http://doi.org/10.35784/iapgos.8086

received: 11.07.2025 | revised: 02.01.2026 | accepted: 05.01.2026 | available online: 30.03.2026

SMARTPHONE SHELL TEMPERATURE CONTROLLER AUTOMATIC TUNING METHOD

Danylo Zinchenko[1], Yurii Mariiash[2]

[1]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine, [2]Institute of Nuclear and Thermal Energy, Department of Energy Processes Automation, Kyiv, Ukraine

Abstract. *The performance of mobile Systems-on-Chip is frequently limited by elevated shell temperatures that degrade user comfort. This paper presents a PID-based thermal controller that co-manages central and graphics processing units' frequencies, featuring a novel auto-tuning framework to optimize its parameters for any given device. The principal result is a universal tuning formula, derived by simulating diverse hardware thermal models to establish a polynomial relationship. This formula maps easily measurable system characteristics from an on-device relay experiment to optimal PID parameters, ensuring controller stability across different hardware configurations, with average gain margin of 2.14, and phase margin of 78°. Validation on a commercial smartphone confirmed the formula's accuracy, yielding PID parameters with less than 5% deviation from the theoretical optimum. When benchmarked against the device's default hysteresis governor, our controller demonstrated significant gains, simultaneously increasing sustained performance by up to 9% while reducing thermal limit violations by up to 2.6 times. This work provides a validated, practical framework for automatic tuning of control systems, that maximize mobile device performance within user comfort constraints.*

Keywords: dynamic thermal management, PID control, frequency control, controller auto-tuning

METODA AUTOMATYCZNEGO DOSTRAJANIA REGULATORA TEMPERATURY OBUDOWY SMARTFONU

Streszczenie. *Wydajność mobilnych układów SoC jest często ograniczana przez podwyższoną temperaturę obudowy, co negatywnie wpływa na komfort użytkownika. W niniejszym artykule przedstawiono regulator termiczny oparty na algorytmie PID, który współzarządza częstotliwościami procesora centralnego i procesora graficznego, wykorzystując nowatorską strukturę automatycznego dostrajania w celu optymalizacji parametrów dla dowolnego urządzenia. Głównym wynikiem jest uniwersalna formuła strojenia, wyprowadzona poprzez symulację różnych modeli termicznych sprzętu w celu ustalenia zależności wielomianowej. Formuła ta przekłada łatwo mierzalne charakterystyki systemu z eksperymentu przekaźnikowego przeprowadzonego na urządzeniu na optymalne parametry PID, zapewniając stabilność regulatora w różnych konfiguracjach sprzętowych, ze średnim marginesem wzmocnienia wynoszącym 2,14 i marginesem fazowym wynoszącym 78°. Walidacja na komercyjnym smartfonie potwierdziła dokładność formuły, dając parametry PID z odchyleniem mniejszym niż 5% od teoretycznego optimum. W porównaniu z domyślnym regulatorem histerezy urządzenia nasz regulator wykazał znaczące korzyści, jednocześnie zwiększając trwałą wydajność nawet o 9% i zmniejszając przekroczenia limitów termicznych nawet 2,6-krotnie. Praca ta zapewnia sprawdzoną, praktyczną strukturę do automatycznego dostrajania systemów sterowania, która maksymalizuje wydajność urządzeń mobilnych w ramach ograniczeń komfortu użytkownika.*

Słowa kluczowe: dynamiczne zarządzanie temperaturą, sterowanie PID, kontrola częstotliwości, automatyczne strojenie regulatora

Introduction

The landscape of mobile device computing has been profoundly reshaped by rapid technological advancements. Most notably, the widespread adoption of Systems-on-Chip (SoCs), integrating a number of diverse processing elements, from central and graphics processing units to various accelerators, onto a single die, has become the crucible for executing increasingly complex and computationally demanding tasks [12]. The increasing demand for the computational power within these devices, pushed by the widespread use of demanding applications such as deep learning, high-fidelity gaming [11], and extensive data processing, necessitates continuous performance scaling [13].

In a direct response to this escalating requirement, manufacturers have continuously increased the number and capability of processing cores integrated into these SoCs. This intensive concentration of computational resources, while delivering increased performance, inherently drives up the power density on the silicon, consequently presenting critical challenges in effective thermal management [14].

