teoretická východiska

Pro návrh nového řešení bylo nutné vyjít z principů Industrial Internet of Things (IIoT) [3]. IIoT představuje propojení průmyslových senzorů a řídicích systémů se softwarovými aplikacemi za účelem optimalizace výroby [6].

3.1 Průmyslová komunikace a OPC UA

Klíčovým prvkem v této vrstvě je standard OPC Unified Architecture (OPC UA) [9]. Na rozdíl od starších proprietárních protokolů nabízí OPC UA platformní nezávislost a vysokou míru zabezpečení [10]. Umožňuje vytvářet sémanticky bohaté informační modely, kde jsou data organizována do uzlů (nodes) obsahujících nejen hodnotu, ale i metadata.

V architektuře systému hraje roli integračního middlewaru software KEPServerEX [2, 19]. Ten pomocí specifických ovladačů (drivers) komunikuje s heterogenními zařízeními (Siemens, Allen-Bradley) a následně tato data publikuje přes OPC UA server směrem k nadřazeným aplikacím.

3.2 Webové technologie a backend

Moderní webové aplikace jsou postaveny na architektuře klient-server [5]. Pro vývoj backendu byly zvažovány tři hlavní frameworky jazyka Python: FastAPI, Flask a Django.

FastAPI bylo zvoleno pro svou vysokou výkonnost, nativní podporu asynchronního programování a automatické generování dokumentace [11]. Django sice nabízí robustní systém [13], ale pro potřeby rychlé integrace s průmyslovými servery se jevilo jako příliš komplexní. Flask [12] naopak postrádá některé moderní prvky, které FastAPI implementuje již v základu.

4 Analýza požadavků a návrh řešení

Na základě provedené teoretické analýzy a identifikovaných nedostatků současného stavu byly definovány požadavky na nový systém, které vycházejí z reálných potřeb podnikového prostředí závodu v Dobříši. Tyto

požadavky tvoří základ pro následný návrh vícevrstvé architektury, způsobu zpracování dat a volbu technologií.

4.1 Funkční požadavky

Funkční požadavky definují konkrétní služby a operace, které musí systém vykonávat pro koncové uživatele [5].

- **Evidence zařízení:** Základním požadavkem je vytvoření rozšiřující evidence průmyslových zařízení. Protože systém KEPServerEX standardně neobsahuje všechna metadata, aplikace tyto informace doplňuje o IP adresy, fyzické označení datových zásuvek, příslušnost k výrobní hale a přesné souřadnice umístění [1]. Tato struktura umožňuje efektivní propojení provozních (OT) a informačních (IT) dat.
- **Monitoring v reálném čase:** Klíčovou funkcionalitou je sledování aktuálního stavu technologií bez nutnosti přepínat mezi více nástroji. Data jsou získávána pomocí standardu OPC UA a u každého zařízení jsou sledovány stavové proměnné (např. *Online, Offline, Error, Idle* nebo *Maintenance*) [10]. Tyto stavy jsou v systému reprezentovány barevným značením pro okamžitou orientaci uživatele [9].
- **Interaktivní mapa:** Systém nabízí vizualizaci výrobního prostoru pro obě haly (horní i spodní závod) a speciální detail pro robotická pracoviště. Mapa je navržena hierarchicky – uživatel může přecházet od celkového pohledu na halu přes výrobní linky až k detailu konkrétního stroje, kde jsou k dispozici fotografie, síťové parametry a přímé odkazy na technickou dokumentaci nebo 3D modely.

4.2 Nefunkční požadavky

Nefunkční požadavky určují kvalitativní vlastnosti a provozní omezení systému [5].

- **Škálovatelnost a výkon:** Systém je navržen tak, aby zvládl budoucí rozšíření na další výrobní lokality a nárůst počtu uživatelů. Pro zajištění plynulé odezvy byl stanoven interval aktualizace stavových dat na 20 sekund, což představuje optimální kompromis mezi aktuálností informací a zatížením podnikové sítě [7].
- **Bezpečnost a dostupnost:** Aplikace je provozována výhradně v interní síti společnosti, případně přes zabezpečené VPN připojení [8]. Systém podporuje různé uživatelské role (administrátor, technolog, údržba), které definují rozsah oprávnění pro editaci dat nebo přístup k administrátorským funkcím [17].

4.3 Architektura systému

Navržená architektura je striktně vícevrstvá, což zajišťuje modularitu a snadnou údržbu. Celý ekosystém je provozován v kontejnerizovaném prostředí pomocí nástroje Docker Compose [18].

- **Integrační a komunikační vrstva:** Na nejnižší úrovni jsou průmyslová zařízení (PLC, roboty, momentové nástroje), která komunikují s KEPServerEX. Ten plní roli integračního middlewaru a sjednocuje heterogenní data do jednotného adresního prostoru OPC UA [19].
- **Aplikační vrstva (Backend):** Serverová část postavená na frameworku FastAPI využívá asynchronní knihovnu `python-opcua`. Backend vystupuje jako OPC UA klient, který dynamicky odebírá data ze zdroje, transformuje je a předává frontendu.
- **Databázová vrstva:** Pro ukládání statických metadat a logování uživatelských akcí byla zvolena relační databáze PostgreSQL [14]. Využívá se zde také rozšíření `pgvector`, které umožňuje pokročilé sémantické vyhledávání zařízení na základě podobnosti názvů, což zvyšuje uživatelský komfort [14].

4.4 Zabezpečení a autentizace

Zvolené řešení bezpečnosti kombinuje výhody tří standardů, které společně tvoří robustní systém pro správu identit [15].

- **LDAP a OAuth2:** Proces autentizace je delegován na podnikový LDAP server, což umožňuje uživatelům využívat stávající přihlašovací údaje [15]. Model OAuth2 slouží jako zastřešující autorizační rámec definující toky dat mezi klientem a backendem [17].
- **JWT (JSON Web Token):** Po úspěšném ověření identity vygeneruje backend digitálně podepsaný bezstavový token JWT [16]. Tento token obsahuje informace o uživateli a jeho oprávněních a je přenášen v hlavičce každého následného požadavku. Díky bezstavovosti JWT je systém snadno škálovatelný, protože server nemusí uchovávat informace o aktivních relacích v paměti [16].

5 Implementace

Tato kapitola se věnuje praktické realizaci navrženého systému. Implementace vychází z navržené architektury a popisuje postupné kroky vedoucí k vytvoření funkční aplikace.

Projekt byl realizován v jazyce Python ve verzi 3.10 za využití vývojového prostředí PyCharm. Pro zajištění čistoty projektu a oddělení závislostí bylo využit virtuální prostředí (venv), ve kterém byla správa knihoven prováděna prostřednictvím nástroje `pip`. Pro správu citlivých údajů, jako jsou přihlašovací údaje k databázi, byl využit soubor `.env`.

5.1 Kontejnerizace

Zásadním krokem pro zajištění přenositelnosti a snadného nasazení aplikace byla kontejnerizace pomocí nástroje Docker [18]. Celý systém je definován a spravován pomocí nástroje Docker Compose, který se skládá ze dvou hlavních služeb:

- **Web (Backend):** Služba využívá obraz postavený na `python:3.10-bullseye`. Dockerfile zahrnuje instalaci systémových závislostí pro databázové ovladače a následné spuštění aplikace pomocí serveru Uvicorn [11].
- **DB (Databáze):** Služba reprezentuje databázový server PostgreSQL s rozšířením `pgvector`. Důležitým prvkem je využití mechanismu *healthcheck*, který ověřuje připravenost databáze, než se k ní backend pokusí připojit [14].

5.2 Integrace OPC UA

Klíčovým prvkem implementace je napojení na průmyslový server KEPServerEX [2]. Pro tento účel byla využita knihovna `python-opcua`, která umožňuje backendu vystupovat v roli OPC UA klienta.

Backend se připojuje k endpointu serveru a prochází jeho hierarchický adresní prostor. Uzly v OPC UA stromu jsou mapovány na konkrétní ID zařízení vedená v relační databázi. Provozní data nejsou do databáze ukládána permanentně, ale jsou načítána *on-demand* při požadavku uživatele [10]. Tento přístup zaručuje, že zobrazované hodnoty jsou vždy aktuální přímo ze zdroje a snižuje nároky na synchronizaci dat.

6 Uživatelské rozhraní

Uživatelské rozhraní aplikace bylo navrženo s cílem poskytnout přehledný a intuitivní nástroj pro práci s daty z průmyslových zařízení, přičemž hlavní důraz byl kladen na srozumitelnost technických informací a minimalizaci složitosti ovládání. Návrh vychází z principu centralizace – uživatel má všechny relevantní informace na jednom místě, což eliminuje potřebu přecházet mezi více specializovanými aplikacemi. Rozhraní je realizováno jako webová aplikace přístupná prostřednictvím běžného prohlížeče, což odstraňuje nutnost instalace dodatečného softwaru na klientské stanice.

Základní struktura stránek je generována na straně serveru pomocí šablonovacího systému frameworku FastAPI, zatímco dynamické prvky a interaktivita jsou zajištěny klientskou logikou v jazyce JavaScript. Vstupním bodem do systému je domovská obrazovka, která uživatele logicky směruje do jednotlivých modulů (seznam zařízení, mapy, správa) na základě jeho uživatelské role definované v systému LDAP.

6.1 Interaktivní mapa výrobního závodu

Hlavním pilířem rozhraní je interaktivní mapa, která slouží jako primární nástroj pro prostorovou orientaci v rámci výrobních hal. Mapa je navržena hierarchicky, což uživateli umožňuje plynule přecházet od celkového pohledu na halu (horní nebo spodní závod) přes konkrétní výrobní linky až k detailu jednotlivých strojů. Tento přístup zabraňuje informačnímu zahlcení a umožňuje rychlou identifikaci problémových oblastí.

Důležitým prvkem je vizuální kódování stavu zařízení pomocí barevného značení. U běžných zařízení je barva ikony odvozena od typu použitého komunikačního ovladače, což usnadňuje identifikaci typu připojení [4, 3]. U kritických zařízení, jako jsou robotická pracoviště, barva reprezentuje aktuální provozní stav v reálném čase (např. zelená pro aktivní provoz, červená pro chybový stav, žlutá pro údržbu) [10]. Tato vizualizace je aktualizována v pravidelných intervalech bez nutnosti manuálního obnovení stránky.

6.2 Dashboard a vizualizace dat

Dashboard představuje detailní pohled na konkrétní průmyslové zařízení a je strukturován do několika logických sekcí pro maximální přehlednost.

- **Technické informace:** Úvodní sekce zobrazuje metadata uložená v databázi PostgreSQL, jako je název zařízení, komunikační kanál, IP adresa a síťový port.
- **Akční panel:** Umožňuje uživateli provádět operace nad zařízením. Pro běžné uživatele jde o refresh dat, zatímco administrátoři zde mají přístup k editačním funkcím.
- **Lokalizace:** Součástí dashboardu je vizualizace fyzického umístění v rámci haly a fotografie zařízení, což usnadňuje identifikaci stroje přímo na dílně.
- **Provozní data:** Klíčovou část tvoří dynamická tabulka aktuálních hodnot jednotlivých tagů, které jsou v reálném čase vyčítány ze systému KEPServerEX prostřednictvím protokolu OPC UA.

Tato kombinace statických metadat a živých procesních dat poskytuje operátorům i technologům komplexní přehled nezbytný pro rychlé rozhodování a efektivní diagnostiku.

7 Závěr

V rámci projektu byl úspěšně navržen a implementován systém pro monitoring průmyslových zařízení ve společnosti Doosan Bobcat EMEA s.r.o. Výsledná webová aplikace sjednocuje data z roztříštěných zdrojů, nahrazuje neefektivní tabulkové evidence a poskytuje vizuální přehled o výrobním prostředí v reálném čase.

Implementované řešení využívající standard OPC UA a moderní framework FastAPI prokázalo vysokou stabilitu a nízkou latenci. Díky kontejnerizaci pomocí Dockeru je aplikace snadno přenositelná a připravená na budoucí rozšiřování o další moduly, jako je prediktivní údržba nebo analytické reporty. Tento projekt tak tvoří pevný pilíř pro další fáze digitální transformace podniku.

Reference

[1] PTC Inc. *KEPServerEX OT Connectivity Platform*. Dostupné z: https://www.ptc.com/en/products/kepware/kepserverex. [cit. 2026-03-18]

[2] FOXON s.r.o. *OPC server pro Windows – KEPServerEX*. Dostupné z: https://kepware-opc.cz/produkty/opc-server-pro-windows-kepserverex/. [cit. 2026-03-18]

[3] A. Gilchrist, *Industry 4.0: The Industrial Internet of Things*. Berkeley, CA, USA: Apress, 2016.

[4] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld and M. Hoffmann, "Industry 4.0," Business & Information Systems Engineering, vol. 6, no. 4, pp. 239–242, 2014.

[5] A. S. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2007.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[7] R. Fielding and J. Reschke, *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*, RFC 7230, IETF, 2014. Dostupné z: https://datatracker.ietf.org/doc/html/rfc7230

[8] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, IETF, 2018. Dostupné z: https://datatracker.ietf.org/doc/html/rfc8446

[9] W. Mahnke, S.-H. Leitner and M. Damm, *OPC Unified Architecture*. Berlin, Germany: Springer, 2009.

