# Autotuned PID for Accurate Temperature Control – A Hot Approach

**Vadim Radu[1,2],AlexandraAvram[1,2], Vlad Anghel[1], Gheorghe Brezeanu[1]**
[1]University "Politehnica" of Bucharest
[2]UNDA Technologies Romania
vadim@unda.tech, alexandra@unda.tech, gheorghe.brezeanu@dce.pub.ro

***Abstract*—This paper describes an accurate temperature control system, based on an optimized proportional – integrative – derivative (PID) algorithm. The proposed solution also includes an autotuning module which detects load changes and adjusts the algorithm parameters for optimal temperature control. The autotuned PID architecture is implemented as a control system for the heating element of a soldering station. Experimental results show proper temperature control, eliminating both temperature overshoot and ripple. Furthermore, the autotuner achieves optimal performance regardless of the heating element used, as demonstrated in the paper for two heating elements (30W and 48W).**

***Keywords*—*PID algorithm, autotuned, temperature control***

## 1. Introduction

One can observe constant evolution of technology and electronic devices, which pushes not only the boundaries of the components used to develop them, but also the tools used to build them. The soldering station is a key device for any individual, from a hobbyist to an expert in application engineering. Electronic components require special care with respect to their soldering profile, as temperature overshoot (*OS*) should be avoided at all costs in order not to damage the component and so should big ripple due to its impact on the quality of the solder joint. For achieving this task, four major technologies are available for controlling the temperature: thermostat based, bang-bang control, Pulse-Width Modulation (PWM) switching and Proportional - Integrative - Derivative (PID) [1]-[4]. While the complexity is less challenging for the first two methods, they fail to achieve a stable temperature corridor, suffering from thermal ripple. The third one has granular control over the power applied to the heating element and presents better results, but with increased costs.

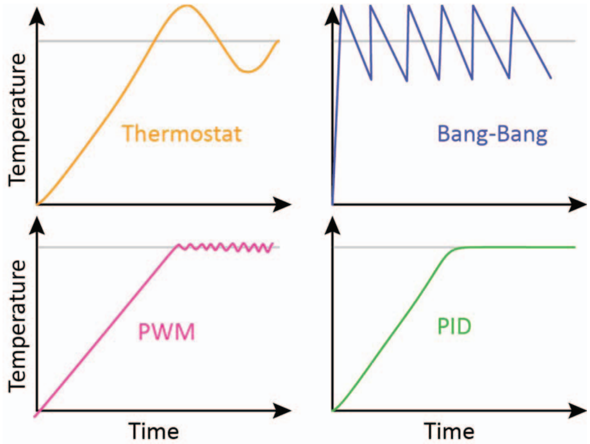


Fig. 1. Various control methods for soldering temperature.

In this paper, an alternative solution is presented, derived and improved from the classical PID, with adaptive tuning added. The proposed method eliminates the drawbacks that the aforementioned state-of-the-art solutions suffer from, while maintaining a cost and part count identical to a PWM based solution.

## 2. Proposed Architecture

Architecture wise, the proposed control system is an autotuned PID – based algorithm (AT – PID), as shown in Fig 2. For demonstrative purposes, the AT – PID system is implemented as a temperature control unit in a soldering station.

The temperature generated by a heating element is sensed via a K-type thermocouple and digitized by an analog-to-digital converter (ADC block in Fig. 2). Also, the reference temperature ($T_{REF}$) is user defined through a generic I/O block. Finally, the heating element is controlled through a PWM signal from within the AT – PID system.

The PID-based algorithm is a core component of the proposed AT – PID architecture. Fig. 3 illustrates the operating principle of the PID module [1][2]. The input of the algorithm is the digitized temperature (*T(t)*) registered by the K-

978-1-5090-1207-7/16/$31.00©2016 IEEE

type thermocouple at a given time, while the reference temperature ($T_{REF}$) is the value to be reached. The error term ($e(t)$) is calculated by subtracting $T(t)$from $T_{REF}$, in order to evaluate the difference between the measured temperature and the desired one.

The parameters $K_P$, $K_I$, $K_D$ are used to determine the sign and contribution gain of each term represented in Fig. 3, while the time $t$ refers to the current system time.

The three parameters determine the duty cycle ($D$ from Fig. 3) outputed by the PID block. $P_{TERM}$ ensures that the current temperature is moving towards the reference temperature. The differential term ($D_{TERM}$) shows how fast in time the current temperature rises towards the reference value. In case the variation is too rapid, meaning that $T(t)$ reaches $T_{REF}$ too fast, there will be an imminent temperature overshoot ($OS$).