Crucially, for mobile devices, which are designed for handheld operation, the internal thermal state translates directly to the external shell temperature. This thermal interface with the user is a matter of comfort and safety, which is a fundamental determinant of the user experience. Elevated shell temperatures, a direct consequence of the inability to fully dissipate the heat generated by high-density SoCs [18], lead to a negative impact on device usability during prolonged or intensive tasks. Overheating has been demonstrated to induce discomfort, pain, and negative emotional responses in users [20], with these effects potentially persisting for up to an hour post-usage [2]. It has been observed that case temperatures exceeding 45°C, a state which can be achieved during common activities, result in discomfort for more than 75% of individuals surveyed [7].

Existing thermal management strategies primarily prioritize controlling internal junction temperatures, often neglecting the external thermal profile. In the realm of academia, the state-of-the-art in advanced management is often represented by advanced predictive algorithms. However, these methodologies have yet to achieve widespread adoption. This is primarily due to the intricacies inherent in their implementation and real-time application. In the contemporary context, the implementation of hysteresis and Proportional-Integral-Derivative (PID) approaches is predominant [19], particularly in systems where computational capacity is paramount. The industry standard for limiting shell temperature, the hysteresis controller, while preventing extreme temperatures, significantly degrades performance through aggressive throttling. While some research explores advanced shell temperature regulation, such as cascaded PID junction and shell controllers [3] or step-wise controllers incorporating predictive inputs [4], these approaches only manage the central processing unit (CPU) frequencies, omitting the second-most heat producing element of a typical mobile SoC – the graphics processing unit (GPU). Moreover, they critically lack a robust mechanism for automatic tuning of algorithmic parameters specific to individual hardware platforms. This deficiency is significant, as industry practice frequently misconfigures PID controllers using generalized tuning methods unsuitable for individual device thermal dynamics [10].

The field of Dynamic Thermal Management (DTM) for multi-core mobile platforms is a well-established area of research, with comprehensive surveys by authors such as Pasricha, Ayoub, Kishinevsky, Mandal, and Ogras [15] detailing the landscape of proposed techniques, which range from reactive throttling to proactive, predictive control. Within this field, the evolution of control strategies has progressed from rudimentary on-off methods to more sophisticated approaches. Researchers have analyzed the application of PID controllers, concluding they offer superior performance over basic hysteresis controllers by enabling smoother adjustments to processor frequencies.

However, these advanced implementations have largely focused on the CPU as the sole element of control. This narrow focus is a significant limitation, as demonstrated by the findings of Pathania, Jiao, Prakash, and Mitra [16] who concluded that independent CPU and GPU throttling is "inadequate" for modern gaming workloads. Their research established that a lack of coordination between the two main heat sources leads to "recurrent frequency throttling" and significantly degraded performance, proving that a cooperative CPU-GPU management strategy is essential. The negative impact of uncoordinated throttling on the user experience is further reinforced by Lee, Im, Huh, and Han [9], who noted that conventional DTM often degrades the performance of the primary application the user is interacting with.

The effectiveness of any PID controller is critically dependent on the tuning of its parameters. The classical tuning rules developed by Ziegler and Nichols, while foundational, have been shown to be ill-suited for systems with complex thermal dynamics. Modern analysis by Huba, Chamraz, Bisták, and Vrančić [5] notes the limitations of these methods in digital implementations, while other technical reviews confirm they produce an "oscillatory closed loop response" with significant overshoot, a characteristic that is undesirable for maintaining a stable and comfortable device shell temperature.

A safer and more robust alternative for system identification is the relay auto-tuning method developed by Åström and Hägglund [1]. This technique has become an industrial standard for experimentally determining a system's ultimate gain and ultimate period. These parameters, in turn, are used in various established tuning rules to calculate PID parameters. However, these rules are designed to be "universal," providing robust control for a wide array of industrial processes. This generality becomes a liability in this specific application. They are not optimized for the unique, higher-order thermal dynamics of a multi-layered mobile device, nor are they designed to balance the specific trade-off between user comfort and maximum computational performance. Applying such a formula may result in a suboptimal compromise – either being overly conservative and unnecessarily sacrificing performance, or failing to adequately constrain temperature.

In this paper, we propose an auto-tuned PID-based shell thermal throttling technique that co-manages both CPU and GPU frequencies to maintain the device shell temperature at acceptable levels. While the coordinated control of both major heat sources presents a significant improvement, the primary contribution of this work is a novel auto-tuning algorithm. This algorithm generates a hardware-specific tuning formula, enabling the rapid derivation of optimal PID parameters from a simple system identification experiment, thereby maximizing performance without compromising user comfort.

