MethodsX 13 (2024) 102873

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/methodsx

Peltier-based temperature regulation: A method for performance optimization in solid-state lasers☆,☆☆

Felipe Flores Montalvão[a], Iago Carvalho de Almeida[a], Vinícius Pereira Pinto[a,*], Bruno Pereira de Oliveira[a], Fátima Maria Mitsue Yasuoka[a,b], Jarbas Caiado de Castro Neto[a]

[a] São Carlos Institute of Physics, University of São Paulo, PO BOX 369, São Carlos, 13560-970, SP, Brazil
[b] BR Labs Tecnologia Óptica e Fotônica Ltda, Juan Lopes St., 222 - Jardim São João Batista, São Carlos, 13567-020, SP, Brazil

ARTICLE INFO

Method name:
Simple & Efficient Temperature Control of a Solid-State Laser

Keywords:
Temperature control
Peltier cooling
Proportional-integral controller
Laser crystal
Thermal regulation

ABSTRACT

This article presents a direct method for temperature control in solid-state lasers, where temperature stability is crucial for optimizing the performance and reliability of such lasers. The proposed method utilizes Peltier chips for both cooling and heating the laser crystal to achieve precise temperature regulation.

The system design is based on the step response of the open-loop thermal system and employs a proportional-integral (PI) controller for closed-loop temperature control. Comprehensive testing on a femtosecond Titanium-Sapphire Laser (Ti:Sapphire laser) demonstrated that the system is capable of maintaining the desired operating temperature with remarkable stability and efficiency, highlighting its practicality for real-world applications.

Method Outline:

- Utilization of Peltier chips for precise temperature control.
- Estimation of first-order transfer function based on step response.
- Implementation of a proportional-integral (PI) controller for closed-loop temperature regulation.

Specifications table

Subject area:	Engineering
More specific subject area:	*Photonics and Optoelectronics*
Name of your method:	Simple & Efficient Temperature Control of a Solid-State Laser
Name and reference of original method:	Automated PID Tuning Method
Resource availability:	Equipment: Peltier chips, NTC Thermistors. Software: MATLAB, Arduino platform, Autodesk Eagle. Hardware: Arduino Nano ATMega328.

☆ **Related research article**: None
☆☆ **For a published article**: None

* Corresponding author.
E-mail address: viniciusppinto@usp.br (V.P. Pinto).

https://doi.org/10.1016/j.mex.2024.102873
Received 7 May 2024; Accepted 19 July 2024
Available online 21 July 2024

Background

Solid-state lasers function by pumping energy into crystals, which serve as gain media for amplified light production. Both the pumped power and intracavity power significantly increase the crystal's temperature, altering its refractive index and subsequently affecting laser alignment and efficiency. Therefore, a precise temperature control system is essential to maintain crystal stability and ensure optimal laser performance amidst these temperature fluctuations. Although Peltier-based temperature regulation is a well-understood and widely reported method for its cost-efficiency and suitability for compact systems [1,2], its specific and detailed implementation in solid-state lasers for laser cooling has not been thoroughly documented. This method article aims to bridge this gap by detailing the unique aspects, challenges, and necessary considerations for the development and implementation of Peltier-based cooling systems tailored specifically for solid-state lasers, highlighting its advantages and addressing potential limitations compared to other cooling methods.

Method details

The forthcoming methods section stems from a project focused on the temperature control of a femtosecond Titanium-Sapphire laser cavity [3]. This endeavor was motivated by the objective of achieving precise temperature regulation within the laser crystal.

The strategy employed Peltier chips, Arduino Nano microcontrollers, NTC thermistors, and power drivers, all systematically integrated to construct an efficient temperature control system for the laser. The system implemented a customized version of the Automated PID Tuning Control method, leveraging the capabilities of the *Matlab Root Locus Design Tool* to optimize temperature regulation [4]. The following section will offer a comprehensive breakdown of the methodological steps detailing the design, implementation, and validation processes.

This ensures the method is reproducible and robust.

Materials and methods

Before delving into the step-by-step description of the temperature PI controller synthesis method, the materials and tools used in the construction of the control system are first outlined, along with an explanation of their specific roles in the assembly process.

Following this, several necessary implementations for assembling the control system will be detailed, setting the stage for the comprehensive explanation of the temperature control methodology in the subsequent sections.

Peltier thermoelectric cooler

The configuration of the cooling system can be achieved by following these steps:

1. **Peltier module setup:** A Peltier module such as the TEC1–12,706 can be utilized. It comprises 127 semiconductor components in a single layer and supports a maximum electrical current up to 6 A. The module can be affixed to a high thermal conductivity metal base, such as copper [5], which provides structural support for the crystal. Copper is preferred as the base structure due to its desirable properties for thermally efficient and durable heat exchangers, such as its high thermal conductivity of 402 W $\cdot\, m^{-1} \cdot K^{-1}$ (300 K) [6]. The initial configuration, depicted in Fig. 1, may include an aluminum finned heatsink along with the Peltier module, securely fastened onto the copper base.
2. **Heatsink integration:** The heatsink is properly integrated with the hot side of the Peltier module. This step is vital for generating a heat flux. Without the heatsink, efficient heat exchange may be disrupted, potentially leading to overheating of the Peltier module and damage to its semiconductor elements. The use of thermal paste is necessary for coupling the equipment, ensuring efficient heat transfer; however, care should be taken to prevent any unwanted residue that could potentially contaminate or affect the optical components of the laser.
3. **Enhanced heat dispersion:** The integration of a fan with suitable dimensions, for instance, 80 mm × 80 mm, can enhance heat dispersion into the surrounding environment. This fan size is chosen to fit onto the heat sink, enabling it to cover the entire surface of the hot side of the Peltier module effectively. This additional component aids in maintaining optimal operating conditions and preventing excessive temperature buildup within the system.