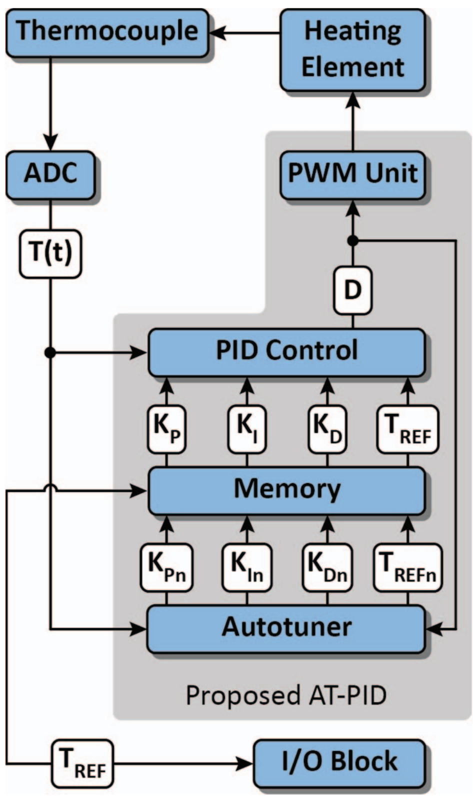


Fig. 2. Proposed AT-PID block diagram and system implementation.

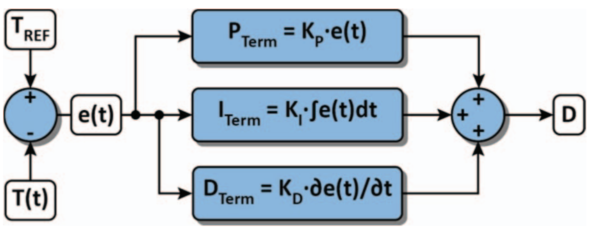


Fig. 3. Block schematic of the PID concept.

The integral term, $I_{TERM}$ is a measure of the duty cycle at equilibrium. This term takes into account all previous states of the system, playing a crucial role in deciding how the steady – states are reached.

The mathematical representation of the PID module is expressed as follows:

$$D = K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt} \qquad (1)$$

In order to improve the accuracy of the control loop, we considered adding an autotuner along with the basic PID algorithm. Would it be necessary for the heating element to be changed, the autotuning feature allows optimum temperature control. For the purpose of a clear understanding, it is imperative to mention that autotuning differs from adaptive tuning.

The autotuner is a subsystem capable of automatically determining the parameters of the PID module, during a number of working cycles [3][4]. The autotuner computes the parameters without any user interaction.

The autotuner starts by observing the heating development, comparing it to the previous profile. Should there be a difference regarding the temperature curve, a calibration routine is enabled, as illustrated in Fig. 4. During the initial calibration cycle, $T_{REF}$ is adjusted to a predefined value ($T_{REF0}$ – Fig. 4). The subsystem computes the parameters for the PID block, storing them in the memory block. At the end of this cycle the reference temperature is increased again and the next calibration cycle begins. The loop runs continuosly, incrementing $n$, until the overshoot ($OS$) and ripple ($RP$) phenomena are below predefined thresholds.

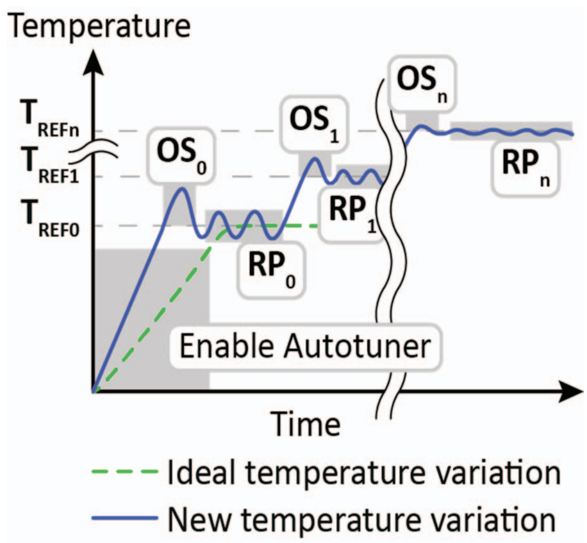


Fig. 4. Calibration sequence of the autotuner.

The autotuning subsystem is based on the Ziegler-Nichols tuning method for process variation. This method consists of detecting the peak temperature values after each change in $T_{REF}$. Convergence is achieved when the ratio between consecutive peaks is below a predefined threshold. At least 4 working cycles are required for this method.