1. Control loop structure

The control loop is configured as a negative feedback system, as illustrated in Figure 1. The measured shell temperature T_shell, obtained from an internal temperature sensor, serves as the process variable (PV). This PV is compared against a desired temperature level – setpoint (SP), chosen as the user discomfort threshold (e.g., 45°C). The resulting error signal, $e(t) = SP - PV(t)$, is fed into the PID controller.

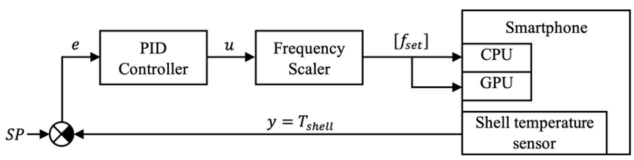

Fig. 1. Control loop structure

The loop operates reactively: if the shell temperature rises above the setpoint, the error becomes negative, causing the PID controller to output a lower signal. This lower signal commands the frequency scaler to reduce CPU/GPU speeds, decreasing heat generation, and allowing the shell to cool. Conversely, if the temperature is below the setpoint (or the error is positive), higher frequencies are permitted, increasing performance and heat generation.

The controller structure is selected as an ISA standard PID controller with the derivative dampening filter [6], whose continuous-time transfer function is given by

$$W_{PID} = K_p\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1+(T_d s)/N}\right) \tag{1}$$

where K_p is the controller gain, T_i and T_d are the integral and derivative time constants and N is the derivative filter parameter, representing the ratio between the T_d and the time constant of and additional dampening pole, and s is the complex variable of the Laplace transform. A derivative filter is incorporated to reduce sensitivity to high-frequency noise present in temperature measurements and improve robustness.

To prepare the controller for a real-world smartphone environment, the continuous-time form is digitalized using the bilinear transform, which maps the continuous s-plane to the discrete z-plane according to the substitution

$$s = \frac{2}{T_s} * \frac{1-z^{-1}}{1+z^{-1}} \tag{2}$$

where T_s represents the sampling period. Applying this transformation to (1) yields the final discrete-time transfer function used for implementation:

$$W(z) = K_p\left(\begin{array}{c} 1 + \frac{T_s(z+1)}{2T_i(z-1)} + \\ + \frac{2T_d N(z-1)}{z(2T_d+NT_s)+(NT_s-2T_d)} \end{array}\right) \tag{3}$$

where z is the complex variable of the Z-transform, and z^{-1} can be interpreted as a unit time delay operator.

As the supported frequencies for each of controlled processing units are different and not continuous, a frequency scaler control loop element is introduced. This element maps the controller output u to a target frequency value f_{target}, within the supported range $[f_{min}, f_{max}]$, using the linear scaling function

$$f_{target} = f_{min} + u(f_{max} - f_{min}) \tag{4}$$

after which the mechanism selects the nearest available frequency, f_{set} from the predefined list of supported frequencies, $f_{supported}$ by finding the one that minimizes the absolute difference:

$$f_{set} = \mathrm{argmin}_{f \epsilon f_{supported}} \left|f_{target} - f\right| \tag{5}$$

2. PID parameter optimization via process simulation

Desired control system performance is achieved by tuning the controller parameter set (K_p, T_i, T_d), in an optimization loop, using differential evolution (DE) global optimization algorithm [17]. This population-based approach was selected for its robustness in navigating non-convex solution spaces and resistance to local minima convergence, which is particularly advantageous for control systems with complex dynamics.

To capture the shell temperature dynamics of the electronic system under study, a Second-order plus time delay model with a zero (SOPTD+Z) was adopted, defined by the transfer function

$$W_{SOPTD+Z}(s) = K_{plant}\frac{a_1 s+1}{b_2 s^2+b_1 s+1} \cdot e^{-\tau s}. \tag{6}$$

where its parameters were experimentally identified as K_{plant} = 0.805°C/%, a_1 = 291.3 s, b_1 = 1145 s, b_2 = $1.148 \cdot 10^5$ s^2 and τ = 5 s. This structure accounts for the characteristic initial temperature rise observed in electronic systems, attributed to direct Joule heating prior to thermal diffusion through layered materials.