Arduino Nano and sensors

In this method, an Arduino Nano board, utilizing the ATmega-328 microcontroller architecture, functions as the programmable platform for the effective management of system sensing and control logic operations.

The Arduino Nano was selected for its open-source nature and cost-effectiveness, contributing to an economical control system. Its compact size and adequate processing power render it well-suited for the intended application, thereby providing an overall efficient solution [7].

This method employs two NTC 3950 100 kΩ thermistors connected to the Arduino. NTC (Negative Temperature Coefficient) sensors are characterized by an inversely proportional relationship between temperature variation and resistance, meaning that an increase in temperature results in decreased resistance and vice versa [8].

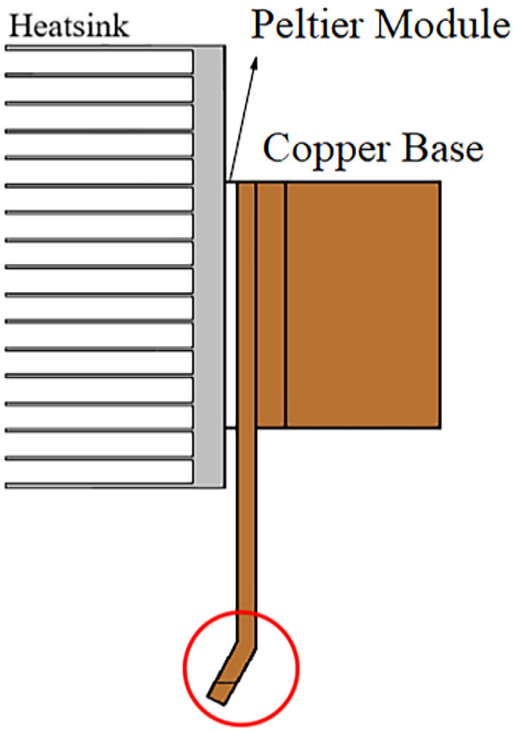

Fig. 1. Recommended Configuration of Heatsink Assembly, Copper Base, and Peltier Module. The red mark indicates the location of the titanium-sapphire crystal encased in the copper structure. In this case, the crystal is subjected to the incidence of the 532 nm green pump laser [3].

The sensors, encapsulated in glass and covered by a platinum film, can operate across a wide temperature range of −40 ˚C to 270 ˚C [9]. This broad operating range, coupled with their compact size, makes them suitable for monitoring a wide spectrum of temperatures in various applications.

One thermistor serves as the primary sensor for data acquisition, capturing the electrical current output value that represents the crystal's temperature. The second thermistor functions as a reference, measuring the ambient temperature for comparative purposes. Monitoring the temperature of the surrounding environment is crucial as it provides valuable context for evaluating the data from the primary sensor.

The temperature is measured using a voltage divider method, which involves determining the thermistor's resistance by analyzing the electrical current. The resulting voltage is then read using an analog input on the Arduino. Configure this input port to connect to the appropriate sensor.

In the utilization of NTC thermistors with Arduino, it is imperative to establish a relationship between temperature and resistance. This is due to the fact that the Arduino measures the voltage drop across the thermistor resistance, a value that fluctuates with temperature. This correlation is encapsulated by the β-parameter equation (Eq. (1)) a variant of the Steinhart-Hart equation, which is frequently employed in conjunction with NTC thermistors [10].

Consequently, the resistance value is incorporated into the approximate Steinhart-Hart equation (Eq. (1)), which subsequently computes the corresponding temperature value.

It is important to note that the Steinhart-Hart equation yields results in Kelvin. Therefore, a conversion to degrees Celsius is necessary for suitable interpretation.

The calculation of temperature is a vital element in the accurate control of the temperature regulation system.

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta} \ln \frac{R}{R_0} \tag{1}$$

In this equation, T represents the measured temperature, T_0 is the nominal temperature of the thermistor, β is a coefficient specific to the thermistor determined experimentally by the manufacturer that indicates the variation of the thermistor's resistance as function of temperature, R_0 is the resistance at the nominal temperature, and R is the resistance of the electrical current at the temperature T.

Current monitoring

For precise monitoring, the ACS712 invasive current sensor, operating on the principle of the Hall effect, is recommended [11]. This sensor induces a potential difference in an auxiliary conductor, corresponding to the magnetic field encompassing the primary conductor.

The sensor generates a potential difference within an internal conductor, instigated by the magnetic field resulting from the current being measured.