[10] OPC Foundation, "OPC Unified Architecture Overview," 2023. Dostupné z: https://opcfoundation.org. [cit. 2026-03-25]

[11] FastAPI, *FastAPI Documentation*. Dostupné z: https://fastapi.tiangolo.com. [cit. 2026-03-25]

[12] Flask, *Flask Documentation*. Dostupné z: https://flask.palletsprojects.com. [cit. 2026-03-25]

[13] Django Software Foundation, *Django Documentation*. Dostupné z: https://www.djangoproject.com. [cit. 2026-03-25]

[14] PostgreSQL Global Development Group, *PostgreSQL Documentation*. Dostupné z: https://www.postgresql.org/docs/. [cit. 2026-03-25]

[15] Microsoft, *Lightweight Directory Access Protocol (LDAP)*. Dostupné z: https://learn.microsoft.com/en-us/windows/win32/adsi/ldap. [cit. 2026-03-25]

[16] IETF, *RFC 7519: JSON Web Token (JWT)*. Dostupné z: https://datatracker.ietf.org/doc/html/rfc7519. [cit. 2026-03-25]

[17] IETF, *RFC 6749: The OAuth 2.0 Authorization Framework*. Dostupné z: https://datatracker.ietf.org/doc/html/rfc6749. [cit. 2026-03-25]

[18] Docker Inc., *Docker Documentation*. Dostupné z: https://docs.docker.com/. [cit. 2026-03-25]

[19] IBM, "What is middleware?", Dostupné z: https://www.ibm.com/topics/middleware. [cit. 2026-04-15]

IDENTIFIKACE A ŘÍZENÍ HYDRAULICKÉHO SUBSYSTÉMU MECHATRONICKÉHO DÁVKOVAČE KAPALIN

Daniel Čech [1]

[1] ***Fakulta strojní, České vysoké učení technické v Praze (FS ČVUT)***

Abstrakt: Tento článek se zabývá analýzou, identifikací a návrhem řízení hydraulického subsystému automatického dávkovače kapalin. Komplexní systém je dekomponován na tři základní části, přičemž důraz je kladen na silně nelineární hydraulický obvod. Kvůli výraznému pásmu necitlivosti a asymetrickému setrvačnému doběhu membránového čerpadla byl sestaven matematicko-fyzikální model a provedena experimentální identifikace. Na základě simulační komparace zpětnovazebního PD regulátoru a dopředného řízení byla pro softwarovou implementaci zvolena dopředná kompenzace s využitím lineární interpolace po částech. Validace systému, podložená analýzou nejistot dle metodiky GUM, potvrdila dosažení požadované přesnosti dávkování s rozšířenou nejistotou max. $0,22$ *ml, což bezpečně splňuje toleranci* $\pm 1,0$ *ml.*
Klíčová slova: mechatronika, dávkovač kapalin, identifikace systému, dopředné řízení, GUM, Arduino

Abstract: This paper deals with the analysis, identification, and control design of a hydraulic subsystem for an automatic liquid dispenser. The complex system is decomposed into three main parts, with a primary focus on the highly non-linear hydraulic circuit. Due to the significant dead zone and asymmetric overrun of the diaphragm pump, a mathematical-physical model was formulated and an experimental identification was carried out. Based on a simulation comparison between a closed-loop PD controller and feedforward control, a feedforward compensation using piecewise linear interpolation was chosen for the software implementation. System validation, supported by a GUM uncertainty analysis, confirmed the achievement of the required dosing accuracy with an expanded uncertainty of up to 0.22 *ml, well within the* ± 1.0 *ml tolerance.*
Keywords: mechatronics, liquid dispenser, system identification, feedforward control, GUM, Arduino

1 Úvod

Tento článek vychází z návrhu a realizace prototypu automatizovaného nápojového dávkovacího systému s otočným ramenem. Zařízení je integrováno do šasi plastového modelu zemědělského stroje John Deere T670i. Systém využívá distribuovanou dvouvrstvou architekturu: nízkoúrovňové deterministické řízení v reálném čase obstarává mikrokontrolér Arduino UNO, zatímco nadřazenou vrstvu s dotykovým uživatelským rozhraním a databází zajišťuje jednodeskový počítač Raspberry Pi 4B. Hlavním cílem tohoto mechatronického stroje je nezávisle detekovat přítomnost sklenic na šesti plnicích pozicích a následně do nich s vysokou přesností automaticky dávkovat nastavený objem tekutiny v rozmezí 10 až 40 ml.

Pro dosažení spolehlivého chodu a požadované přesnosti dávkování je nezbytné analyzovat dynamiku celého zařízení. Z hlediska teorie automatického řízení lze komplexní systém navrženého dávkovače dekomponovat na tři základní, vzájemně interagující subsystémy:

1. **Hydraulický subsystém:** Představovaný membránovým čerpadlem, hadicovým vedením a dávkovací tryskou. Cílem je přesné objemové dávkování tekutiny.

2. **Mechanický subsystém:** Tvořený servopohonem pro přesné polohování ústí hadice. Vykazuje zanedbatelnou dynamiku, neboť servomotor obsahuje vlastní interní regulační smyčku polohy.

3. **Termodynamický subsystém:** Zahmujepřestuptepla uvnitřskříněs elektronikou. Je efektivně řešen zajištěním aktivníhoprouděnívzduchu s využitím dvoustavového spínání s hysterezí (spínací práh 40 °C, vypínací 30 °C), čímž je zamezeno nežádoucímu kmitání akčního členu.

Jelikož mechanický a termodynamický subsystém lze považovat za spolehlivě ošetřené, předmětem další analýzy a návrhu řízení je výhradně hydraulický subsystém [1], který je pro uživatelskou funkci – přesné dávkování v rozsahu10 až 40 ml s chybou do ±1, 0 ml – nejkritičtější.

2 Metody a identifikace modelu

2.1 Matematicko-fyzikální model

Systém se skládá ze dvou úzce provázaných fyzikálních oblastí. Pohon čerpadla zajišťuje stejnosměrný motor, jehož dynamikaje popsána diferenciální rovnicí obvodu kotvy a rovnicí mechanické rovnováhy [1]:

$$u(t) = R \cdot i(t) + L\frac{di(t)}{dt} + c_e\omega(t) \tag{1}$$

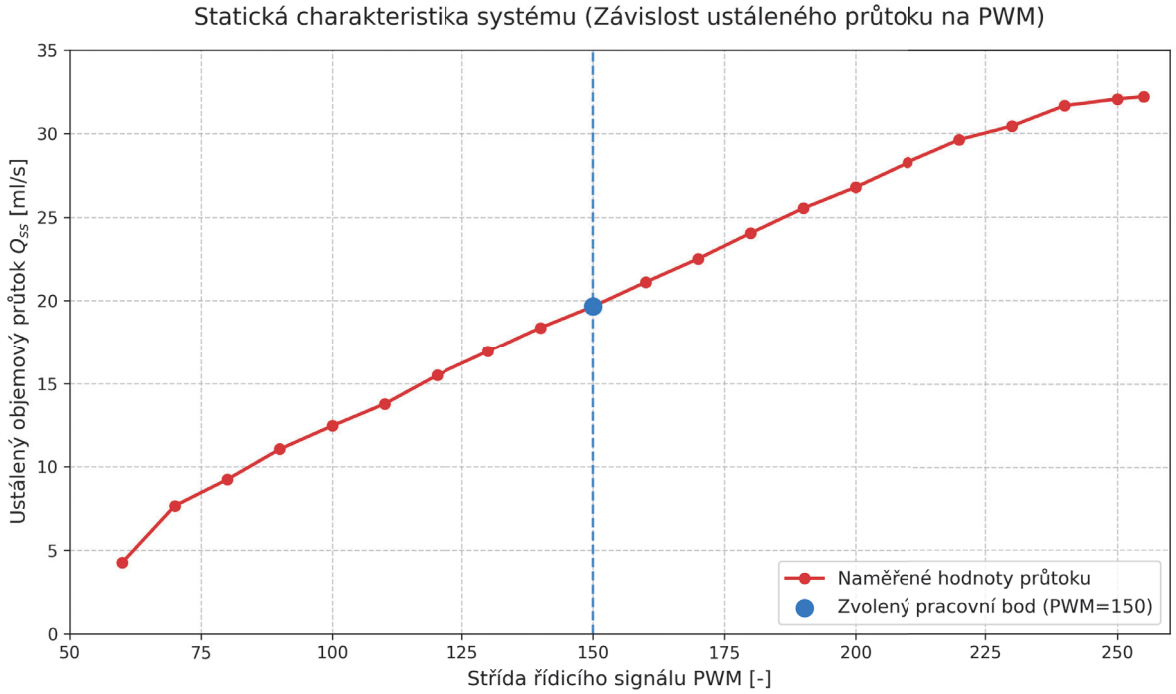
$$J\frac{d\omega(t)}{dt} = c_m i(t) - M_z(t) - B\omega(t) \tag{2}$$

Zátěžnýmoment$M_z(t)$ působícízpětněna motor je přímou funkcí tlakového spádu. Reálný průtok $Q(t)$ vychází z Bernoulliho rovnice a empirického modelu tlakových ztrát [2]:

$$\Delta p(t) = R_h \cdot Q(t)|Q(t)| + \rho g \Delta h \tag{3}$$

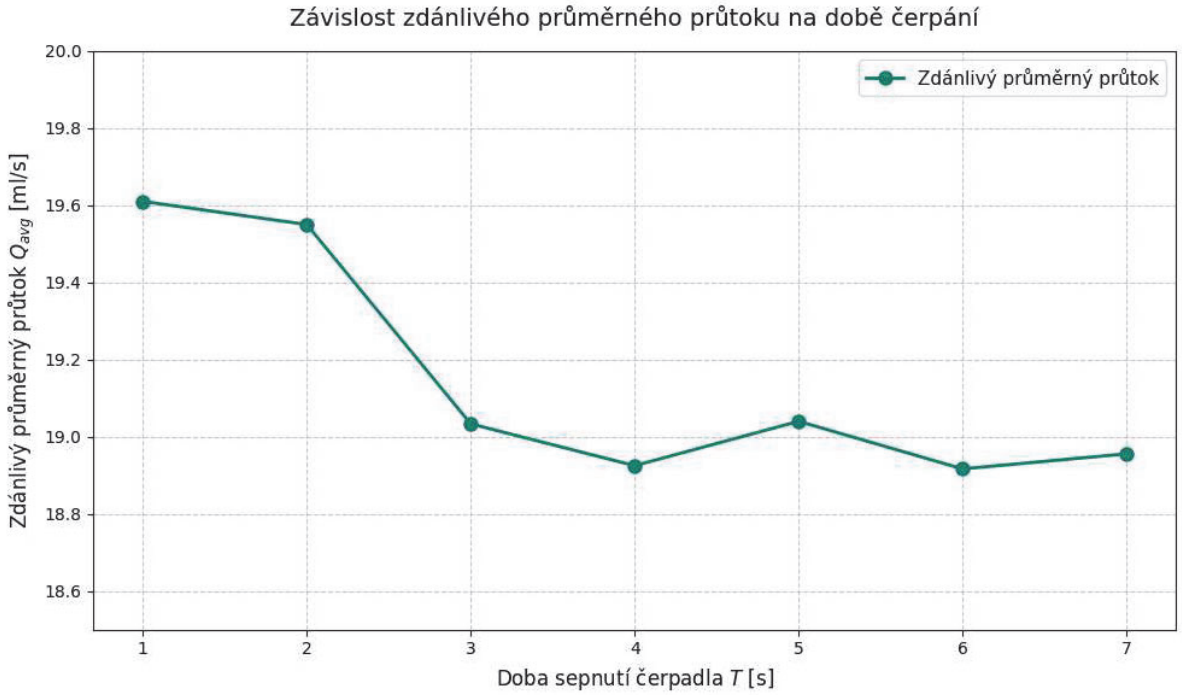
2.2 Experimentální identifikace

Systém byl podroben experimentální identifikaci [3], kde vstupem je střída PWM signálu $u(t) \in \langle 0, 255\rangle$ a výstupem průtok $Q(t)$. V intervalu PWM $\in \langle 0, 60\rangle$ byla identifikována mrtvá zóna (Obrázek 1). Jako nominální pracovní bod byl pro zajištění stabilního proudění bez pěnění zvolen výkon PWM = 150.



Obrázek1: Statická charakteristika čerpadla odhalující mrtvé pásmo v nízkých hodnotách buzení.

Zásadnímzjištěním byla asymetrie mezi fází náběhu a setrvačným doběhem (Obrázek 2). Zatímco ustálený průtok odpovídal hodnotě 19, 06 ml/s, při zkrácení doby čerpání vykazoval průměrný průtok prokazatelně vyšší



Obrázek 2: Závislost průměrného průtoku na době čerpání – projev asymetrie náběhu a doběhu.

hodnoty (až 19,61 ml/s pro dobu běhu 1 sekundu). Tento konstantní zbytkový objem, způsobený kinetickou energií kapaliny, by při lineárním řízení způsoboval systematické přelévání malých dávek.