Model validation was conducted using the shell temperature step response, that was experimentally obtained from the device under CPU and GPU stress workloads. Parameters for each candidate model structure tested (first-order plus time delay (FOPTD), and the adopted second-order plus time delay with zero) were identified by fitting their respective step responses to this experimental data using generalized reduced gradient method [8]. This comparison demonstrated the superiority of this formulation against alternative structures, yielding a mean absolute error (MAE) of 0.0751 versus 0.308 for a first-order model or a standard second-order model when evaluated against the experimental step response (Figure 2).

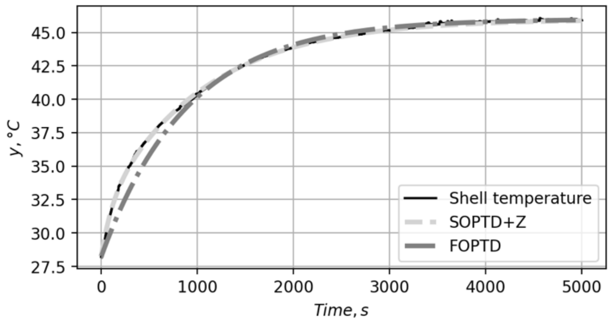

Fig. 2. Model comparison

The simulations for controller performance evaluation were specifically designed to assess the system's disturbance rejection capabilities. This was achieved by applying a disturbance d to the controller's output signal, thereby directly impacting the input to the plant. The decision to focus exclusively on the disturbance-output channel, rather than set point tracking, is justified by the typical operational context of a smartphone, where frequent changes of the shell temperature set point are highly unlikely.

The optimization process seeks to minimize a composite objective function, J, which balances control error against control effort oscillations, formulated as

$$J = \sum_{i=0}^{n} \left(\frac{y_i}{K_{plant}}\right)^2 + (u_i - d)^2 \tag{7}$$

where y_i is the deviation of the system output from its set point (which is implicitly zero for disturbance rejection analysis), and u_i is the controller output.

The first term of the objective function, represents the squared control error. Normalizing this error by the plant's static gain, K_{plant}, balances this term, by yielding a dimensionless measure of performance deviation. The second term, aims to penalize excessive control effort. Here, and, by subtracting the known disturbance d, this term effectively isolates and penalizes the difference between the controller's output signal and its final value. Minimizing this component discourages overly aggressive control actions, suppressing the number of frequency switches, which degrade the performance and energy efficiency.

By running the optimization procedure, optimal PID parameters were identified as $K_p = 40.079$ %/°C , $T_i = 164.3$ s, $T_d = 4.62$ s. The gain and phase stability margins of the obtained control system are $GM = 2.58$ and $PM = 76.8°$, which are within typical requirements for robust control systems in practical applications.

3. Automated tuning sequence

The core approach adapts the principles of the Åström and Hägglund relay experiment within a comprehensive simulation framework to derive a predictive formula linking system characteristics to optimal controller parameters.

The Åström and Hägglund relay experiment is carried out as follows (Fig. 3). Firstly, the process value is brought to a steady-state position. After that, a short control pulse is applied to initiate oscillations in the system. Once the oscillations begin, the controller is switched to a relay (on-off) mode. In this mode, the control signal alternates between two fixed values $[A_u, -A_u]$ each time the PV crosses the steady-state value. This switching results in sustained oscillations of the PV. The amplitude a and ultimate period T_u of these oscillations are measured. From these measurements and the known control action amplitude $d = 2A_u$, the ultimate gain of the system, is calculated as

$$K_u = \frac{4 * d}{\pi a} \tag{8}$$

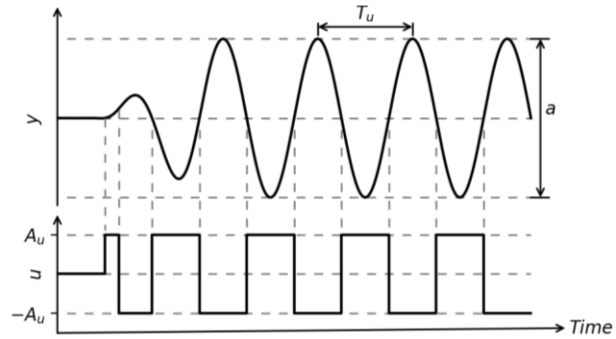

Fig. 3. Åström and Hägglund relay experiment

The methodology proceeds through the following steps. First, a dataset representing various possible system dynamics is generated. This is achieved by simulating a multitude of system models based on the previously identified second-order model with a zero, while perturbing its parameters within a ±50% range account for variations in device hardware, materials, and operating conditions across different smartphone models.