## 3. Reduction to Practice

The proposed AT-PID architecture from Fig. 2 was implemented in an ATmega328 micro-controller [5]. The micro-controller accommodates the PID algorithm, the autotuner, the PWM unit and sufficient memory storage. The MAX6675 K-Type thermocouple sensor was also used for digitizing the temperature data required as input for the PID module [6][7]. The implementation exploits the advantages of the autotuned PID algorithm by using two heating elements (i. e.30W and 48W). This demonstrates flexibility in selecting the proper soldering pen for the required task. The only prerequisite for connecting a readily available soldering tip is to have a K-type thermocouple for correct temperature sensing.

A printed circuit board (PCB) layout was developed for the proposed AT – PID arhitecture, as shown in Fig. 5. The PCB size is 5cm x 5cm.

For testing purposes, a serial connection between the microcontroller and a PC is provided. Live data from the implemented AT – PID (current temperature, reference temperature and duty cycle) can be logged and analyzed.

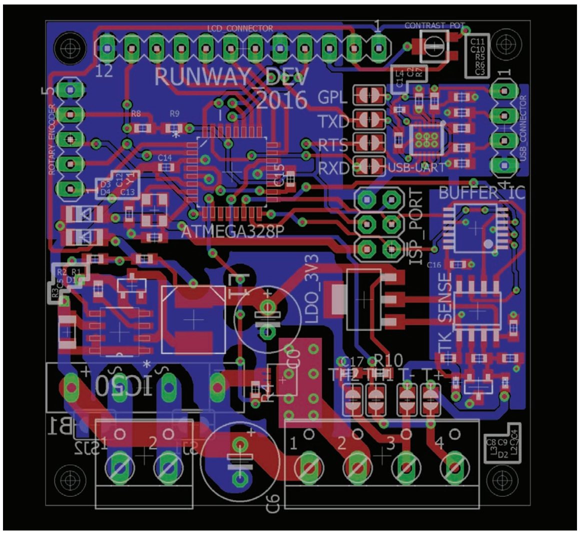


Fig. 5. PCB layout of the proposed AT – PID implementation.

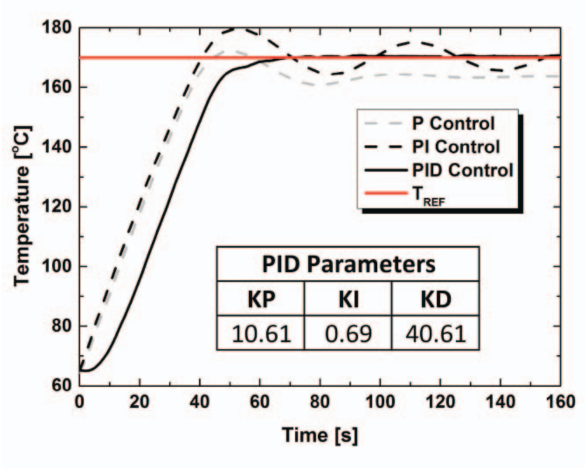


Fig. 6. Temperature control for different PID parameters using a 30W heating element.

Fig. 6 is representative for the experimental part, as it shows the variation of the temperature in three different cases: when using a P controller, a PI controller and a PID controller. The set-up included a 30W heating element. As can be observed from the plot, when using a single gain controller (only $P_{TERM}$) an overshoot appears before reaching a steady – state, which is under the desired value, $T_{REF}$. In case of a PI controller, $P_{TERM}$ remains the same as in the P controller and $I_{TERM}$ is added, the system encounters the two damaging phenomena, overshoot (*OS*) and ripple (*RP*). In spite of this phenomena, the controller achieves the reference temperature, being able to maintain stability.

When considering all parameters in the algorithm, the system reaches steady-state by having a slower rise, with no overshoot and no ripple.

Fig. 7. illustrates the functionality of the autotuning feature, after switching heating elements from 30W to 48W. The latter yielded two distinct temperature profiles before and after autotuning. The grey profile presents an undesired temperature overshoot. This behaviour is eliminated after the autotuning sequence calculated the optimum PID parameters for the 48W heating element. Fig. 7 also shows the duty cycles used by the AT – PID system to drive the heating element. The presence of an overshoot before tuning is corroborated by the additional high duty cycle between 9 and 10 seconds.