Dynamika náběhu byla na základě analýzy vysokofrekvenčního videozáznamu (240 FPS) aproximována modelem 1. řádu s dopravním zpožděním:

$$G(s) = \frac{0,13}{0,15s + 1} e^{-0,1s} \tag{4}$$

3 Výsledky a návrh řízení

3.1 Simulace uzavřené regulační smyčky vs. dopředné řízení

Regulovanou veličinou je celkový objem $V(t)$, který je dán integrálem okamžitého průtoku:

$$V(t) = \int_0^t Q(\tau) d\tau \tag{5}$$

Z přítomnosti integrálu vyplývá, že systém vykazuje astatismus. Pokud by byl pro řízení aplikován standardní PID regulátor, vznikl by systém s astatismem 2. řádu, což by vedlo k překmitům. Vzhledem k nemožnosti odčerpání kapaliny zpět byl simulačně testován pouze PD regulátor s dopřednou kompenzací:

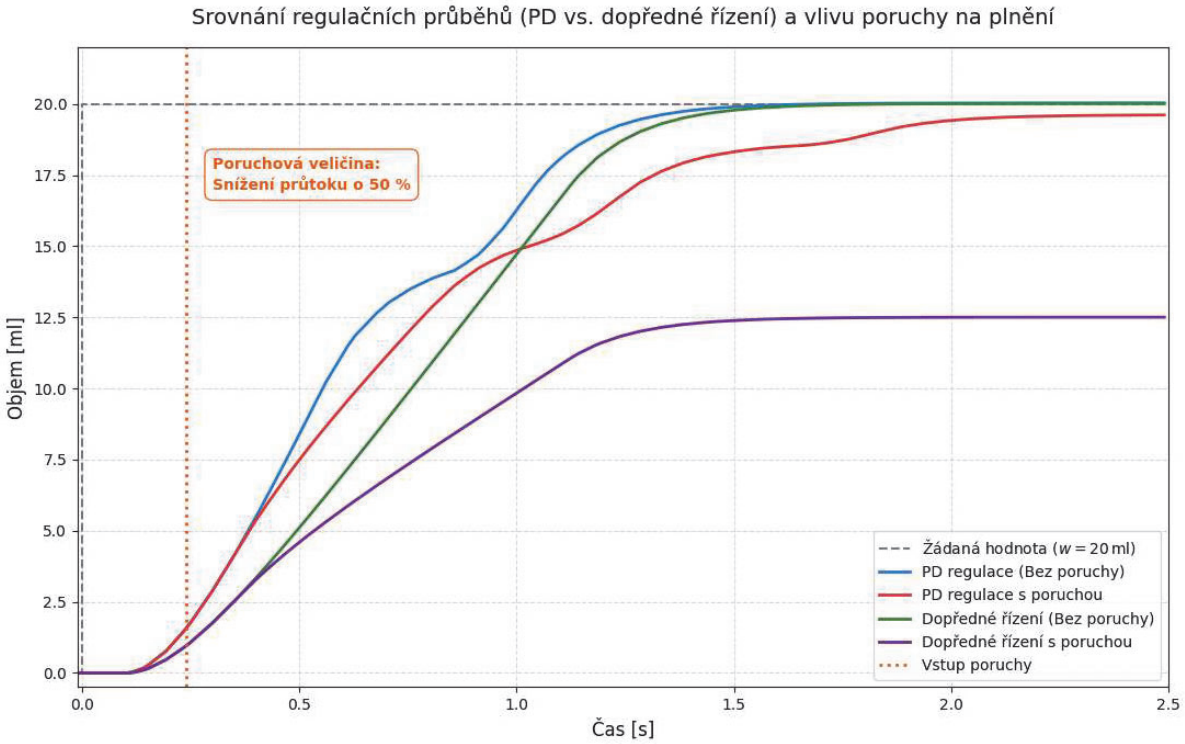
$$G_R(s) = K_p + K_d \frac{N \cdot s}{s + N} \tag{6}$$

Ačkoliv PD regulátor v simulacích vykazoval nulovou trvalou odchylku, jeho praktická realizace by vyžadovala nákladný snímač průtoku. Pro finální implementaci bylo proto zvoleno **dopředné řízení** s využitím 1-D kalibrační tabulky a po částech lineární interpolace (Obrázek 3). Tento přístup v sobě implicitně zohledňuje nelinearity i asymetrický doběh kapaliny:

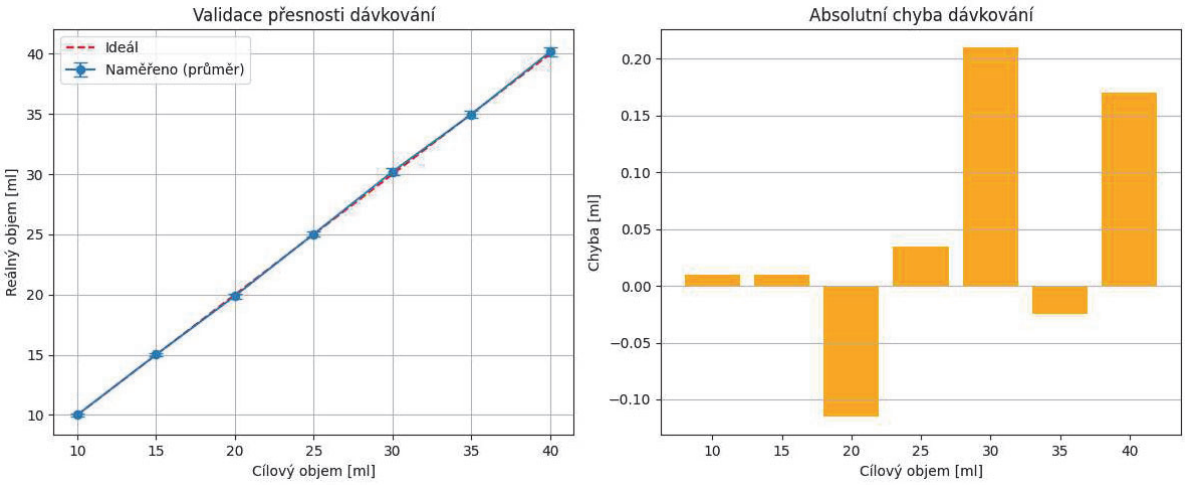
$$T_{pump} = T_1 + \frac{V_{ref} - V_1}{V_2 - V_1} \cdot (T_2 - T_1) \tag{7}$$

3.2 Validace a GUM analýza nejistot

Validace dopředného řízení na nezávislé sadě žádaných hodnot ($n = 20$ pro každou hladinu 10–40 ml) prokázala střední absolutní chybu MAE = 0,08 ml (viz Obrázek 4).



Obrázek3: Simulační porovnání uzavřené smyčky s PD řízením a dopředného řízení.



Obrázek4: Grafická prezentace validace přesnosti dávkování.

Pro exaktní verifikaci byla provedena analýza dle mezinárodní metodiky GUM [4]. Byla zahrnuta nejistota typu A (u_A) a tři složky typu B (u_B): digitální rozlišení váhy ($\pm 0,1$ g), ulpívání kapek na trysce ($\pm 0,05$ ml) a zjednodušením modelu hustoty referenčního média – vody (1 g = 1 ml) [5]. Výsledná rozšířená nejistota měření U (pro koeficient rozšíření $k = 2$) se v závislosti na objemu pohybovala v intervalu:

$$U = 2 \cdot \sqrt{u_A^2 + u_B^2} \in \langle 0,14; 0,22 \rangle \text{ ml} \tag{8}$$

Tabulka 1: Komplexní bilance nejistot měření pro vybrané validační hladiny

V_{ref} **[ml]**	u_A **[ml]**	u_B **[ml]**	u_c **[ml]**	$U(k = 2)$ **[ml]**
10	0,030	0,065	0,072	**0,14**
20	0,047	0,068	0,082	**0,16**
30	0,072	0,072	0,101	**0,20**
40	0,081	0,077	0,112	**0,22**

4 Závěr

Experimentální validace prokázala, že navržené empirické dopředné řízení splňuje s vysokou rezervou stanovený požadavek na maximální chybu dávkování $\pm 1,0$ ml v celém pracovním rozsahu. Rozšířená nejistota dosáhla maximálně 0,22 ml.

Ačkoliv absence zpětné vazby zvyšuje citlivost systému na vnější parametry – primárně na změnu viskozity vlivem teploty a na výkyvy napájecího napětí – představuje toto řešení pro danou aplikaci optimální kompromis mezi přesností a technologickou složitostí. Částečnou eliminaci citlivosti na viskozitu pomocí dynamické teplotní kompenzace lze uvažovat jako předmět budoucího vývoje.

Reference

[1] Norman S. Nise. *Control Systems Engineering*. Wiley, 8 edition, 2019.

[2] Yunus A. Cengel and John M. Cimbala. *Fluid Mechanics: Fundamentals and Applications*. McGraw-Hill, 4 edition, 2019.

[3] Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall, 2 edition, 1999.

[4] Joint Committee for Guides in Metrology (JCGM). *Evaluation of measurement data – Guide to the expression of uncertainty in measurement (GUM)*, first edition, 2008. JCGM 100:2008.

[5] Jan Leinveber and Pavel Vávra. *Strojnické tabulky: pomocná učebnice pro školy technického zaměření*. Albra, Úvaly, 6. doplněné vydání edition, 2017.

Moderní trendy v prediktivním řízení (MPC) a energetickém managementu: Od fyzikálních modelů k AI a digitálním dvojčatům

Pavel Hron

31. května 2026

Abstrakt

Model Predictive Control (MPC) se v posledním desetiletí etabloval jako přední strategie pro řízení složitých energetických systémů, včetně bateriových úložišť a rezidenčních mikrosítí. Tento článek přináší ucelený přehled současných výzkumných a průmyslových trendů v této oblasti. Analyzuje přechod od tradičního k ekonomickému prediktivnímu řízení (EMPC), integraci metod strojového učení v rámci Data-Driven MPC, propojování s bateriovými digitálními dvojčaty a implementaci stochastických přístupů pro efektivní zvládání systémových nejistot.

1 Úvod: Od klasické regulace k EMPC

Tradiční regulační smyčky a heuristické systémy řízené pevnými pravidly (Rule-Based Control) narážejí v moderní energetice na své limity. V prostředí s vysokou penetrací obnovitelných zdrojů, kde dochází k prudkým výkyvům ve výrobě a na trzích s elektřinou se objevují negativní spotové ceny, je nutné nasazovat systémy schopné proaktivního plánování.

Odpovědí na tyto výzvy je Model Predictive Control (MPC), optimalizační metoda, která využívá interní model k predikci budoucího chování systému a vypočítává sekvenci řídicích akcí sítě [1]. Tradiční MPC je ideální pro sledování pevně dané referenční trajektorie. Pokud je však cílem například komplexní energetická optimalizace, moderní praxe se přiklání k tzv. Economic Model Predictive Control (EMPC).

EMPC přímo integruje ekonomické cíle do účelové funkce. Systém neoptimalizuje pouze technický chod, ale napřímo maximalizuje finanční zisk (nebo minimalizuje provozní náklady) na základě dynamických spotových cen, distribučních poplatků a nákladů na opotřebení [2]. Obecná účelová funkce EMPC se obvykle zapisuje jako:

$$\min_{u} \sum_{k=0}^{H-1} \ell_e(x_k, u_k, p_k) \tag{1}$$

kde ℓ_e je ekonomická nákladová funkce, x_k je stav systému, u_k je řídicí akce (např. výkon baterie) a p_k reprezentuje časově proměnné externí parametry (ceny elektřiny, počasí) v horizontu H (prediction horizon). V praxi to znamená, že EMPC dokáže chování systému proaktivně plánovat. Algoritmus se tak může například rozhodnout přetopit budovu či nabít bateriové úložiště v době, kdy je k dispozici nadbytek levné solární energie (nebo

jsou spotové ceny na trhu velmi nízké či dokonce záporné). Takto naakumulovanou energii pak systém využije k překlenutí večerní cenové špičky, čímž napřímo minimalizuje celkové provozní náklady bez narušení uživatelského komfortu.

2 Nástup strojového učení: Data-Driven MPC

Klasický přístup k MPC (tzv. White-box modelování) vyžaduje přesné diferenciální rovnice popisující řízený systém. Sestavení a kalibrace takových termodynamických či elektrochemických modelů pro každou individuální budovu je však v masovém měřítku komerčně neudržitelná.

Tuto bariéru v současnosti překonává Data-Driven MPC (DDMPC). Místo řešení složitých fyzikálních rovnic využívá DDMPC modely strojového učení (Black-box nebo Grey-box), jako jsou Random Forest, XGBoost nebo hluboké neuronové sítě. Tyto algoritmy se učí dynamiku systému čistě na základě historických dat ze senzorů a elektroměrů [3].

Pokročilé studie z posledních let [4] navíc ukazují trend propojování DDMPC s vysvětlitelnou umělou inteligencí (XAI, např. SHAP hodnoty). To umožňuje nejen efektivně řídit spotřebu a výrobu, ale také vizualizovat, které faktory (např. venkovní teplota vs. cena elektřiny) měly na rozhodnutí algoritmu největší vliv. Výhodou DDMPC je snadná škálovatelnost a schopnost adaptace na měnící se zvyklosti spotřebitelů [5].