Next, for each of these simulated model variants, two sets of data are obtained:

- The results of a simulated Åström and Hägglund relay experiment – the ultimate gain K_u and period T_u. This simulation follows the procedure for the relay test, utilizing the previously identified process model.
- The calculated optimal PID controller parameter set $(K_p^{opt}, T_i^{opt}, T_d^{opt})$ for the specific model, determined through the previously described optimization process.

This systematic simulation and data collection across a range of models yields a comprehensive dataset where each entry contains a tuple of relay experiment parameters, paired with the optimal controller settings (Fig. 4).

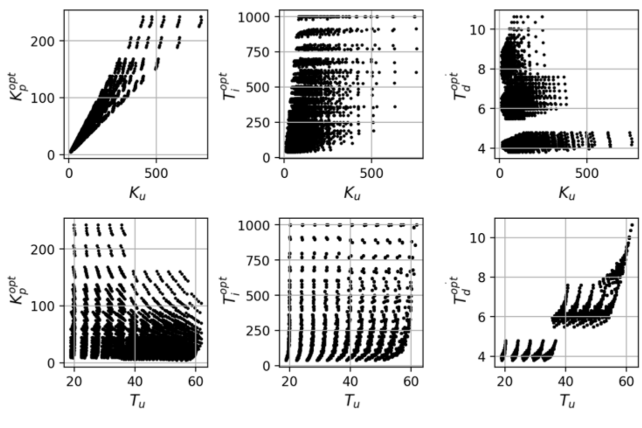

Fig. 4. Optimal PID coefficients versus critical gain and period

The final step involves using this dataset to establish the tuning formula. The data is divided into training and testing sets to ensure the formula's generalizability. An exhaustive search through all polynomial combinations up to the second order is then performed on the training set. This algorithm identifies the polynomial relationship that best describes the mapping from (K_u, T_u) to each of the PID controller parameters, by minimizing the prediction error when validated on the testing set. The coefficients of this best-fit polynomial constitute the parameters of the universal tuning formula, enabling the automatic calculation of PID parameters given the values obtained from a relay experiment on the target system.

This process yields a set of second-order polynomial equations that define the relationship between the experimentally determined system characteristics and the optimized controller settings:

$$K_p = 0.1172 \cdot K_u + 0.002 \cdot K_u T_u + \\ +0.4393 \cdot T_u - 0.006 \cdot T_u^2 - 15.684 \tag{9}$$

$$T_i = 0.4419 \cdot K_u - 0.0003 \cdot K_u^2 + \\ +0.0222 \cdot K_u T_u + 0.1392 \cdot T_u - 219.54 \tag{10}$$

$$T_d = 0.0002 \cdot K_u + 0.0959 \cdot T_u + 0.0002 \cdot T_u^2 + 0.3173 \tag{11}$$

A critical outcome of this methodology is that controllers tuned via these derived formulas are inherently robust. When evaluated across the range of simulated hardware models, the resulting control systems consistently demonstrated stability, achieving average stability margins: $GM = 2.12$ with a standard deviation of $\sigma = 0.16$ and $PM = 79.5°$ with a standard deviation of $\sigma = 6.74°$.

4. Experimental validation

The validation experiments were performed on a ZTE Nubia RedMagic 7s Pro smartphone, based on a triple cluster Qualcomm SM8450 Snapdragon 8 Gen 1 SoC, which is widely used on production smartphones. This smartphone features a virtual shell temperature sensor, with the update period of 5 seconds, which will be used for final parameters identification and control.

To provide a comprehensive assessment of performance, two distinct benchmarks were selected to stress different aspects of the SoC:

- To quantify the raw computational throughput of the CPU, the Whetstone benchmark was utilized. This benchmark executes a standardized suite of floating-point and integer arithmetic workloads, providing a direct measure of the system's calculation speed under CPU-intensive tasks – whetstone instructions per second (WIPS). To be fair on a heterogeneous system, the benchmark was modified to have joint operational budget between threads. The benchmark was run consecutively to simulate a dynamic workload.
- To evaluate performance in a realistic, graphically intensive scenario, the "Wild Life Extreme Stress Test" from the 3DMark benchmarking suite was employed. This test is specifically designed to stress both the CPU and GPU by rendering complex gaming scenes over an extended period. The primary metric recorded is the frame rate (frames per second), which serves as a direct indicator of user experience and gameplay fluidity.