The voltage derived from this process is directly proportional to the electric current, with the proportionality constant delineated in the manufacturer's datasheet. This sensor offers a reliable technique for accurate current measurement, significantly contributing to the precise evaluation of performance.

Analogic regulation of temperature

To begin the implementation and ensure reliable outcomes, integrate a potentiometer into the system to set the desired temperature reference point. The potentiometer enables users to define the target temperature for system control. While this is a straightforward solution, the reference temperature can also be configured in other ways such as using a digital thermostat, that may be more practical or robust, depending on preference.

LCD display

Integrating a user interface for real-time feedback is crucial. An LCD display is recommended for this task. It should display both the thermistor outputs and the reference temperature for system control. Additionally, consider displaying information about the conducting and energy consumption assessment of the Peltier electrical module.

While an LCD display is commonly used for real-time feedback, it's not the only option. Alternatives such as LED displays, computer-based graphical user interfaces (GUIs), or web-based interfaces may be more suitable depending on your system's specific needs.

Manufacture of the printed circuit board (PCB)

The Supplementary Material section will include an appendix containing the schematic diagram (Appendix A) and PCB layout (Appendix B) for the integration of these components. To ensure proper assembly and functionality, follow these step-by-step instructions and recommendations:

1. **Initial breadboard testing:** Begin by assembling the entire system on a breadboard for preliminary testing. This step aids in identifying potential issues, allowing for effective troubleshooting before proceeding further. At this stage, it is equally recommended to simulate the initial circuit using a software of your choice.
2. **Fabricate a PCB:** Upon successful breadboard testing, proceed with the fabrication of a PCB. This board will house the control system along with the optical cavity's lens system, providing a more stable and permanent setup for the components.
3. **Connect to external power supply:** For the power supply, connect both the Arduino and the Peltier module to an external 12 V switched-mode power supply. Ensure that the power supply can deliver up to 10 A to provide the necessary current for smooth system operation.
4. **Integrate voltage regulator:** To prevent potential overload to the Arduino and other components, integrate a L7805CV voltage regulator with the switched-mode power supply. This regulator maintains a dedicated 5 V level, ensuring safe and stable connections for peripheral devices.

Instead of using a 12 V power supply with an L7805CV voltage regulator, users have the option to directly choose a 5 V power supply for the peripheral components. Alternatively, a buck converter circuit could be used, which is more efficient than a voltage regulator. A buck converter is a power electronics circuit that efficiently steps down voltage from its input to its output. It does this by storing energy in an inductor and then releasing it to the load at a lower voltage, making it a practical choice for power-sensitive applications [12].

Charging and pulse width modulation

To effectively power the Peltier module, use the control signal generated by the implemented PI controller on the Arduino through Pulse Width Modulation (PWM).

This step is crucial for the control system's ability to request varying voltage levels, which in turn influence the temperature adjustments. While the Arduino's digital outputs can function as PWM, certain factors warrant the use of a charge pump for the Peltier module, rather than a direct connection to the microcontroller.

Given that the maximum output voltage level of Arduino pins is 5 V, the voltage across the terminals of the Peltier module would not be high enough to generate the necessary electrical current for a substantial temperature gradient. Furthermore, the variation of the electrical current could jeopardize the Arduino's output pins, which are constrained to provide a maximum of 40 mA.

The driver circuit, as depicted in the schematic representation in Fig. 2, accommodates the Arduino's maximum output voltage of 5 V. We employ an LM358 operational amplifier in a non-inverting configuration with a gain of two. This setup extends the operating range to a maximum of 10 V.

The operational amplifier biases a power transistor TIP122 through its base. Connected to the 12 V switched-mode power supply via the collector, this transistor provides a current proportional to the bias to drive the Peltier module.

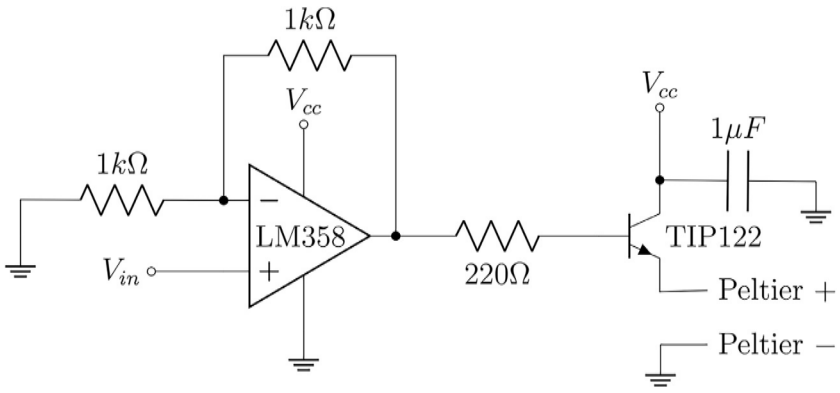

Fig. 2. Power driver with TIP122 transistor.