3 Digitální dvojčata a Health-Aware Control

Agresivní využívání baterií pro cenovou arbitráž přináší problém v podobě zrychlené degradace článků. Aby EMPC nepřinášelo krátkodobé zisky na úkor zničení hardwaru, integrují se do řídicích smyček Bateriová digitální dvojčata (Digital Twins).

Digitální dvojče je virtuální replika fyzického systému, která v reálném čase asimiluje data (napětí, proud, teplota) a pomocí pokročilých algoritmů simuluje vnitřní elektrochemické stavy, které nelze přímo změřit. Frameworky jako *PyBaMM* dnes umožňují rychlý běh zjednodušených, avšak vysoce přesných models (např. Single Particle Model - SPM), které predikují prostorové rozložení teploty a koncentraci iontů lithia [6].

Nejnovější vývoj směřuje k tzv. *Physics-Informed Neural Networks* (PINN) a hybridním dvojčatům, která kombinují fyzikální základy s neuronovými sítěmi pro zpřesnění odhadu stavu zdraví baterie (SoH) [7]. MPC regulátor propojený s takovým dvojčetem může dynamicky upravovat své limity (tzv. Health-Aware Control) a odmítnout lukrativní obchod na spotovém trhu, pokud by predikovaný nárůst vnitřní teploty článku vedl k neadekvátní degradaci a narušil tak celkovou ekonomiku vlastnictví.

4 Zvládání nejistot: Stochastické MPC

Všechny výše zmíněné prediktivní strategie jsou silně závislé na přesnosti vstupních dat. Deterministické EMPC předpokládá, že predikce solární výroby nebo spotřebitelského chování jsou absolutně přesné. Sebemenší odchylka v predikci však může vést k překročení limitů jističe a k obrovským finančním penalizacím.

Řešením, které se postupně dostává z akademického výzkumu do praxe, je Stochastic MPC (SMPC). Na rozdíl od deterministického přístupu modeluje SMPC budoucnost pravděpodobnostně. Pracuje s nejistotou pomocí tzv. *chance constraints* (pravděpodobnostních omezení) [8].

SMPC matematicky garantuje, že systémové limity budou dodrženy s předem definovanou pravděpodobností (např. 95 %). To umožňuje řídicímu systému najít optimální balanc mezi agresivním vytěžováním sítě pro maximální zisk a konzervativním přístupem zajišťujícím bezpečnost. Studie z Národní laboratoře pro obnovitelnou energii (NREL) prokazují, že v rámci HEMS (Home Energy Management Systems) poskytuje SMPC výrazně robustnější výsledky při poskytování služeb flexibility a odezvy na stranu poptávky (Demand Response) [8].

5 Závěr

Budoucnost řízení energetických úložišť a mikrosítí se posouvá z roviny hardwarové infrastruktury do sféry pokročilého softwaru. Fúze ekonomického Model Predictive Control s metodami datového modelování (XGBoost, LSTM) odstraňuje bariéru nutnosti sestavování složitých matematických modelů pro každou instalaci. Propojení s bateriovými digitálními dvojčaty následně zajišťuje, že tato umělá inteligence respektuje limity fyziky a maximalizuje nejen denní zisk, ale i dlouhodobou životnost aktiv.

Reference

[1] Ma, S., et al. (2022). *Model predictive control strategies in microgrids: A concise revisit.* Heriot-Watt University Research Portal.

[2] Zirkel-Kruse, F., et al. (2019). *Economic model predictive control of combined thermal and electric residential building energy systems.* Applied Energy, 240, 372-385.

[3] Ek, J. (2025). *Data-driven model predictive control for demand-side management of district-heated residential buildings.* Master's thesis, Aalto University, School of Science.

[4] Khelil, A., et al. (2026). *EcoImpact: energy conservation using data-driven model predictive control and interpretable machine learning in the buildings sector.* ResearchGate Preprint.

[5] Hempel, L., et al. (2025). *Outperforming the heating curve with data-driven model predictive control.* Heat Pumping Technologies Magazine, Vol. 43, No. 2.

[6] Zhu, T., et al. (2025). *Towards intelligent systems for battery management: A five-tier digital twin architecture.* arXiv:2509.02366.

[7] Kumar, A., et al. (2024). *Explainable data-driven digital twins for predicting battery states in electric vehicles.* IEEE Access.

[8] Zhuang, Y., et al. (2018). *Stochastic model predictive control for demand response in a home energy management system.* 2018 IEEE Power & Energy Society General Meeting (PESGM). (Dostupné také přes OSTI, ID: 1496049).

PIPELINE PRO AUTOMATIZOVANÉ NASAZENÍ MODELŮ Z PYTORCH NA FPGA KRIA KV260 S VITIS AI 3.5

Ondřej Pecina

Fakulta strojní, České vysoké učení technické v Praze (FS ČVUT)

Abstrakt: Nasazování hlubokých neuronových sítí na FPGA akcelerátory, jako je AMD Xilinx Kria KV260, je typicky křehký proces vázaný na konkrétní model, který těsně provazuje kvantizaci, kompilaci a ladění běhového prostředí, a nutí tak vývojáře přepisovat kód pro každou novou architekturu. Předkládám univerzální pipeline pro nasazení řízenou registry, která automatizuje celou cestu od checkpointu v PyTorch až po inferenci na desce s využitím Vitis AI 3.5 a akcelerátoru DPUCZDX8G. Ústředním návrhovým principem je oddělení aspektů modelu, datové sady a hardwaru do tří autoritativních registrů, díky čemuž může jediné jádro nezávislé na úloze — inspekce kompatibility s DPU, volitelné strukturní prořezávání, INT8 kvantizace, kompilace, přenos a běhové prostředí na desce — obsloužit klasifikaci obrazu i detekci objektů bez úprav kódu pro konkrétní model. Pipeline ve fázi hardwarově orientované přípravy vynucuje podmnožiny operátorů přívětivých pro DPU, nahrazuje nepodporované aktivace a izoluje postprocessing na straně CPU tak, aby složité dekodéry zůstaly numericky konzistentní od kalibrace na hostiteli až po spuštění na desce. Tok doplňuje vícevláknové běhové prostředí typu producent–konzument s profilováním po jednotlivých fázích, které zpřístupňuje běhové parametry — souběžnost instancí DPU, hloubku vstupní fronty a vektorizovanou cestu předzpracování. Výsledný framework představuje reprodukovatelný a rozšiřitelný vzor pro nasazování na FPGA, který podstatně snižuje inženýrské úsilí při přenosu nových modelů z PyTorch na edge hardware při zachování přesnosti a kompatibility s DPU.
Klíčová slova: FPGA, Kvantizace, Hardwarová akcelerace, Vitis AI, Pytorch

Abstract: Deploying deep neural networks on FPGA accelerators such as the AMD Xilinx Kria KV260 is typically a brittle, model-specific process that tightly couples quantization, compilation, and runtime tuning, forcing developers to rewrite code for every new architecture. I present a universal, registry-driven deployment pipeline that automates the complete path from a PyTorch checkpoint to on-board inference using Vitis AI 3.5 and the DPUCZDX8G accelerator. The central design principle is the separation of model, dataset, and hardware concerns into three authoritative registries, allowing a single task-agnostic core — DPU compatibility inspection, optional structural pruning, INT8 quantization, compilation, transfer, and the on-board runtime — to serve both image classification and object detection without model-specific code changes. In a hardware-aware preparation stage, the pipeline enforces DPU-friendly operator subsets, replaces unsupported activations, and isolates CPU-side post-processing so that complex decoders remain numerically consistent from host calibration through on-board execution. The flow is complemented by a multi-threaded, producer–consumer runtime with per-stage profiling that exposes runtime parameters — DPU instance concurrency, input-queue depth, and a vectorized preprocessing path. The resulting framework provides a reproducible and extensible blueprint for FPGA deployment that substantially reduces the engineering effort of porting new PyTorch models to edge hardware while preserving accuracy and DPU compatibility.
Keywords: FPGA, Quantization, Hardware Acceleration, Vitis AI, Pytorch

1 Úvod

Rozšíření hlubokých neuronových sítí ve vizuálních aplikacích zvýšilo poptávku po inferenci na okraji sítě (edge), kde omezení latence, spotřeby a ochrany soukromí dat často znemožňují přenést výpočet do cloudu. Programovatelná hradlová pole (FPGA) jsou pro toto nasazení atraktivní platformou: spojují energetickou účinnost vlastního datového toku s rekonfigurovatelností potřebnou pro sledování rychle se vyvíjejících architektur modelů. Deska AMD Xilinx Kria KV260, postavená kolem akcelerátoru DPUCZDX8G[1, 2], představuje typického zástupce této třídy levných edge akcelerátorů určených pro produkční nasazení. Praktická cena za dostání modelu na zařízení však zůstává vysoká.

V typickém pracovním postupu Vitis AI musí natrénovaná síť projít kontrolou kompatibility s DPU[3], INT8 kvantizací s reprezentativní kalibrací, hardwarově orientovanou přípravou grafu, kompilací do .xmodel specifického pro zařízení, přenosem na desku a nakonec spuštěním přes běhové prostředí, které kolem DPU organizuje preprocessing a postprocessing. Tyto fáze jsou těsně provázané a ve většině publikovaných postupů jsou spojeny kódem specifickým pro daný model. Každá nová architektura tak znovu vyžaduje značné inženýrské úsilí: je nutné vyhledat a nahradit nepodporované operátory, znovu rozdělit dekodéry mezi DPU a hostitelské CPU, udržet numerickou shodu mezi kalibrací a předzpracováním na desce a znovu doladit vláknové zpracování běhového prostředí. Výsledkem je proces nasazení, který je křehký, obtížně reprodukovatelný a těžko rozšiřitelný napříč úlohami.

Tento článek představuje univerzální pipeline pro nasazení řízenou registry, která toto oddělení činí explicitním. Metadata modelu, datové sady a hardwaru jsou zachycena ve třech autoritativních registrech, což umožňuje, aby jediné jádro nezávislé na úloze — inspekce, volitelné strukturní prořezávání, INT8 kvantizace, kompilace, přenos a běhové prostředí na desce — obsluhovalo heterogenní úlohy bez úprav kódu pro konkrétní model. Fáze hardwarově orientované přípravy vynucuje podmnožiny operátorů přívětivých pro DPU a izoluje postprocessing na straně CPU, takže složité dekodéry zůstávají numericky konzistentní od kvantizace na hostiteli až po spuštění na desce.

Obecnost frameworku demonstruji napříč klasifikací obrazu i detekcí objektů, přičemž detekce zahrnuje dvě zásadně odlišná paradigmata integrovaná za jediným rozhraním: detektor založený na kotvách (anchor-based), který zachovává surové víceúrovňové tenzory DPU s dekódováním a potlačením nemaxim (NMS) na CPU, a detektor bez kotev (anchor-free) typu end-to-end, který eliminuje NMS pomocí párování one-to-one a výběru top-k. Postup doplňuje vícevláknové běhové prostředí na desce typu producent–konzument s profilováním po jednotlivých fázích, jež zpřístupňuje běhové parametry — souběžnost DPU, hloubku vstupní fronty a vektorizovanou cestu předzpracování — které řídí kompromis mezi propustností a latencí na reálném hardwaru.

Přínosy této práce jsou:

- Registry-driven architecture, která odděluje aspekty modelu, datasetu a desky a umožňuje, aby jediné jádro nasazení cílilo na více úloh a modelů bez změn logiky jednotlivých fází.
- Metodika hardware-aware preparation, která systematicky vynucuje omezení operátorů DPU a rozděluje post-processing mezi DPU a ARM CPU při zachování numerické shody v celém postupu.
- Rozšiřitelné, instrumentované běhové prostředí na desce, které integruje heterogenní detekční paradigmata za jediným rozhraním a zpřístupňuje laditelné parametry souběžnosti a předzpracování.
- Komplexní vyhodnocení na desce Kria KV260, jež vykazuje reprodukovatelnou kompilaci a spuštění napříč úlohami spolu s charakterizací propustnosti, latence DPU, spotřeby a energie na snímek.

2 Vitis AI nástroje

2.1 Tok nástrojů Vitis AI a akcelerátor DPU

Vitis AI 3.5 je vývojový zásobník společnosti AMD pro inferenci neuronových sítí na hardwaru Xilinx[3]. Pro edge zařízení jako Kria KV260 je výpočet realizován jednotkou DPU (Deep-learning Processing Unit) – v našem případě konfigurací DPUCZDX8G_ISA1_B4096 běžící na 300 MHz[2]. DPU je programovatelný akcelerátor s pevně danou instrukční sadou, který vykonává kvantované celočíselné (INT8) operace. Tok nasazení sestává z několika fází: inspekce kompatibility grafu s DPU, kvantizace s kalibrací na reprezentativních datech, kompilace do mezireprezentace XIR a následně do souboru `.xmodel` specifického pro cílovou architekturu, a konečně běhové spuštění prostřednictvím knihovny VART (Vitis AI Runtime). Pro analýzu kompatibility a kvantizaci poskytuje knihovna `pytorch_nndct` rozhraní `Inspector` a `torch_quantizer`.