The performance of the devised auto-tuned PID control strategy was compared against the smartphone's default thermal governor, which serves as the experimental baseline. The default governor implements a standard bang-bang (hysteresis) controller, which lowers the CPU and GPU frequency at shell temperature of 46°C, and removes the limitations at 44°C. Such a setup allowed us to test the device close to real temperature limits.

To ensure repeatability and a fair comparison between control strategies, all tests were conducted under controlled environmental conditions. The ambient temperature was maintained at 25°C, and each test commenced only after the device's initial shell temperature reached 35°C.

To compute the final controller parameters based on the identified relationship, a relay experiment was conducted on the device. The CPU architecture comprises three clusters – "Little", "Big", and "Prime". Only the "Big" and "Prime" clusters were included in the control process, as the thermal contribution of the "Little" cluster was deemed negligible. Although the CPU and GPU operate at discrete frequency levels, both the steady-state operating points and control amplitudes were selected to be approximately matched in terms of their relative positions within the full frequency range. This ensured that the steady-state amplitude and the amplitude of the relay control action were proportionally consistent across units (Table 1).

Table 1. Selected frequency values for the relay experiment

SoC device	Steady-state frequency, MHz	Relative steady-state amplitude, %	Positive relay level frequency, MHz	Negative relay level frequency, MHz	Combined relative relay amplitude, %
Big CPU cluster	1113.6	22.7	1324	883	20.8
Prime CPU cluster	1286.5	22.6	1536	1036	22.6
GPU	364	21.2	439	285	22.6

The 5-second update interval of the temperature sensor introduced a delay in the control response during the relay experiment, resulting in non-uniform oscillations (Fig. 5). Consequently, the ultimate parameters were determined by averaging the observed oscillation characteristics. This approach yielded the system's ultimate parameter values of $K_u = 260.18$ and $T_u = 36.3\ s$. The controller parameters were calculated using the identified relationship, getting $K_p = 41.73\ \%/°C$, $T_i = 162.4\ s$, $T_d = 4.11\ s$, which are all within 5% of absolute error from the values, that were got from the direct optimization.

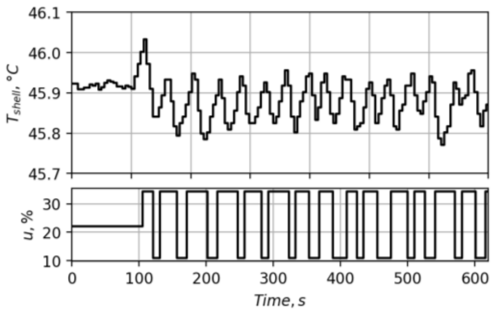

Fig. 5. On-device relay experiment temperature and control action traces

Figure 6a and 6b show the shell temperature, Big (f_{Big}) and Prime (f_{Prime}) CPU clusters, as well as GPU (f_{GPU}) frequency profiles of the smartphone in the time of running the Dhrystone and 3DMark benchmarks under proposed and default control strategies. It shows that in both tests the proposed controlled strategy effectively prevented skin temperature violations, while sustaining relatively high frequency levels, compared to the baseline logic that throttled the frequency aggressively, while allowing the shell temperature to go higher than the discomfort level.

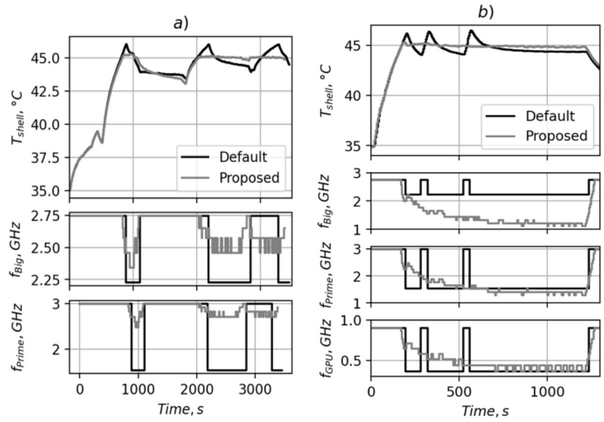

Fig. 6. Smartphone run-time metrics comparison during control strategies evaluation on the Whetstone benchmark (a) and the 3DMark benchmark (b)

Numeric results in the Table 2 show that in the Whetstone benchmark, proposed strategy, while reducing the thermal violations by 2.6 times, allowed for 2.3% of performance improvement with 1.8% decrease in power consumption. In the 3DMark benchmark, the developed strategy reduced the T_{shell} violations by 70% and increased the performance by 9%. The power increase in the latter case can be explained by the benchmark nature: in contrast to the Whetstone benchmark, which has a fixed computational workload for a single run, 3DMark is fixed-time benchmark, with the primary goal of maximizing the number of frames rendered within that period.