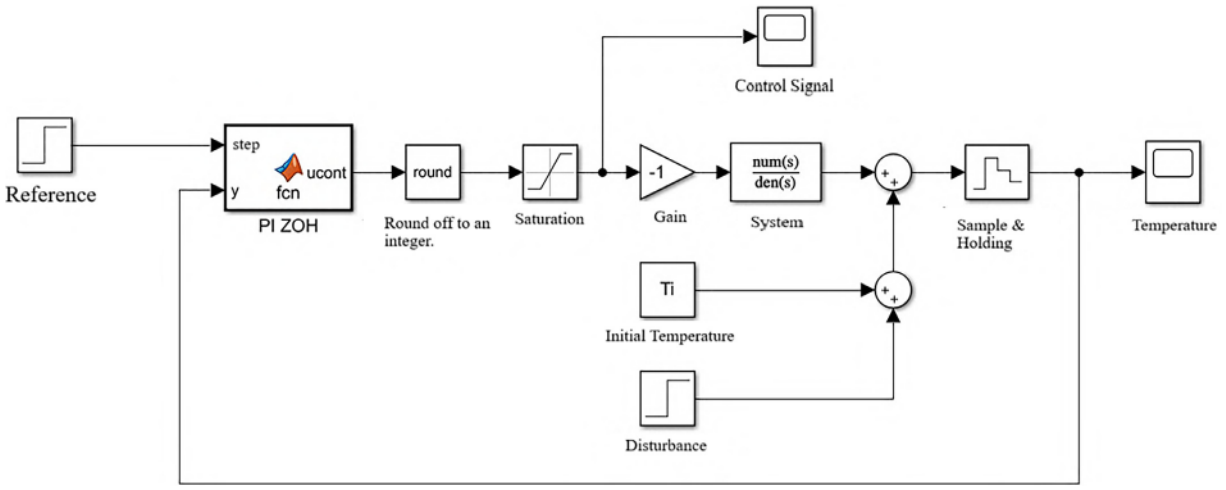

Fig. 3. Closed-loop system block diagram [3].

It is essential to note that the voltages indicated as V_{cc} in the schematic correspond to the 12 V power supply voltage, and the V_{in} corresponds to the output of an Arduino analog pin. This should vary from 0 to 5 V in accordance with the required control signal [7].

Please be aware that the TIP122 transistor must not be directly employed on the breadboard due to the high current flowing through the collector and emitter, which could damage the breadboard tracks. Therefore, you must assemble the transistor separately with its own heatsink during the initial breadboard testing, and safeguard its connections with heat-shrink tubing.

Simulation methodology

For an optimal method implementation, it is advisable to begin by conducting simulations in a software environment, utilizing the open-loop transfer function that has been acquired. For instance, the MATLAB System Identification toolbox can be used for this purpose. During these simulation tests, consider using the featured block diagram, as it simplifies the mathematical model and reduces the complexity of the equation (see Fig. 3) [13].

The closed-loop control system, depicted in the block diagram by the **PI ZOH** block, contains the code segment responsible for executing the control logic. This code should be compiled in the Arduino IDE using a C/C++ script language and executed by the microcontroller.

It is necessary to round the output signal to the nearest integer (**Rounding to integer** block). This is crucial due to the discrete nature of the 8-bit PWM output from the Arduino, which ranges from 0 to 255 and corresponds to an average output voltage of 0 to 5 V.

To simulate the behavior of the Arduino's PWM, incorporate a **Saturation** block during script development. This will prevent values from exceeding 255 or becoming negative in the corresponding Arduino output, thus mitigating errors and potential improper system behavior.

The inversion of the control signal through a gain is necessary due to the step response of the open-loop system being a decreasing curve, and the controller design being carried out with a normalized and increasing version (**−1 gain** block). The control signal serves as an input to the open-loop transfer function of the system (**System** block), which characterizes the power driver, the Peltier

chip, and the copper base. You might also consider introducing disturbances (**Disturbance** block), such as simulating increasing temperatures, over the course of the simulation time. This could allow you to verify the effect of the null steady-state error due to the integrator of the PI controller.

The output temperature signal should be sampled and held, as it occurs in practice through the Arduino's A/D converter for reading the temperature from the crystal thermistor. The **Sample & Holding** block simulates this. The output signal can also be used for feedback that produces the error signal within the **PI ZOH** block.

Through **Scopes**, the control signal and the output signal can be analyzed for temperature control system validation (**Control Signal** and **Temperature** blocks). After obtaining the desired results, the system can be set up and tested experimentally. This approach provides a comprehensive understanding of the system's performance and effectiveness.

The **Initial Temperature** block provides the initial conditions of the system. This is a crucial component as it sets the starting point for the system's operation and influences how the system responds to subsequent inputs. It's important to set this accurately to ensure the system operates as expected.