2.2 INT8 kvantizace a omezení operátorů DPU

DPU vykonává pouze podmnožinu operací podporovaných běžnými frameworky a pracuje výhradně v celočíselné aritmetice. To přináší dvě zásadní omezení. Za prvé, kvantizace po trénování (PTQ) převádí váhy a aktivace na INT8 pomocí kalibračních statistik; nesprávně zarovnaná normalizace mezi kalibrací na hostiteli a předzpracováním na desce vede k degradaci přesnosti[4, 5]. Za druhé, ne všechny operátory jsou podporovány nebo efektivně mapovány na DPU. Konvoluce, dávková normalizace (BatchNorm), ReLU, pooling a concatenation patří mezi spolehlivě podporované; naopak attention mechanismy, softmax, dynamický reshape, operace náročné na permutace a některé fúze aktivací (například depthwise konvoluce s LeakyReLU) jsou problematické nebo nepodporované. Operace, které DPU nezvládá, musí být buď nahrazeny ekvivalenty přívětivými pro DPU, nebo přesunuty na ARM CPU jako součást postprocessing.

2.3 Adaptace modelů přívětivá pro DPU

Důsledkem těchto omezení je, že modely navržené pro GPU často nelze nasadit na DPU bez úprav. Běžné strategie zahrnují nahrazení nepodporovaných aktivací (například SiLU za LeakyReLU nebo ReLU), odstranění attention bloků a rozdělení výpočtu mezi DPU (páteř sítě a hlavy) a CPU (dekódování, NMS, sestavení masek). U detektorů objektů je to obzvláště výrazné: kotevní dekódování a NMS zahrnují řazení, indexování a podmíněnou logiku, které jsou pro DPU nevhodné, a proto se typicky ponechávají na CPU[6].

2.4 Související nástroje pro nasazení

AMD poskytuje Model Zoo s předkompilovanými modely a referenčními příklady[3], které však pokrývají fixní sadu architektur a předpokládají, že uživatel pro nové modely převezme a upraví příklady specifické pro daný model. Existující komunitní a vendorové postupy obvykle váží logiku nasazení přímo na konkrétní model, čímž vzniká duplicita kódu a křehkost při přechodu na novou architekturu. Akademické práce o nasazení CNN na FPGA se často zaměřují buď na návrh hardwarového akcelerátoru, nebo na optimalizaci jediného modelu, méně pozornosti však věnují softwarově-inženýrské reprodukovatelnosti a rozšiřitelnosti celého toku napříč více úlohami[7]. Chci tuto mezeru zaplnit: namísto optimalizace jediného modelu navrhujeme architekturu řízenou registry, která odděluje aspekty modelu, datové sady a hardwaru, a umožňuje tak jedinému jádru nasazení obsluhovat heterogenní úlohy bez úprav kódu pro konkrétní model.

3 Architektura systému

Navrhovaná pipeline je postavena na jediném návrhovém principu: oddělení aspektů toho, *co* se nasazuje, *jaká data* se přitom používají a *kde* výsledek běží. Tento princip se promítá do tří autoritativních registrů a do jádra nezávislého na úloze, které tyto registry konzumuje. V této kapitole popisuji návrhové cíle, datové registry, fáze toku a mechanismus hardwarově orientované přípravy modelu.

3.1 Návrhové cíle

Architekturu řídí tři cíle. Prvním je *nezávislost na úloze*: tytéž fáze inspekce, kvantizace a kompilace musí obsloužit klasifikaci i detekci bez větvení logiky podle konkrétního modelu. Druhým je *reprodukovatelnost*: každý běh musí být plně určen deklarativní konfigurací, takže výsledné artefakty lze odvodit z příkazu a obsahu registrů. Třetím je *rozšiřitelnost*: přidání nového modelu, datové sady nebo cílové desky má vyžadovat pouze záznam v registru, nikoli změnu kódu jednotlivých fází.

3.2 Registry řízené metadaty

Tři kořenové konfigurační soubory tvoří jediný zdroj pravdy. Registr modelů (`model_config.py`) zachycuje typ úlohy, název modelu, zavaděč zdrojového grafu, vstupní tvar, cestu k checkpointu, výpočetní náročnost v GOPs a u detektorů též metadata dekodéru (počet tříd, kroky, kotvy nebo typ bezkotevního dekodéru, prahy, `reg_max` a `max_det`). Registr datových sad (`dataset_config.py`) zachycuje kalibrační a obrazové cesty, názvy složek, uspořádané popisky tříd a normalizační konstanty. Registr desky (`board_config.py`) zachycuje cestu k architektuře DPU, IP adresu a uživatele desky, špičkový výkon DPU v GOPS, počet aktivních vláken a pomocné funkce pro telemetrii spotřeby.

Toto oddělení je klíčové: skripty jednotlivých fází nikdy nezakódovávají konstanty modelu, datové sady ani desky natvrdo, nýbrž je čtou z registrů. V důsledku toho se změna cílového modelu nebo desky provede úpravou

deklarativních metadat, a nikoli logiky toku, čímž se eliminuje třída chyb plynoucích z roztroušených, vzájemně nekonzistentních konstant.

3.3 Jádro nezávislé na úloze

Sdílené jádro orchestruje šest fází. Fáze *inspekce* ověří kompatibilitu grafu s DPU a vytvoří zprávu identifikující operace přiřazené CPU. Volitelná fáze *prořezávání* provádí strukturní redukci modelu[8, 9]. Fáze *kvantizace* provede INT8 kalibraci na reprezentativní podmnožině dat a exportuje kvantovaný graf[4]. Fáze *kompilace* přeloží kvantovaný graf přes mezireprezentaci XIR do souboru `.xmodel` specifického pro architekturu desky. Volitelná fáze *přenosu* přesune na desku pouze soubory potřebné pro aktivní úlohu. Konečně *běhové prostředí* na desce vykoná inferenci. Orchestrátor volí task-specifické zavaděče a běhové programy na základě typu úlohy uvedeného v registru modelů, takže rozšíření o nový model nevyžaduje zásah do řídicí logiky.

3.4 Hardwarově orientovaná příprava modelu

Mezi zdrojovým grafem v PyTorch a kvantizací leží fáze přípravy nezávislá na úloze, jejímž úkolem je uvést model do podoby přívětivé pro DPU. Tato fáze plní tři role[10]. Zaprvé *vynucuje podmnožinu podporovaných operátorů* – nahrazuje nepodporované aktivace a odstraňuje konstrukce, které DPU efektivně nemapuje, jako jsou attention bloky. Zadruhé *rozděluje postprocessing* mezi DPU a ARM CPU: páteř sítě a hlavy běží na DPU, zatímco operace náročné na řazení a indexování, jako je dekódování detekcí a NMS, zůstávají na CPU. Zatřetí *udržuje numerickou shodu*, neboť normalizační konstanty z registru datových sad používá identicky při kalibraci na hostiteli i při předzpracování na desce, čímž se předchází degradaci přesnosti při kvantizaci.

Tato fáze je záměrně parametrizovaná typem úlohy a metadaty dekodéru z registru. Tentýž mechanismus přípravy tak obslouží různé architektury: rozhoduje na základě deklarovaných metadat, nikoli na základě napevno zakódovaných předpokladů o konkrétním modelu. Tím je zachována nezávislost jádra na úloze i v nejcitlivějším místě celého toku.

4 Implementace a rozšiřitelnost

V této kapitole ukazuji, jak se návrhové principy z kapitoly 3 projevují v praxi. Nejprve popisuji, jak se nový model přidává výhradně přes registr, poté na případové studii detekce demonstrujeme schopnost jediného rozhraní obsloužit zásadně odlišná paradigmata, a nakonec popisujeme instrumentované běhové prostředí na desce.

4.1 Přidání modelu přes registr

Rozšíření pipeline o nový model spočívá v zápisu jediného záznamu do registru modelů, který deklaruje typ úlohy, zavaděč zdrojového grafu, vstupní tvar a cestu k checkpointu[10]. U vlastních modelů záznam dále uvádí pole `source: "custom"` spolu s odpovídajícími cestami k souboru, YAML konfiguraci nebo repozitáři, podle nichž zavaděč v modulu pro práci s modely sestaví graf. Žádná fáze toku – inspekce, kvantizace ani kompilace – se přitom nemění; jádro odvodí veškeré chování z deklarovaných metadat. Tím se přidání modelu redukuje z úpravy kódu napříč fázemi na lokální, deklarativní změnu.

4.2 Heterogenní detekce za jediným rozhraním

Sílu oddělení modelu od toku ilustruji integrací dvou detektorů založených na odlišných paradigmatech, jež však sdílejí tutéž infrastrukturu nasazení.

Model YOLOv5 představuje detekci založenou na kotvách (anchor-based)[11]. Jeho detekční hlava je na hostiteli ponechána ve formě surových víceúrovňových výstupů P3/P4/P5, zatímco kotevní dekódování a NMS probíhají na ARM CPU. Aktivace SiLU jsou nahrazeny ReLU, aby byl graf přívětivý pro DPU. Metadata dekodéru – kroky, kotvy a prahy – jsou uvedena v registru a sdílena mezi kvantizačním testem a běhovým programem na desce.

Model YOLOv26 naproti tomu představuje bezkotevní (anchor-free) detekci typu end-to-end[12]. Pomocí obalu nad architekturou Ultralytics se na DPU vystaví pouze větev one2one, čímž se eliminuje potřeba NMS; výběr detekcí se provádí top-k selekcí přímo v INT8 prostoru. Nepodporované moduly (C2PSA, PSABlock, Attention) jsou z grafu cíleného na DPU vyloučeny a aktivace SiLU jsou nahrazeny ReLU.

Klíčové je, že obě architektury využívají tytéž fáze inspekce, kvantizace a kompilace; liší se pouze deklarovanými metadaty dekodéru a logikou postprocessing na CPU. Tatáž pipeline tak bez úprav řídicího kódu obslouží detektor s kotvami i bez nich, čímž se prakticky potvrzuje nezávislost jádra na úloze.

4.3 Instrumentované běhové prostředí na desce

Běhové prostředí na desce je navrženo jako vícevláknová architektura typu producent–konzument. Producentská vlákna provádějí předzpracování na CPU (změna velikosti, letterboxing, normalizace) a vkládají snímky do fronty, zatímco konzumentská vlákna asynchronně spouštějí inferenci na DPU přes VART. Tím se překrývá CPU předzpracování s výpočtem na DPU a maximalizuje se propustnost.

Běhový program zpřístupňuje několik laditelných parametrů, jež řídí kompromis mezi propustností a latencí: počet souběžných běhových instancí DPU (`--threads`), počet producentských vláken (`--producers`), hloubku vstupní fronty (`--queue-size`) a přepínače pro vypnutí vykreslování a ukládání výstupů (`--no-draw`, `--no-save`). Předzpracování normalizace je urychleno vektorizovanou vyhledávací tabulkou (LUT): pokud jsou normalizační koeficienty všech tří kanálů shodné, aplikuje se tabulka jediným voláním `np.take` nad celým snímkem, čímž se výrazně sníží náklady na normalizaci v horké smyčce.

Pro analýzu výkonu obsahuje běhový program rozsáhlé profilování po jednotlivých fázích (`--profile`, `--profile-json`). Pro každý snímek se zaznamenávají dílčí latence od připravenosti předzpracování, přes čekání ve vstupní a výstupní frontě, až po připravenost výsledku a celkový výstup. Tyto metriky umožňují odhalit úzká místa – například hladovění konzumentů či nárůst zpoždění ve frontě – a kvantifikovat dopad souběžnosti DPU.

5 Diskuse

5.1 Získané poznatky

Zkušenost s návrhem pipeline potvrzuje, že hlavní zdroj tření při nasazování na DPU není samotná kvantizace či kompilace, nýbrž roztroušené předpoklady specifické pro konkrétní model, datovou sadu a desku. Soustředění těchto předpokladů do tří registrů přineslo velký přínos vůči vynaloženému úsilí: jakmile jsou metadata deklarativní, stává se přidání modelu lokální změnou a fáze toku zůstávají stabilní. Druhým poznatkem je, že numerická shoda mezi kalibrací na hostiteli a předzpracováním na desce je křehká a musí být vynucena sdílením týchž normalizačních konstant z registru, nikoli jejich nezávislým definováním na obou stranách. Třetím poznatkem je, že hranice mezi DPU a CPU je u detektorů přirozeným návrhovým rozhraním: ponechání dekódování a NMS na CPU umožnilo integrovat kotevní i bezkotevní paradigma beze změny řídicí logiky.

5.2 Omezení

Pipeline má v současné podobě několik omezení. Podpora klasifikace a detekce je stabilní a ověřená na hardwaru, avšak sémantická a instanční segmentace zůstávají ve stavu rozpracovanosti: existuje registrace modelů a kostra běhového programu, ale cesta vyhodnocení přesnosti není dosud dokončena. Obdobně je strukturní prořezávání pomocí nástroje Vitis AI Optimizer přítomno, smyčky pro obnovu přesnosti po prořezání se však stále ladí. Dále je pipeline cílena na konkrétní konfiguraci DPU desky Kria KV260; přenositelnost na jiné architektury DPU je sice díky registru desky koncepčně přímočará, nebyla však v této práci experimentálně ověřena.