Table 2. On-device controller evaluation performance, thermal and power metrics

Control logic	Performance		Portion of T_{shell} violations, %	Top T_{shell} reached, °C	Average system power consumption, W
	Value	Metric			
Benchmark: Whetstone					
Baseline	4341.5	MWIPS	23.52	46.1	4.41
Proposed	4442.8		9.04	45.3	4.33
Benchmark: 3DMark					
Baseline	37.6	Average FPS	17.4	46.4	5.29
Proposed	41.4		10.2	45.3	5.46

A key outcome of the verification is the controller's ability to maintain the shell temperature at the desired level via smooth, continuous adjustments to core frequencies. This stable operation not only prevented aggressive thermal throttling but also enabled a marginal yet measurable increase in sustained performance throughout the benchmarks.

5. Conclusions

This paper addresses the challenge of thermal management in modern mobile devices, where increased computational power from Systems-on-Chip (SoCs) must be balanced with the need to maintain an acceptable external shell temperature. We identified a gap in existing thermal management strategies, which lack an efficient method for device-specific controller tuning. In response, we proposed a shell temperature control system utilizing a PID controller that modulates both CPU and GPU frequencies.

The primary contribution of this work is an auto-tuning methodology for PID controllers. Through offline simulation of a validated second-order thermal model across a range of hardware parameters, we established a polynomial relationship between a system's ultimate gain and period – determined from a relay experiment – and its optimal PID controller settings. This process yields a hardware-specific tuning formula, automating what is otherwise a complex manual tuning process.

Experimental validation on a commercial smartphone confirmed the methodology's efficacy. The auto-tuned controller reduced thermal limit violations by up to 2.6 times and increased performance by up to 9% when compared to the device's default hysteresis-based governor.

In conclusion, this research presents a framework for optimizing the trade-off between performance and user comfort in mobile computing. By providing a systematic method to derive tailored PID parameters from a single system identification experiment, our work connects control theory with industrial application, offering manufacturers a direct method to enhance user experience while maintaining computational performance.

Future work will focus on extending this methodology to power-allocation-based control strategies, and exploring adaptive mechanisms that adjust tuning parameters in response to changing ambient conditions and device aging.