Approach for establishing control logic

1. **System dynamics:** The open-loop dynamics of the thermal system are ascertained by applying a voltage step to a Peltier module and measuring the step response of the crystal temperature. This response enables the approximation of the system's transfer function to a first-order system using MATLAB's `System Identification toolbox`.
   The step input is approximately 2.51 V, which corresponds to a digital output value of 128 using the Arduino Nano's PWM. The control signal, represented by 8 bits, can assume any value from 0 to 255. This range of values corresponds to a voltage amplitude that varies from 0 to 5 V.
   The system's step response is derived by processing the measured data. This can be achieved through various methods such as using Python or C++ code to store and process results, or employing MATLAB for reading data from the serial port, among others.
   This processed data represents the temporal variation of the crystal temperature when a constant voltage is applied to the Peltier chip. The step response is normalized using the System `Identification toolbox` to estimate the open-loop system transfer function. The input signal is a step of value 128, and the output signal consists of the measured temperatures. The sampling time $T_0 = 1.315$ s is determined by the execution time of a single loop of the code in Arduino.
2. **Controller design:** Use MATLAB's rltool to design a PI controller in the Laplace complex frequency domain. Adjust the controller parameters to influence the stability and speed of temperature control according to your specific needs and preferences.
3. **Digital control:** In the Supplementary Material section (Appendix C), a technique for discretizing the transfer function of the PI controller obtained in the rltool is presented. This involves the use of digital control techniques to discretize the controller for digital implementation in discrete time through the corresponding difference equation.
4. **Simulation and implementation:** First, test the digital control logic through Simulink block diagrams, then implement it in C/C++ language using the open-source Arduino Nano platform. The algorithm for this can be found in the Supplementary Material section (Appendix D).

Method validation

Simulation results

To test the designed PI controller, Simulink was used to simulate the closed-loop system results for a reference temperature of 15 °C. The system block diagram can be seen in Fig. 4. The **Rounding to integer** block is responsible for converting the controller output to an integer, making it compatible with the Arduino's digital logic. The **Saturation** block prevents a request for an output greater

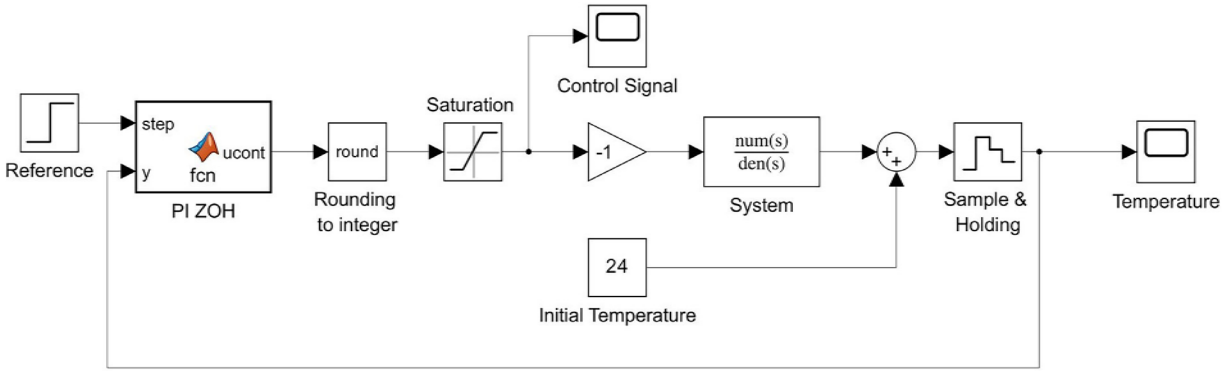

Fig. 4. Block diagrams for simulation.

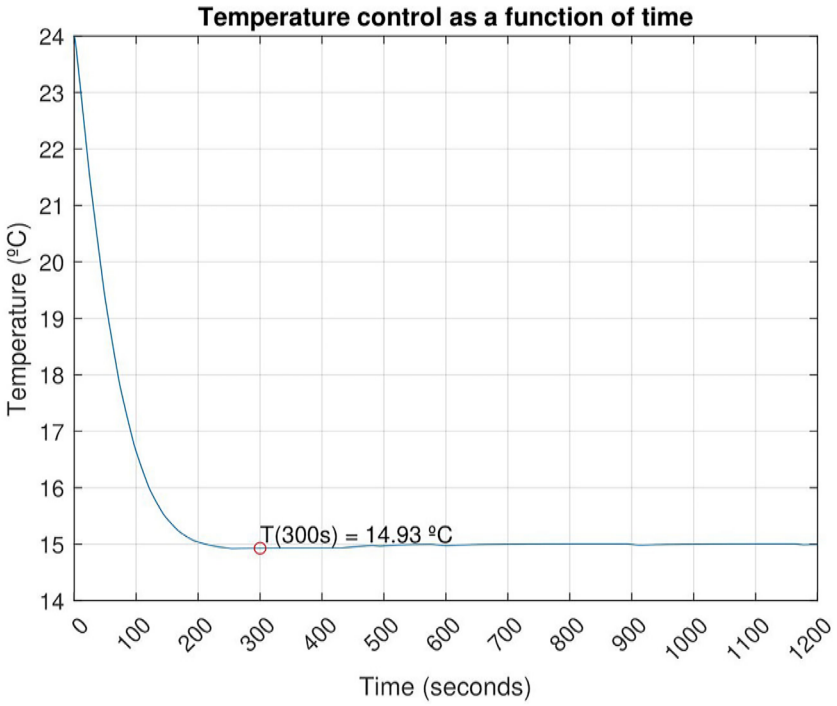

Fig. 5. Crystal temperature simulated. The graph depicting the controlled temperature as a function of time showcases the results of the Simulink simulation for temperature control. Starting from an initial temperature of 24ºC, the simulation does not account for potential disturbances generated by the laser. It can be observed that the reference temperature of 15ºC is reached approximately after 200 s. The red mark indicates the simulated temperature of the titanium-sapphire crystal after 300 s, demonstrating the effectiveness of the control system in maintaining the desired temperature over time.