5.3 Hrozby pro validitu

Z hlediska měření výkonu je relevantní, že běhové prostředí je založeno na Python rozhraní VART a vektorizovaném zpracování v NumPy. Přesné hardwarové časování přenosů DMA a výpočtu uvnitř DPU proto není od režie Python vrstvy plně oddělitelné; vykazované latence DPU zahrnují i náklady na volání API. Měřená propustnost a latence rovněž závisí na počtu vláken, hloubce fronty a počtu producentů, takže jednotlivá čísla je nutno interpretovat vždy v kontextu konkrétní konfigurace běhu. Konečně reprodukovatelnost kompilace byla ověřena na omezené podmnožině modelů a datových sad; zobecnění na širší spektrum architektur je předmětem dalšího ověřování.

6 Závěr a budoucí práce

V této práci jsem představil univerzální pipeline pro nasazení modelů z PyTorch na FPGA akcelerátor Kria KV260 s využitím Vitis AI 3.5. Ústředním přínosem je architektura řízená registry, která odděluje aspekty modelu, datové sady a hardwaru do tří autoritativních konfigurací a umožňuje, aby jediné jádro nezávislé na úloze – inspekce, volitelné prořezávání, INT8 kvantizace, kompilace, přenos a běhové prostředí na desce – obsloužilo klasifikaci i detekci bez úprav kódu pro konkrétní model. Ukázal jsem, že hardwarově orientovaná příprava grafu dokáže systematicky vynutit omezení operátorů DPU a rozdělit postprocessing mezi DPU a ARM CPU při zachování numerické shody napříč celým tokem. Rozšiřitelnost frameworku jsem doložili integrací dvou zásadně

odlišných detekčních paradigmat – kotevního a bezkotevního – za jediným rozhraním a doplnili ji instrumentovaným, vícevláknovým běhovým prostředím s laditelnými parametry souběžnosti a profilováním po jednotlivých fázích.

Výsledkem je reprodukovatelný a rozšiřitelný vzor pro nasazování na okraji sítě, který podstatně snižuje inženýrské úsilí při přenosu nových modelů z PyTorch na FPGA hardware, aniž by obětoval přesnost či kompatibilitu s DPU.

V budoucí práci se zaměřím na dokončení cesty vyhodnocení přesnosti pro sémantickou a instanční segmentaci, dotažení smyček pro obnovu přesnosti po strukturním prořezání a na rozšíření podpory o další architektury DPU prostřednictvím registru desky. Dále plánuji přesnější oddělení hardwarového časování DPU od režie běhové vrstvy a širší experimentální ověření reprodukovatelnosti kompilace napříč rozsáhlejší sadou modelů.

Reference

[1] Advanced Micro Devices, Inc. (AMD/Xilinx). *Kria KV260 Vision AI Starter Kit User Guide (UG1089)*, 2022. [Online; cit. 2026-06-15].

[2] Advanced Micro Devices, Inc. (AMD/Xilinx). *DPUCZDX8G for Zynq UltraScale+ MPSoCs Product Guide (PG338)*, 2023. Version 4.1. [Online; cit. 2026-06-15].

[3] Advanced Micro Devices, Inc. (AMD/Xilinx). *Vitis AI User Guide (UG1414)*, 2023. Version 3.5. [Online; cit. 2026-06-15].

[4] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2704–2713, 2018.

[5] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. https://arxiv.org/abs/1806.08342, 2018. arXiv:1806.08342 [cs.LG]. [Online; cit. 2026-06-15].

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[7] Kaiyuan Guo, Shulin Zeng, Jincheng Yu, Yu Wang, and Huazhong Yang. [DL] a survey of FPGA-based neural network inference accelerators. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 12(1):1–26, 2019.

[8] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015.

[9] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient ConvNets. In *International Conference on Learning Representations (ICLR)*, 2017.

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

[11] Glenn Jocher and Ultralytics. YOLOv5 by ultralytics. https://github.com/ultralytics/yolov5, 2020. [Online; GitHub repository, cit. 2026-06-15].

[12] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO. https://github.com/ultralytics/ultralytics, 2023. [Online; GitHub repository, cit. 2026-06-15].

REŠERŠE MODELŮ KOROZIVITY ATMOSFÉRY A MODELŮ STROJOVÉHO UČENÍ PRO ÚČELY PREDIKCE KOROZE HLINÍKOVÝCH KONSTRUKCÍ

Matěj Mansfeld

České vysoké učení technické v Praze
Fakulta strojní
mansmat@cvut.cz

Abstrakt: Tato práce se zabývá rešerší modelů používaných pro popis a predikci atmosférické koroze materiálů se zaměřením na klasické korozní modely a moderní metody strojového učení. V první části jsou popsány vybrané modely korozivity atmosféry, včetně jejich principů, výhod a omezení. Druhá část práce je věnována metodám strojového učení využitelným pro analýzu a predikci korozního chování na základě experimentálních a environmentálních dat. Pozornost je zaměřena zejména na metody redukce dimenzionality dat a predikční algoritmy, jako jsou Random Forest, Support Vector Regression, Gradient Boosting, umělé neuronové sítě, LSTM a ARIMA.
Práce je součástí projektu zaměřeného na predikci míry atmosférické koroze z naměřených dat pomocí metod strojového učení. Cílem projektu je analyzovat závislosti mezi parametry prostředí a korozním úbytkem materiálu a navrhnout model schopný přesné predikce korozního chování v reálných podmínkách. Výsledky rešerše slouží jako teoretický základ pro návrh a implementaci predikčního systému.
Klíčová slova: koroze, strojové učení, predikce

Abstract: This thesis presents a review of models used for the description and prediction of atmospheric corrosion of materials, with a focus on both classical corrosion models and modern machine learning approaches. The first part describes selected models of atmospheric corrosivity, including their principles, advantages, and limitations. The second part is devoted to machine learning methods applicable to the analysis and prediction of corrosion behavior based on experimental and environmental data. Particular attention is paid to dimensionality reduction techniques and predictive algorithms such as Random Forest, Support Vector Regression, Gradient Boosting, Artificial Neural Networks, LSTM, and ARIMA.
The work is part of a project focused on predicting the rate of atmospheric corrosion from measured data using machine learning methods. The aim of the project is to analyze relationships between environmental parameters and corrosion loss of materials and to develop a model capable of accurately predicting corrosion behavior under real operating conditions. The results of this review serve as the theoretical foundation for the design and implementation of the predictive system.
Keywords: corrosion, machine learning, prediction

1 Modely korozivity atmosféry

1.1 Heuristické modely

Heuristické modely představují empiricky založené přístupy, které využívají experimentální data k identifikaci vzorů chování vedoucích k predikci průběhu atmosférické koroze. Tyto modely jsou zpravidla vytvářeny na základě statistických korelací mezi environmentálními parametry a míry atmosferické koroze. Výhodou heuristických modelů je jejich relativně jednoduchá implementace a nízká výpočetní náročnost. Nevýhodou je však skutečnost, že získané vztahy bývají platné pouze pro specifické materiály, prostředí a experimentální podmínky. Modely navíc neposkytují hlubší vhled do fyzikálních a elektrochemických mechanismů koroze. Výsledky mohou být citlivé i na malé změny okolních podmínek, což omezuje jejich zobecnitelnost a využitelnost v praxi.[1]

1.1.1 Norma ISO 9223:2012

Tato norma se zaměřuje na korozivitu kovů a slitin a korozivitu atmosféry. Zejména jejich klasifikaci, určení a odhad. V části 8.2 "Normative corrosivity estimation based on calculated first-year corrosion losses" rovnice č. 4 popisuje úbytek materiálu strukturních konstrukcí z hliníku za první rok na základě průměrné roční depozice SO_2 a Cl^-, průměrné roční teploty a průměrné roční relativní vlhkosti. Rovnice nebere v potaz tvar konstrukce ani jiné podmínky[2]

Rovnice vypadá následovně:

$$r_{corr} = 0,0042P_d^{0,73}e^{(0,025RH+f_{Al})} + 0,0018S_d^{0,60}e^{(0,02RH+0,094T)}$$

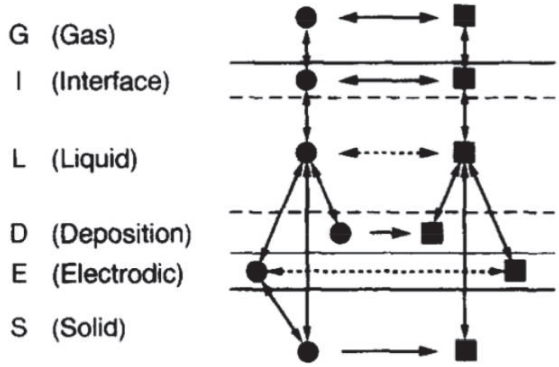
, kde
r_{corr} je přírůstek koroze/úbytek konstrukce za první rok $[\mu m]$,
T je průměrná teplota za tento rok $[°C]$,
RH je průměrná relativní vlhkost během roku [%],
P_d je průměrná depozice SO_2 za den během roku $[mg/m^2d]$,
S_d je průměrná depozice Cl^- za den během roku $[mg/m^2d]$.

1.2 Deterministické vícestupňové modely

Deterministické modely jsou založeny na matematickém a elektrochemickém popisu jednotlivých dílčích procesů atmosférické koroze. Tyto modely umožňují postupné rozšiřování o další fyzikální a chemické děje, například transport iontů, tvorbu korozních produktů nebo difuzní procesy. Díky tomu lze dosáhnout vyšší přesnosti simulací, avšak za cenu zvýšené výpočetní náročnosti.[1]

1.2.1 GILDES

Model GILDES se používá pro popis lokalizované atmosférické koroze kovů v přítomnosti tenkých vrstev elektrolytu. Zahrnuje kinetiku korodujícího systému včetně interakcí mezi plynným a kapalným prostředím. Model je rozdělen do šesti vrstev: plyn – rozhraní – kapalina – usazenina – elektroda – pevný materiál. Každá z těchto vrstev je popsána samostatnými diferenciálními rovnicemi, které modelují transport látek, chemické reakce a elektrochemické procesy. Model GILDES tak poskytuje detailní fyzikální reprezentaci korozního systému. Nevýhodou modelu je skutečnost, že nezohledňuje například vliv geometrie konstrukce nebo lokální nehomogenity materiálu.[1]



Obr. 1: Schéma modelu GILDES [3]

1.2.2 MITReM

MITReM je matematický model určený pro simulaci transportu iontů a chemických reakcí v elektrochemických systémech. V oblasti atmosférické koroze se využívá zejména pro modelování procesů probíhajících v tenkých vrstvách vlhkosti na povrchu kovů. Model umožňuje detailní analýzu změn koncentrací iontů, pH prostředí a tvorby korozních produktů v čase. Přestože poskytuje velmi přesné výsledky, bývá jeho použití v průmyslových aplikacích omezeno vysokou výpočetní náročností, zejména při trojrozměrných simulacích.[1]

1.2.3 Celulární automat

Celulámí automat (CA) je diskrétní matematický model složený z pravidelně uspořádaných buněk, jejichž stav se mění v jednotlivých časových krocích podle předem definovaných pravidel. Tento přístup je vhodný pro simulaci složitých systémů založených na lokálních interakcích.[1]

J. Saunier a jeho tým definovali čtyři možné stavy buněk pro využití CA při simulaci koroze: rozpouštědlo (S), kov (M), produkt koroze (L) a chodec (W). Chodci reprezentují ionty kovů difundující skrze vrstvy korozních produktů.Výhodou celulárních automatů je schopnost modelovat lokální šíření koroze a vznik korozních defektů s relativněnízkýmivýpočetníminároky Chemická podstata procesu je však do značné míry nahrazena statistickými pravidly přeměny buněk.[1]

2 Modely strojového učení

Metody strojového učení nacházejí v oblasti predikce korozivity atmosféry stále širší uplatnění. Jejich hlavní výhodou je schopnost zpracovávat rozsáhlé datasety a identifikovat poměrně jednoduše i nelineární vztahy mezi environmentálními parametry a korozním poškozením materiálů.

2.1 Algoritmy pro snížení dimenzionality vstupních dat

V případě rozsáhlých datasetů může velké množství atributů vést k nadměrné výpočetní náročnosti, přeučení modelu nebo snížení interpretovatelnosti výsledků. Proto se využívají algoritmy pro redukci dimenzionality dat, jejichž cílem je zachovat co největší množství relevantních informací při současném omezení redundantních nebo irelevantních atributů.