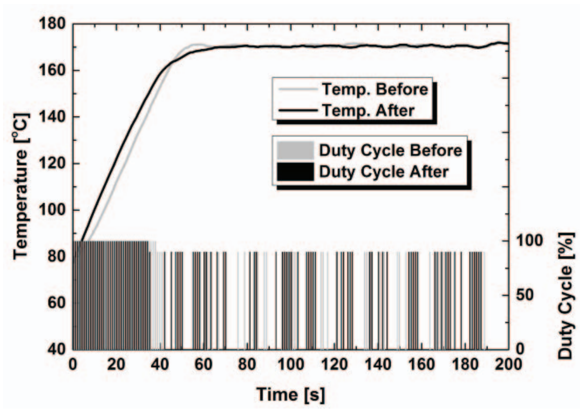


Fig. 7. AT – PID temperature profile for a 48W heating element before and after tuning.

The operation of the autotuning algorithm is illustrated in Fig. 8. As previously mentioned, when detecting a change in temperature profile, the AT – PID system enables a calibration sequence. This sequence records 4 temperature peak values within the sampling windows $At_1$ to $AT_4$ and calculates the new PID parameters. For the 48W heating element, $K_P$, $K_I$ and $K_D$ are 9.67, 0.48 and 48.19, respectively.

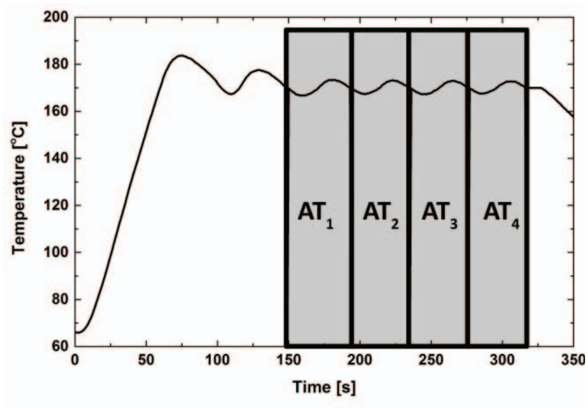


Fig. 8. The autotuning sequence for a 48W heating element.

## 4. Conclusions

This paper introduced a temperature control system based on an autotuned PID algorithm. Unlike modern temperature control techniques, such as thermostat, bang-bang or PWM, the proposed architecture provides additional flexibility when changing heating elements due to the autotuning feature.

The AT – PID architecture was implemented in an ATmega328 micro-controller. The micro-controller included a PWM unit for driving heating elements and a memory block which stored temperature profiles. Also, a MAX6675 K-Type thermocouple was employed to sense and digitize the temperature. Furthermore, a 5 cm x 5 cm sized printed circuit board was also developed for the proposed AT- PID system.

Experiments were conducted to assess the functionality of the AT – PID application, using two heating elements (30W and 48W). The 30W heating element was used to emphasize the superior results of the PID algorithm in terms of overshoot and temperature ripple, as opposed to P or PI controls. The obtained PID parameters were $K_P$ = 10.61, $K_I$ = 0.69 and $K_D$ = 40.61. When switching to the 48W heating element, the functionality of the autotuner was highlighted, as the PID parameters were adjusted to eliminate any temperature overshoot ($K_P$ = 9.67, $K_I$ = 0.48 and $K_D$ = 48.19). Considering the results reported in this paper, the AT – PID architecture, along with the optimized PID parameters, is suitable in any temperature control system for soldering stations.

## References

[1] A. Leva, C. Cox and A. Ruano, “Hands-On PID Autotuning - A guide to Better Utilisation”, [Online], Available 2016: www.ifac-control.org.

[2] K. H. Ang, G. Chong and Y. Li, "PID control system analysis, design, and technology," in IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pp. 559-576, July 2005.

[3] K. J. Astrom and T. Hagglund, “Revisiting the Ziegler–Nichols step response method for PID control”, in Elsevier Journal of Process Control, vol. 14, pp. 635-650, 2004.

[4] K. Li, "PID Tuning for Optimal Closed-Loop Performance With Specified Gain and Phase Margins," in IEEE Transactions on Control Systems Technology, vol. 21, no. 3, pp. 1024-1030, May 2013.

[5] Application Note of AVR221, Discrete PID Controller (2006), ATMEL, [Online], Available 2016: www.atmel.com.

[6] J. Shtargot and S. Mirza, “Modern Thermocouples and A High-Resolution Delta-Sigma ADC Enable High-Precision Temperature Measurement”, Maxim Integrated Application Note, [Online], Available 2016: www.maximintegrated.com.

[7] Datasheet of MAX6675, Cold-Junction Compensated Thermocouple-to-Digital Converter (2015), [Online], Available 2016: www.maximintegrated.com.