than 5 V or negative from the Arduino, protecting the system in case of algorithm failure. The **negative Gain** and the **summation** of the initial temperature reverse the system normalization. The **System** block contains the transfer function $H(s)$ of the normalized system. The **Sampling & Holding** block performs the discretization of the output, and the **PI ZOH** block executes the control logic according to the code shown below. This logic is based on the difference equation obtained for the designed controller (Section b).

```
function ucont = fcn ( step , y )
persistent u0 ;
persistent e0 ;
if isempty ( u0 )
      u0 = 0;
      e0 = 0;
end
e = y - step ;
ucont = u0 + 5.569* e - 5.326* e0 ;
e0 = e ;
u0 = ucont ;
```

The graph depicted in Fig. 5 presents the simulated output temperature as a function of time, derived from the block diagram simulated in Simulink. This graph considers the previously mentioned parameters, offering a comprehensive visualization of the system's behavior over time.

Fig. 6 presents the steady-state error in the simulated temperature, computed from the 200-second mark. Each bar depicts the absolute error relative to the reference temperature, offering a detailed perspective of the system's performance in a steady state.

This graph is instrumental in promptly discerning trends and discrepancies between the simulated and reference temperatures. Substantial fluctuations in the steady-state error could indicate the need for controller adjustments or system enhancements. The analysis of steady-state error is vital for guaranteeing the stability and precision of the temperature control system, providing valuable insights for system optimization.

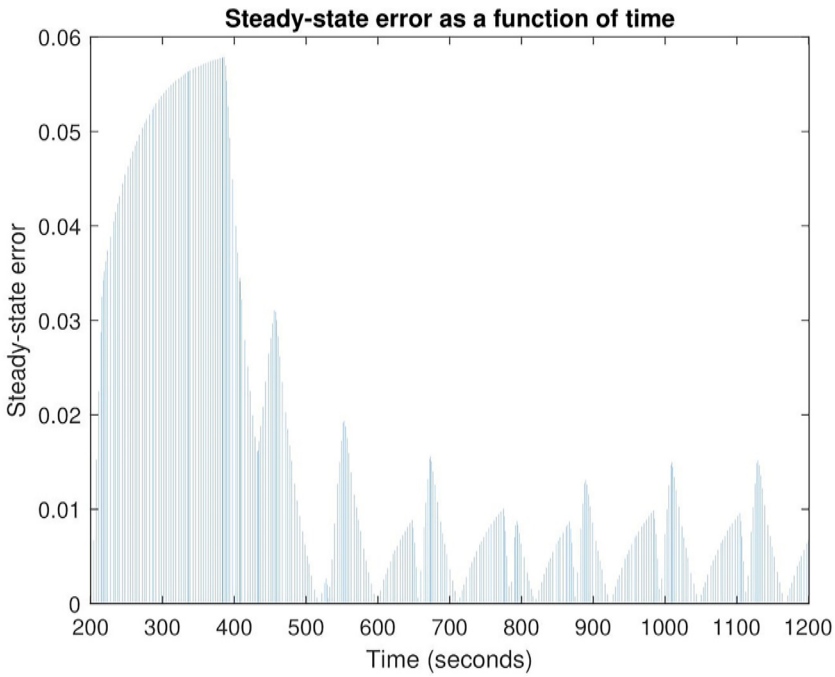

Fig. 6. Steady-state error graph. The visual representation depicts the absolute error between the simulated and reference temperatures, starting from the 200-second mark. A detailed examination of these bars provides critical insights into the precision of the temperature control system under steady-state conditions.

In the real-world laboratory experiments, a 2 W laser operates as a disturbance. However, this disturbance is absent in the simulation of the temperature control system. As a result, even though the reference temperature is achieved in both situations, the active operation of the laser in the real-world experiments leads to a longer duration to reach the reference temperature.

Experimental results

Fig. 7 presents an analysis of temperature dynamics based on thermocouple measurements in two distinct scenarios. The blue circles denote the experimental data for the titanium-sapphire crystal, capturing temperature fluctuations over a 300-second interval with the laser operating at a power of 2 W. Quadratic polynomial fits, represented by the solid lines, were applied to model these dynamics.

In the top panel of Fig. 7, the temperature profile for Measurement-1 is depicted. The blue circles correspond to the actual thermocouple data, and the solid line represents the quadratic polynomial fit. The associated error bars indicate the variability around the fitted curve, providing insights into the model's precision. The coefficient of determination (R^2) quantifies the goodness-of-fit, indicating the extent to which the polynomial captures the observed temperature changes. The R^2 value for Measurement-1 is 0.99494.

In the bottom panel of Fig. 7, a similar analysis is conducted for Measurement-2. The blue circles again represent the thermocouple measurements, and the solid line denotes the quadratic polynomial fit. The error bars express the uncertainty associated with the model. The R^2 value for Measurement-2 is 0.97972.