2.1.1 Koeficient maximální informace (MIC)

MIC je statistická metoda sloužící ke zjišťování závislostí mezi dvěma proměnnými. Na rozdíl od klasických korelačních metod umožňuje efektivně detekovat také nelineární vztahy mezi atributy. Hodnota MIC se pohybuje v intervalu $[0;1]$, kde vyšší hodnota znamená silnější závislost mezi proměnnými.[4]

Společná informace **I** mezi atributy **x, y** se počítá následující rovnicí:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{1}$$

, kde **p** je neurčitost

Hodnotakoeficientu maximální informace je pak vyjádřena normalizací na interval $[0;1]$

$$MIC(D) = max_{x,y} \left\{ \frac{I^*(D,x,y)}{log(min(x,y))} \right\} \tag{2}$$

, kde **D** je dataset

2.1.2 Analýza hlavních komponent (PCA)

PCA patří mezi běžné techniky pro předzpracování dat. Metoda transformuje původní proměnné na novou množinuvzájemněnezávislých hlavních komponent, které zachovávají co největší část variability dat. Tím dochází ke kompresi dat při minimální ztrátě informace a současně ke zlepšení stability některých predikčních modelů. [5]

Hlavní komponenty jsou vlastní vektory kovarianční matice normalizovaného a centrovaného datasetu. Řadí se podle velikosti jim náležících vlastních čísel a dále se používá pouze určitý počet nejdůležitějších komponent a dochází takke snížení počtu atribut v datasetu. Tyto nové "atributy" však již nemají ve svém jednotlivém významu nic společného s původními veličinami.

Pokud máme matici dat $X \in \mathbb{R}^{n \times p}$ (kde **n** je počet pozorování a **p** je počet proměnných), která je centrovaná, neboli pro ni platí

$$\sum_{i=1}^{n} x_{ij} = 0; j = 1, ..., p$$

, pak se může vypočítat kovariační matice **S**

$$S = \frac{1}{n-1} X^T X.$$

Pro vypočtenou kovariační matici se naleznou vlastní vektory a vlastní čísla podle rovnice

$$Sv_i = \lambda_i v_i$$

, kde λ_i jsou vlastní čísla
a $\mathbf{v_i}$ jsou vlastní vektory/hlavní komponenty.

Poté se vybere prvních **k** hlavních komponent a je možné transformovat data do matice **Z**

$$Z = XV$$

, kde $V = \begin{bmatrix} v_1 & v_2 & \dots & v_k \end{bmatrix}$

Transformovaná data pak budou mít rozměry $\mathbf{n} \times \mathbf{k}$.

2.1.3 Random forest (RF)

Random Forest je ensemble algoritmus využívající velké množství rozhodovacích stromů. Každý strom je trénován na náhodně vybraném podvzorku dat a atributů, čímž dochází ke snížení rizika přeučení modelu. RF lze využít nejen k predikci, ale také k identifikaci nejvýznamnějších atributů ovlivňujících výstupní proměnnou.

Permutační důležitost Po natrénování modelu jsou náhodně promíchány hodnoty vybraného atributu a následně je sledován pokles přesnosti modelu. Výraznější pokles přesnosti indikuje vyšší význam daného atributu pro predikci.[6]Při procesu se dají sledovat i ukazatele jako MSE, nebo R^2.

2.2 Algoritmy predikce

2.2.1 Regrese pomocí podpůrných vektorů (SVR)

SVR je regresní metoda založená na principu podpůrných vektorů. Cílem algoritmu je nalézt funkci, která co nejlépe aproximuje závislost mezi vstupními a výstupními daty při zachování definované tolerance chyby. Díky využití jádrových funkcí je SVR schopna modelovat i výrazně nelineární vztahy mezi proměnnými. [7]

2.2.2 Gradient Boosting

GradientBoosting představuje ensemble metodu založenou na postupném skládání slabých modelů, nejčastěji rozhodovacích stromů. Jednotlivé modely jsou trénovány sekvenčně tak, aby minimalizovaly chyby předchozích modelů. Moderní implementace, například XGBoost nebo LightGBM, patří mezi nejvýkonnější algoritmy pro tabulková data. [8]

2.2.3 Umělé neuronové sítě (ANN)

Umělé neuronové sítě představují pokročilé modely inspirované strukturou biologických neuronových sítí. Využívají se pro regresi, klasifikaci i analýzu časových řad. Hlavní výhodou ANN je schopnost aproximovat velmi složité nelineární vztahy mezi vstupy a výstupy. Významným milníkem ve vývoji ANN bylo zavedení algoritmu zpětného šíření chyby (backpropagation), který umožnil efektivní trénování vícevrstvých sítí.[9, 10]

Běžný model neuronu podle McCullocha a Pittse je definován rovnicí:

$$y = f\left(\sum_{i=1}^{N} w_i x_i - \theta\right)$$

Aktivační funkce neuronu rozhoduje o výsledném výstupu neuronu a zavádí do modelu nelinearitu, která je klíčová pro schopnost sítě učit se složité vztahy.

2.2.4 LSTM

LSTM (Long Short-Term Memory) je speciální typ rekurentních neuronových sítí určený pro práci se sekvenčními daty, například časovými řadami. Architektura LSTM umožňuje uchovávat dlouhodobé závislosti v datech a současně eliminovat problém mizejícího gradientu. Klíčovou součástí modelu jsou tzv. brány (forget gate, input gate a output gate), které regulují tok informací uvnitř sítě.[11]

2.2.5 ARIMA

Model ARIMA kombinuje autoregresivní část (AR), diferencování (I) a klouzavý průměr (MA). Model je vhodný zejména pro analýzu a predikci časových řad se sezónním nebo lineárním charakterem. Výhodou modelu ARIMA je relativně nízká výpočetní náročnost a dobrá interpretovatelnost výsledků. Naopak nevýhodou je požadavek na stacionaritu časové řady a omezená schopnost modelovat výrazně nelineární vztahy.

Obecná rovnice pro model ARIMA(p, d, q) je

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right)(1-L)^d X_t = c + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right)\epsilon_t$$

, kde **L** představuje výstupy z minulých kroků,
p je řád AR části,
$\phi_\mathbf{i}$ jsou AR koeficienty,
d je stupeň diferencování,
$\mathbf{X_t}$ je výstup tohoto kroku,
c je konstanta,
$\theta_\mathbf{i}$ jsou koeficienty MA části,
q je řád MA části,
a $\epsilon_\mathbf{t}$ je náhodná chyba.

Hodnoty **p**, **d** a **q** se určují pomocí různých nástrojů (např. autokorelační funkce, informačních kritérií nebo automatické algoritmy), koeficienty $\phi_\mathbf{i}$ a $\theta_\mathbf{i}$ jsou pak nejčastěji počítány metodou maximální věrohodnosti nebo metodou nejmenších čtverců.[12]

Literatura

[1] Simillion Hans, Dolgikh Olga, Terryn Herman, and Deconinck Johan. Atmospheric corrosion modeling. *https://www.researchgate.net/publication/270577320_Atmospheric_corrosion_modeling*, 2014.

[2] Geneva: ISO. Corrosion of metals and alloys — corrosivity of atmospheres — classification, determination and estimation. *ISO 9223:2012*, 2012.

[3] T. E. Graedel. Gildes model studies of aqueous chemistry 1. formulation and potential applications of the multi-regimemodel. *Corrosion Science*, 12(38):2153–2180, 1996.

[4] Rhondene Wint. On maximal information coefficient: A modern approach for finding associations in large data sets. *https://medium.com/@rhondenewint93/on-maximal-information-coefficient-a-modern-approach-for-finding-associations-in-large-data-sets-ba8c36ebb96b*, 2019.

[5] IBM. What is principal component analysis (pca)? *https://www.ibm.com/think/topics/principal-component-analysis*, 2023.

[6] Somayeh Youssefi. Permutation importance vs impurity-based feature importance. *https://medium.com/@syoussefi600/permutation-importance-vs-impurity-based-feature-importance-1c1a8d027479*, 2023.

[7] Siddhesh Bangar. Understanding support vector regression: A guide to machine learning's powerful tool. *https://medium.com/@siddheshb008/understanding-support-vector-regression-a-guide-to-machine-learnings-powerful-tool-4ada7081934b*, 2023.

[8] Tomonori Masui. All you need to know about gradient boosting algorithm - part 1. regression. *https://medium.com/data-science/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502*, 2022.

[9] Narciso Cerpa Steven Walczak. Artificial neural networks. *Encyclopedia of Physical Science and Technology (Third Edition)*, pages 631–645, 2003.

[10] IBM. What is backpropagation? *https://www.ibm.com/think/topics/backpropagation*, 2024.

[11] Christopher Olah. Understanding lstm networks. *https://colah.github.io/posts/2015-08-Understanding-LSTMs*, 2015.

[12] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 2002.

VÝPOČETNĚ ÚSPORNÁ METODA PROSTOROVÉHO SROVNÁVÁNÍ CT SNÍMKŮ NOSNÍCH DUTIN

Ing. Oto Vrána [1]

[1] ***Ústav přístrojové a řídicí techniky, Fakulta strojní, ČVUT v Praze, vranaoto@cvut.cz***

Abstrakt: Sledování anatomických změn v čase, například při hodnocení pooperačního hojení, vyžaduje přesné prostorové srovnání 3D obrazových dat z různých vyšetření. Běžně používaný postup spoléhá na ukotvení dat k rozsáhlé stabilní anatomické struktuře, což je ale výpočetně náročné a vyžaduje zpracování velkého objemu dat i mimo oblast skutečného zájmu. Tato práce se zabývá návrhem automatizovaného postupu zpracování CT dat, jehož cílem je dosáhnout srovnatelné přesnosti výrazně úspornějším způsobem, vhodným pro nasazení na běžném výpočetním vybavení klinických pracovišť. Byl vyvinut automatizovaný systém v jazyce Python, který vstupní CT data nejprve izotropně převzorkuje a následně segmentuje. Navržený postup, pracující přímo s oblastí zájmu, byl porovnán s konvenčním přístupem využívajícím rozsáhlou referenční strukturu, a to na datech reálných pacientů s pooperačními změnami v oblasti nosních dutin. Výsledky ukazují, že úspornější přístup dosahuje srovnatelné přesnosti při podstatně nižších nárocích na výpočetní čas i paměť.

Klíčová slova: počítačová tomografie, prostorová registrace, zpracování obrazu, výpočetní efektivita

1 Úvod

Dlouhodobé sledování anatomických změn patří mezi důležité nástroje pro objektivní hodnocení pooperačního vývoje, sledování progrese onemocnění i plánování navazující léčby. Aby bylo možné snímky z různých časových okamžiků vzájemně porovnat, je nutné je nejprve prostorově sjednotit – tomuto procesu se říká registrace [1]. Automatizované zpracování dat za tímto účelem však naráží na dva zásadní problémy.

Prvním je nerovnoměrné rozlišení běžných klinických CT vyšetření, kdy je rozlišení v jedné rovině výrazně jemnější než vzdálenost mezi jednotlivými řezy. To při generování 3D modelu vede k viditelným „schodovitým“ artefaktům, které jsou zvlášť problematické u tenkostěnných struktur, jako jsou právě nosní dutiny, kde mohou znehodnotit přesnost následného výpočtu transformace [2].

Druhým problémem je proměnlivost sledované oblasti po operaci. Chirurgické zákroky, otoky nebo zánětlivé procesy způsobují lokální změny tkáně, kvůli kterým se standardní registrační algoritmy, jako je klasický Iterative Closest Point (ICP) [3], mohou „zaseknout“ a konvergovat k anatomicky nesprávnému řešení.

Z těchto důvodů se v oboru obecně předpokládá, že nejvyšší spolehlivosti lze u podobných dat dosáhnout pouze ukotvením k rozsáhlé stabilní struktuře – typicky nedotčené kostní stavbě lebky, která je pooperačními změnami ovlivněna minimálně [1, 4]. Naproti tomu přímé srovnávání samotné operované oblasti je považováno za rizikové právě kvůli zmíněné topologické proměnlivosti [4].

V praxi to znamená, že řada pracovišť, která by z podobné analýzy mohla těžit, k ní dnes nemá reálný přístup – nikoliv kvůli nedostatku dat, ale kvůli nárokům na výpočetní infrastrukturu. Pokud by se ukázalo, že srovnatelné výsledky lze získat i bez nutnosti zpracovávat rozsáhlá referenční data, otevřelo by to cestu k běžnějšímu využití těchto nástrojů i mimo specializovaná výzkumná centra.

Cílem této práce je experimentálně ověřit, zda je s důsledným předzpracováním dat (zejména izotropním převzorkováním [2]) možné dosáhnout srovnatelné přesnosti i výrazně úspornějším přístupem, pracujícím přímo s oblastí zájmu, a nabídnout tak postup realisticky nasaditelný i bez specializovaného

výpočetního vybavení. Za tímto účelem byl navržen a implementován kompletní automatizovaný zpracovatelský řetězec, od načtení vstupních CT dat až po kvantitativní vyhodnocení obou porovnávaných přístupů, který je popsán v následující kapitole.

2 Metodika

2.1 Předzpracování dat a extrakce 3D modelů

Do práce byla zahrnuta anonymizovaná CT data pacientů s pooperačními změnami v oblasti nosních dutin, vždy ve dvojicích předoperačního a pooperačního snímku.