References

[1] Åström, K. J., & Hägglund, T. (2006). *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society.
[2] Bauer, J., Górecki, I., Kohyt, M., Migasiewicz, A., & Podbielska, H. (2018). The influence of smartphones' operation modes on the superficial temperature distribution in the human auricle region: A small-scale (n = 20) clinical examination. *Journal of Thermal Analysis and Calorimetry*, *133*(1), 559–569. https://doi.org/10.1007/s10973-018-7047-8
[3] Chetoui, S., & Reda, S. (2022). CasCon: Cascaded Thermal and Electrical Current Throttling for Mobile Devices. *IEEE Embedded Systems Letters*, *14*(1), 3–6. https://doi.org/10.1109/LES.2021.3079707
[4] Egilmez, B., Memik, G., Ogrenci-Memik, S., & Ergin, O. (2015). User-Specific Skin Temperature-Aware DVFS for Smartphones. *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*, 1217–1220. https://doi.org/10.7873/DATE.2015.0148
[5] Huba, M., Chamraz, S., Bistak, P., & Vrancic, D. (2021). Making the PI and PID Controller Tuning Inspired by Ziegler and Nichols Precise and Reliable. *Sensors*, *21*(18), 6157. https://doi.org/10.3390/s21186157
[6] ISA Subcommittee 5.9. (2023). *ISA TR5.9 2023 Proportional integral derivative (PID) algorithms and performance*. International Society of Automation.
[7] Kang, S., Choi, H., Park, S., Park, C., Lee, J., Lee, U., & Lee, S.-J. (2019). Fire in Your Hands: Understanding Thermal Behavior of Smartphones. *The 25th Annual International Conference on Mobile Computing and Networking*, 1–16. https://doi.org/10.1145/3300061.3300128
[8] Lasdon, L. S., Fox, R. L., & Ratner, M. W. (1974). Nonlinear optimization using the generalized reduced gradient method. *Revue Française d'automatique, Informatique, Recherche Opérationnelle. Recherche Opérationnelle*, *8*(V3), 73–103. https://doi.org/10.1051/ro/197408V300731
[9] Lee, K. M., Im, Y., Huh, J. H., & Han, T. H. (2018). Adaptive thermal property control technique for holistic thermal management of mobile devices. *IEICE Electronics Express*, *15*(10), 20180187–20180187. https://doi.org/10.1587/elex.15.20180187
[10] Leva, A., Cox, C., & Ruano, A. (2002). *Hands-on PID autotuning: A guide to better utilisation*. IFAC Professional Brief.
[11] Li, X., & Li, G. (2021). An Adaptive CPU-GPU Governing Framework for Mobile Games on big.LITTLE Architectures. *IEEE Transactions on Computers*, *70*(9), 1472–1483. https://doi.org/10.1109/TC.2020.3012987
[12] Miao, Z., Shao, C., Li, H., & Tang, Z. (2025). Review of Task-Scheduling Methods for Heterogeneous Chips. *Electronics*, *14*(6), 1191. https://doi.org/10.3390/electronics14061191
[13] Panchal, V. (2024). Mobile SoC Power Optimization: Redefining Performance with Machine Learning Techniques. *International Journal of Innovative Research in Science, Engineering and Technology*, *13*(12). https://doi.org/10.15680/IJIRSET.2024.1312117
[14] Panchal, V. Thermal and Power Management Challenges in High-Performance Mobile Processors. *International Journal of Innovative Research in Science, Engineering and Technology*, *13*(11). https://doi.org/10.15680/IJIRSET.2024.1311014
[15] Pasricha, S., Ayoub, R., Kishinevsky, M., Mandal, S. K., & Ogras, U. Y. (2020). A Survey on Energy Management for Mobile and IoT Devices. *IEEE Design & Test*, *37*(5), 7–24. https://doi.org/10.1109/MDAT.2020.2976669
[16] Pathania, A., Jiao, Q., Prakash, A., & Mitra, T. (2014). Integrated CPU-GPU Power Management for 3D Mobile Games. *Proceedings of the 51st Annual Design Automation Conference*, 1–6. https://doi.org/10.1145/2593069.2593151
[17] Qiang, J., & Mitchell, C. (2014). A unified differential evolution algorithm for global optimization. *IEEE Transactions on Evolutionary Computation*, *18*(1), 1–14.
[18] Shukla, P., Coskun, A. K., Pavlidis, V. F., & Salman, E. (2019). An Overview of Thermal Challenges and Opportunities for Monolithic 3D ICs. *Proceedings of the 2019 Great Lakes Symposium on VLSI*, 439–444. https://doi.org/10.1145/3299874.3319485
[19] Stepanets, O., & Mariiash, Y. (2018). Analysis of influence of technical features of a pidcontroller implementation on the dynamics of automated control system. *Eastern-European Journal of Enterprise Technologies*, *3*(2 (93)), 60–69. https://doi.org/10.15587/1729-4061.2018.132229
[20] Zhang, A., Rau, P.-L. P., Ma, Z., Gao, Q., & Dong, L. (2019). Does Heat Matter in Phone Usage? Antecedents and Consequences of Mobile Thermal Satisfaction. In P.-L. P. Rau (Ed.), *Cross-Cultural Design. Culture and Society* (Vol. 11577, pp. 421–431). Springer International Publishing. https://doi.org/10.1007/978-3-030-22580-3_31

M.Sc. Danylo Zinchenko
e-mail: d.d.zinchenko@kpi.ua

Danylo Zinchenko is a Ph.D. student at NTUU "Igor Sikorsky Kyiv Polytechnic Institute", Institute of Nuclear and Thermal Energy, Department of Energy Processes Automation.
His research focuses on thermal-aware power (frequency and voltage) control in mobile SoCs. He is involved in scientific work on intelligent control systems for energy efficiency in electronic devices.

https://orcid.org/0000-0001-5524-6938

Ph.D. Yurii Mariiash
e-mail: y.mariiash@kpi.ua

Associate professor at NTUU "Igor Sikorsky Kyiv Polytechnic Institute", Institute of Nuclear and Thermal Energy, Department of Energy Processes Automation. The topic of his scientific research is advanced control system and automatization. The research focuses on implementation issues for control algorithm and intends to present easy ways of implementing them in industry.

https://orcid.org/0000-0002-0812-8960