The experimental results, obtained within a 300-second interval, are in alignment with the simulated outcomes for an initial temperature of 24ºC. However, it is worth noting that the experimental values display some divergence from the simulation. This discrepancy can be ascribed to the absence of a disturbance, introduced by a laser operating at a power of 2 W, in the simulation. As a result, the reference temperature is achieved at 200 s in the simulation, while in the experimental setup, the reference temperature of 15ºC is reached at approximately 300 s.

Impact of temperature control on laser power

In order to validate the methodology and evaluate the efficacy and importance of temperature control, measurements of laser power were carried out with the temperature control systems both activated and deactivated. The outcomes are illustrated in Fig. 8.

The graph compares the temporal evolution of laser power under two conditions: with the Peltier cooling system (solid curve) and without cooling (dashed curve). The initial decline in power in both scenarios is linked to crystal heating, succeeded by power stabilization. Notably, the data presented are in the form of moving averages, providing a refined depiction of power dynamics. The Peltier cooling system demonstrates enhanced power stabilization, achieving lower crystal temperatures during thermal equilibrium.

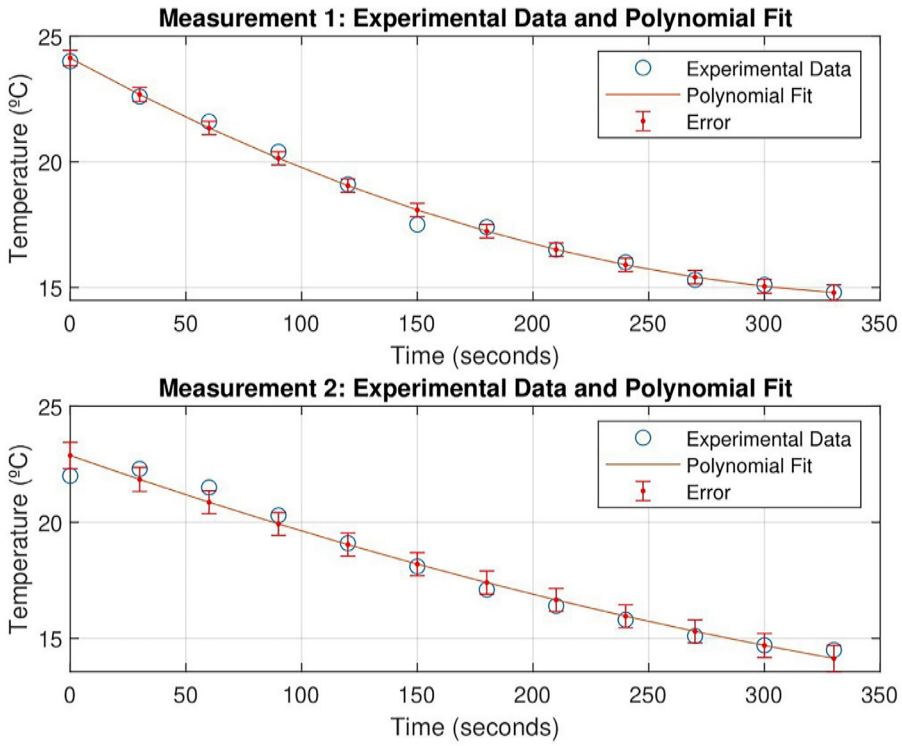

Fig. 7. Thermocouple measurements (represented by blue circles) capture the dynamics of temperature over a span of 300 s. Quadratic polynomial fits (depicted by solid lines) offer a perspective on modeling. Error bars (in red) denote the uncertainty of the fit. The R^2 values for Measurement-1 and Measurement-2 are 0.99494 and 0.97972, respectively.

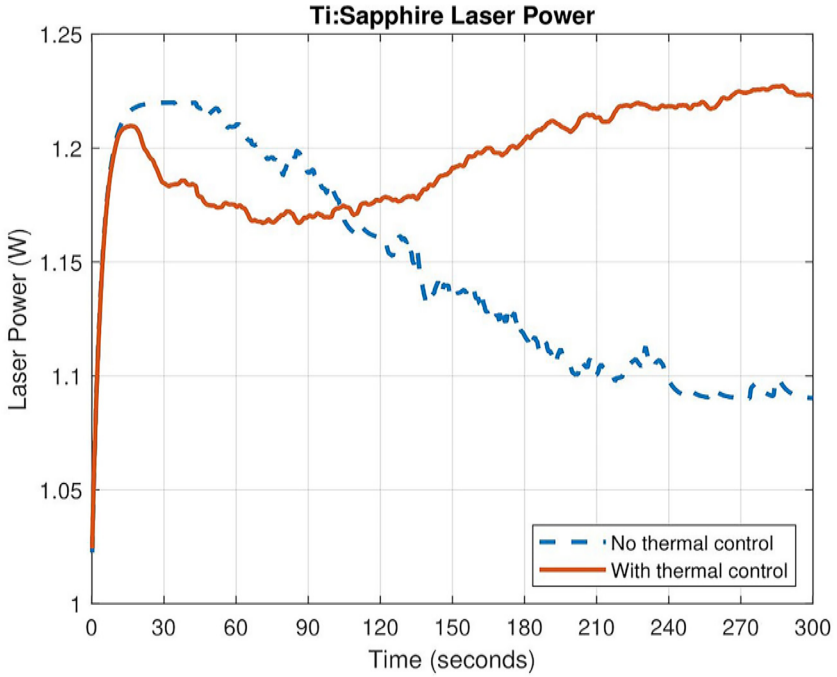

Fig. 8. The **temporal evolution of the moving average of laser power** with the Peltier cooling system (represented by the solid curve) and without cooling (represented by the dashed curve) is depicted. The initial drop in power in both scenarios is attributed to the heating of the crystal, which is subsequently followed by power stabilization. The Peltier cooling system demonstrates more efficient power stabilization by maintaining lower crystal temperatures when the system reaches thermal equilibrium.