Vzhledem k nerovnoměrnému rozlišení vstupních dat byla všechna CT data nejprve izotropně převzorkována na krychlové voxely, jejichž velikost byla odvozena od nejjemnějšího rozlišení v původních datech. Tento krok potlačuje vznik „schodovitých“ artefaktů a zajišťuje hladké a topologicky spojité povrchové modely, nezbytné pro stabilní výpočet registrace [2].

Z takto připravených dat byly následně extrahovány 3D povrchové modely prahováním podle radiodenzity (Hounsfieldovy jednotky). Pro segmentaci vzduchových dutin byl použit práh −300 HU, pro kostní tkáň 200 HU. Po vyhlazení Gaussovým filtrem byla pro generování výsledné polygonální sítě použita metoda Marching Cubes [5].

Pro potřeby srovnávací (referenční) metody byla z dat dále izolována horní část lebky, odpovídající stabilní a pooperačně nedotčené kostní struktuře, zatímco pohyblivé struktury (dolní čelist, krční páteř) byly z dat odstraněny, aby do srovnání nevnášely nežádoucí odchylky.

2.2 Porovnávané přístupy

Byly navrženy a porovnány dva přístupy registrace.

Metoda A (lokální) pracuje přímo s cílovou oblastí (nosními dutinami) pomocí postupně zpřesňovaného algoritmu ICP [3, 6, 7], který v několika krocích (s klesající tolerancí 10, 5 a 2 mm) hledá nejbližší odpovídající si body mezi snímky a iterativně je k sobě přibližuje.

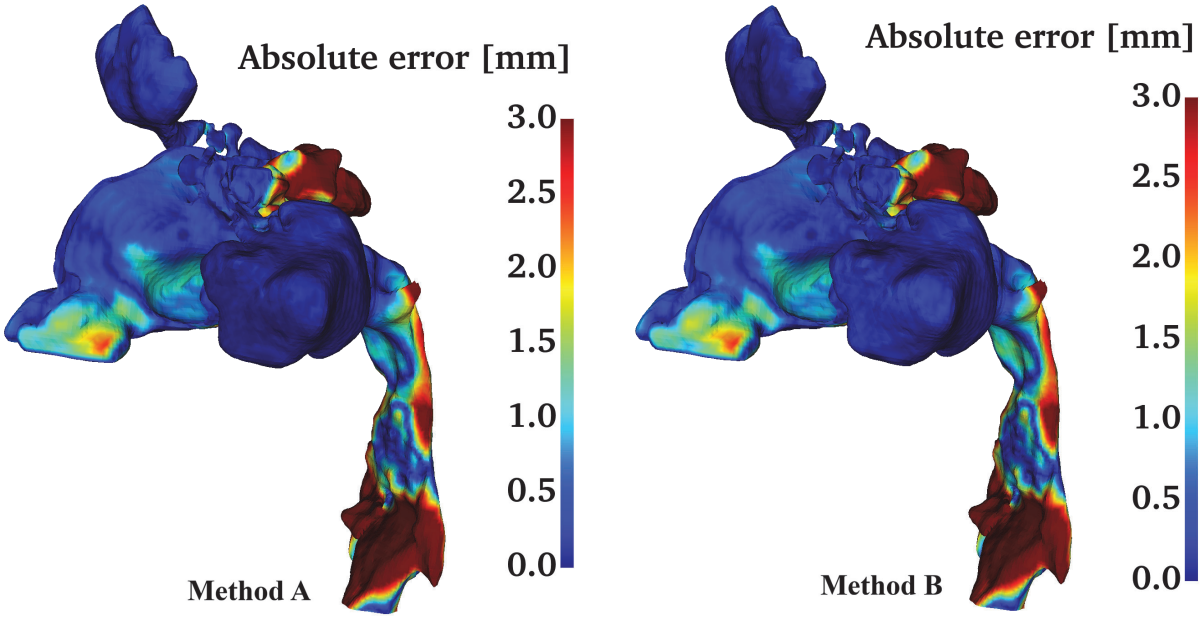
Metoda B (referenční) naopak vychází z izolované horní části lebky jako stabilního anatomického orientačního bodu. Snímky jsou k sobě nejprve hrubě přiřazeny pomocí algoritmu RANSAC [8], odolného vůči lokálním odchylkám, a následně doladěny standardním algoritmem ICP. Takto získaná transformace je pak aplikována na cílovou oblast nosních dutin.

2.3 Hodnocení výsledků

Přesnost obou metod byla hodnocena pomocí střední kvadratické odchylky (RMSE) mezi odpovídajícími si body povrchů a pomocí 95. percentilu Hausdorffovy vzdálenosti (HD95), který na rozdíl od klasické Hausdorffovy vzdálenosti není citlivý na ojedinělé extrémní odchylky způsobené např. neúplným překryvem zorného pole při snímání. Dále byl sledován podíl bodů ležících v rámci zvolené tolerance (2 mm) a souhrnný index kvality registrace, kombinující oba uvedené parametry. Výpočetní náročnost obou metod byla posuzována podle celkového času zpracování a maximální využité operační paměti [9].

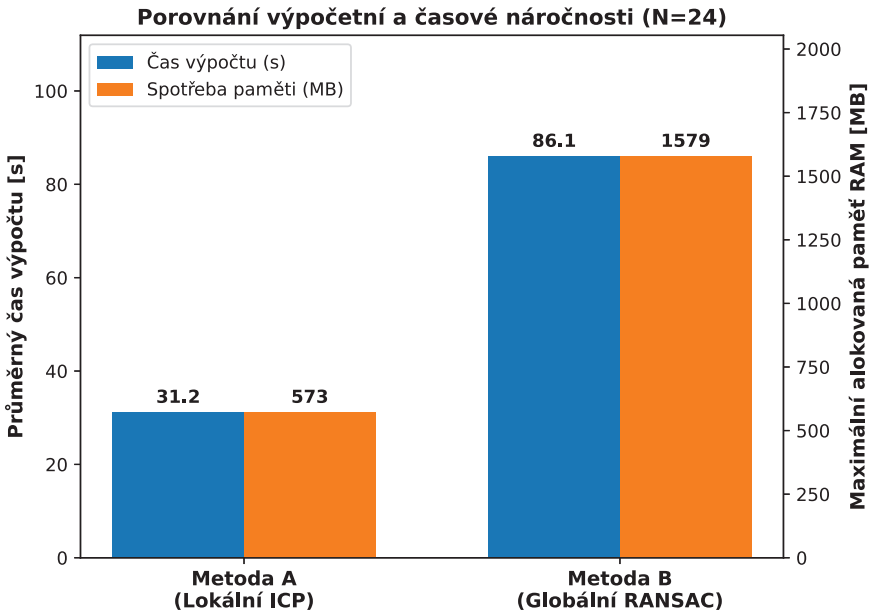
3 Výsledky a diskuse

Srovnání bylo provedeno na datech 24 pacientů. Obě metody dosáhly prakticky shodné přesnosti – střední kvadratická odchylka na cílových strukturách byla u obou přístupů $0{,}73 \pm 0{,}05$ mm, tedy hluboko pod hranicí běžně vyžadovanou pro kraniomaxilofaciální chirurgii. Při podrobnějším srovnání jednotlivých pacientů nepřesáhl rozdíl mezi metodami 0,03 mm u RMSE, 0,43 mm u HD95 a 1,5 % u souhrnného indexu kvality. Obě metody tedy konvergují prakticky ke stejnému prostorovému řešení, a to i v okrajových oblastech ovlivněných neúplným zorným polem snímání.



Obr. 1: Vizuální srovnání prostorové odchylky obou metod na povrchu cílové struktury; barevná škála znázorňuje absolutní odchylku v mm.

Zcela odlišný obrázek se ale ukázal u výpočetní náročnosti. Metoda A dosáhla průměrného výpočetního času $31{,}2 \pm 3{,}7$ s, zatímco Metoda B potřebovala v průměru $86{,}1 \pm 11{,}2$ s – tedy téměř třikrát více. Obdobný poměr platil i pro operační paměť: 573 ± 56 MB u Metody A oproti 1580 ± 173 MB u Metody B.



Obr. 2: Srovnání průměrného výpočetního času a maximální alokované paměti pro obě metody ($N = 24$).

Tento rozdíl plyne z toho, že Metoda B musí navíc segmentovat, izolovat a transformovat rozsáhlá data celé lebky, zatímco Metoda A pracuje pouze s mnohem menší oblastí skutečného zájmu. Získané výsledky tak ukazují, že srovnatelné přesnosti lze dosáhnout i bez nutnosti zpracovávat objemná referenční data, a to i v případech, kdy pooperační snímky vykazují výrazné lokální změny nebo neúplné zorné pole.

Z praktického hlediska je tento rozdíl podstatný zejména tam, kde se podobná analýza má provádět opakovaně nebo na větším počtu pacientů – například při hromadném zpracování dat v rámci klinické studie. Zatímco Metoda B kvůli svým nárokům na paměť i čas prakticky vyžaduje výkonnou pracovní

stanici, Metoda A je díky nízké výpočetní náročnosti dobře škálovatelná a lze ji spustit i souběžně na větším množství dat bez výrazných nároků na hardwarové vybavení pracoviště.

3.1 Omezení a další směřování

Získané výsledky je třeba vnímat v kontextu charakteru použitých dat a zvoleného postupu. Soubor 24 pacientů, byť postačující pro prokázání srovnatelnosti obou metod, je z klinického hlediska stále poměrně malý a pro obecnější závěry by bylo vhodné jej v budoucnu rozšířit, ideálně i o data z jiných pracovišť a CT přístrojů s odlišnými zobrazovacími protokoly. Rovněž by bylo přínosné ověřit, jak si lokální přístup poradí s případy ještě výraznějších pooperačních defektů, než jaké byly zastoupeny v tomto souboru dat.

Další směr výzkumu se nabízí také ve zpracování samotném. Aktuální implementace spoléhá na pevně nastavené prahy pro segmentaci (založené na Hounsfieldových jednotkách) i na manuálně zvolenou posloupnost tolerančních kroků registračního algoritmu; jejich automatická adaptace na konkrétní vstupní data by mohla dále zvýšit robustnost celého postupu. Zajímavým rozšířením by bylo také otestování navrženého přístupu na jiných anatomických oblastech než na nosních dutinách, kde lze podobné výpočetní úspory očekávat za předpokladu dostatečně výrazné lokální geometrie cílové struktury.

4 Závěr

Tato práce ověřila, že pro prostorové srovnání pooperačních CT dat nosních dutin lze dosáhnout srovnatelné přesnosti (RMSE $0{,}73 \pm 0{,}05$ mm) jak postupem využívajícím rozsáhlou referenční strukturu lebky, tak výrazně úspornějším přístupem pracujícím přímo s cílovou oblastí. Druhý uvedený přístup přitom vyžaduje zhruba třetinový výpočetní čas a třetinovou paměťovou náročnost. Navržený postup tak otevírá cestu k nasazení podobných analytických nástrojů přímo na běžném výpočetním vybavení klinických pracovišť, bez nutnosti investic do specializovaného hardwaru.

Přestože byly výsledky ověřeny na omezeném souboru pacientů a téma by si jistě zasloužilo další zkoumání (zejména na větším a různorodějším souboru dat), podařilo se ukázat, že výpočetní úspornost a klinicky dostatečná přesnost nejsou nutně protichůdné požadavky. To je povzbudivé zjištění pro budoucí praktické nasazení podobných nástrojů v každodenní klinické praxi.

Poděkování

Práce byla podpořena grantem Studentské grantové soutěže ČVUT v Praze č. SGS24/125/OHK2/3T/12 „Strojové učení pro optimalizaci řídicích systémů“.

Literatura

[1] Z. Zinser et al. Accuracy of facial scan registration: A comparison between full-cranium vs reduced fov cbct. *Scientific Reports*, 14(1), 2026.

[2] Chenlei Lv, Weisi Lin, and Baoquan Zhao. Intrinsic and isotropic resampling for 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3274–3291, 2023.

[3] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[4] Fakhre Alam et al. An automatic medical image registration approach based on common subregions of interest. *Journal of Medical Imaging and Health Informatics*, 9(2):251–260, 2019.

[5] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.

[6] Yang Chen and Gérard Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

[7] Ali Taleb et al. Registration of preoperative temporal bone ct-scan to otoendoscopic video for augmented-reality based on convolutional neural networks. *European Archives of Oto-Rhino-Laryngology*, 281(6):2921–2930, 2024.

[8] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3212–3217, Kobe, Japan, 2009.

[9] Rohit Jena, Pratik Chaudhari, and James C. Gee. Fireants: Adaptive riemannian optimization for multi-scale diffeomorphic registration, 2024.



Editor: Vladimír Hlaváč
Název díla: Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2026
Vydalo: České vysoké učení technické v Praze
Zpracoval: Ústav přístrojové a řídicí techniky Fakulty strojní ČVUT v Praze
Kontaktní adresa: Ústav přístrojové a řídicí techniky, FS ČVUT v Praze, Technická 4, Praha 6
Tel.: +420 2 2435 9750
Počet stran: 77 Pořadí vydání: 1.
Vydáno dne 30. 9. 2026 jako pdf soubor na CD.
Dostupné na webových stránkách Ústavu přístrojové a řídicí techniky: https://iat.fs.cvut.cz/nmp
ISBN 978-80-01-07602-6