Limitations

None.

Ethics statements

This work adheres to ethical guidelines outlined by MethodsX. No human subjects, animals and data from social media platforms were involved in this study.

Credit author statement

Felipe Flores Montalvão: Validity Testing, Data Curation, Writing - Original Draft Preparation, Visualization, Investigation, Reviewing and Editing. **Iago Carvalho de Almeida:** Conceptualization, Software Implementation, Validity Testing, Data Curation. **Vinícius Pereira Pinto:** Conceptualization, Reviewing and Editing. **Bruno Pereira de Oliveira:** Supervision, Reviewing and Editing, Validity Testing. **Fátima Maria Mitsue Yasuoka:** Supervision, Reviewing and Editing, Validity Testing. **Jarbas Caiado de Castro Neto:** Supervision, Reviewing and Editing, Validity Testing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors acknowledge the support provided by FAPESP (São Paulo Research Foundation), grant number 2020/04657–8 and 2013/07276–1 (CEPOF-CEPID Program).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.mex.2024.102873.

References

[1] V.A. Semenyuk, T.V. Pilipenko, G.C. Albright, L.A. Ioffe, W.H. Rolls, Miniature thermoelectric coolers for semiconductor lasers, AIP. Conf. Proc. 316 (1) (1994) 150–153, doi:10.1063/1.46782.
[2] Q. Zhang, K. Deng, L. Wilkens, H. Reith, K. Nielsch, Micro-thermoelectric devices, Nat. Electron. 5 (6) (2022) 333–347, doi:10.1038/s41928-022-00776-0.
[3] I.C. d. Almeida, J.C. d. Castro Neto, Sistema de controle de temperatura da cavidade laser de titânio-safira em regime femtossegundo utilizando pastilhas de peltier, in: Semana Integrada do Instituto de Física de São Carlos - SIFSC, Instituto de Física de São Carlos - IFSC, 2020, p. 40.
[4] MathWorks, Root Locus Design, MathWorks, accessed: 2024-04-26 (2024). URL https://www.mathworks.com/help/control/ug/root-locus-design.html
[5] P. Nath, K. Chopra, Thermal conductivity of copper films, Thin. Solid Films 20 (1) (1974) 53–62 URL https://www.sciencedirect.com/science/article/pii/0040609074900339 , doi:10.1016/0040-6090(74)90033-9.
[6] C. Uher, in: Thermal Conductivity of Metals, Springer US, Boston, MA, 2004, p. 28, doi:10.1007/0-387-26017-X_2. Ch. 1.2.
[7] R.T. Bernardo, Desenvolvimento De Uma Plataforma Para Aplicação De Técnicas De Controle De Temperatura Por Efeito peltier, Undergraduate thesis, School of Minas, Federal University of Ouro Preto, 2015.
[8] D. Thomazini, P.U.B. de Albuquerque, Sensores Industriais – Fundamentos e Aplicações, 4th Edition, Saraiva Educação S.A., 2023 URL https://books.google.com.br/books?id=-YqwDwAAQBAJ .
[9] B.M. Linares, Development of a Low-Cost Data Logger For Monitoring Operating Parameters of Three-Phase Induction Motors, Universidade Tecnológica Federal do Paraná. Available at, 2021 http://repositorio.utfpr.edu.br/jspui/handle/1/26220.
[10] G.M. Geronymo, Development of a thermal converter with frequency output for ac-dc transfer measurements, Ph.D. thesis, National Institute of Metrology, Quality and Technology (INMETRO), Duque de Caxias, Dissertation For the Master's Program in Metrology and Quality (2017).
[11] U. Khair, A.J. Lubis, I. Agustha, M.Zulfin Dharmawati, Modeling and simulation of electrical prevenion system using arduino uno, gsm modem, and acs712 current sensor, J. Phys.: Conf. Series 930 (1) (2017) 012049, doi:10.1088/1742-6596/930/1/012049.
[12] M. Brown, 4 - switching power supply topologies, in: M. Brown (Ed.), Practical Switching Power Supply Design, Academic Press, San Diego, 1990, pp. 20–24, doi:10.1016/B978-0-08-051454-3.50007-2.
[13] P. Kungwalrut, A. Numsomran, P. Chaiyasith, J. Chaoraingern, V. Tipsuwanporn, A pid controller design for peltier-thermoelectric cooling system, in: 2017 17th International Conference on Control, Automation and Systems (ICCAS), 2017, pp. 766–770. https://doi.org/10.23919/ICCAS.2017.8204331.