Real-Time Systems (2020) 56:1–27
https://doi.org/10.1007/s11241-019-09342-y

Analysis and elimination of noise-induced temperature error in processor thermal control

Dohwan Kim[1] · Juseung Lee[1] · Kyung-Joon Park[1] · Yongsoon Eun[1] · Sang Hyuk Son[1] · Chenyang Lu[2]

Published online: 2 January 2020

Abstract
Processor thermal control in real-time systems is crucial because overheated processors may result in critical performance degradation or even system failure. The main challenges in processor thermal control in real-time systems are as follows: (i) the need to satisfy both real-time and thermal constraints; (ii) uncertain system dynamics; and (iii) thermal sensor noise. The first two issues have been resolved in substantial studies while the issue of sensor noise has not been thoroughly addressed. In this paper, we experimentally identify that even a small zero-mean sensor noise can induce a significant steady-state error in processor thermal control. This steady-state temperature error is contrary to the intuition that zero-mean sensor noise normally induces zero-mean fluctuations around the target temperature. We rigorously analyze the phenomenon based on the stochastic averaging theory and quantify the error in a closed form in terms of noise statistics and system parameters. We propose a thermal control architecture for effectively eliminating the thermal error and tightly controlling the processor temperature. Through extensive experiments, we validate the proposed architecture.

✉ Kyung-Joon Park
kjp@dgist.ac.kr

✉ Yongsoon Eun
yeun@dgist.ac.kr

Dohwan Kim
dhkim@dgist.ac.kr

Juseung Lee
juseunglee@dgist.ac.kr

Sang Hyuk Son
son@dgist.ac.kr

Chenyang Lu
lu@wustl.edu

1 Department of Information & Communication Engineering, DGIST, Daegu, South Korea

2 Department of Computer Science & Engineering, Washington University, St. Louis, MO, USA

Keywords Processor thermal control · Real-time systems · Measurement noise · Stochastic averaging theory

1 Introduction

Thermal control in real-time systems is critical because overheated processors can result in serious performance degradation or even system failure. The main challenges in processor thermal control can be summarized as follows: (i) the need to satisfy both real-time and thermal constraints; (ii) uncertain system dynamics; (iii) thermal sensor noise.

Thermal-aware real-time scheduling schemes deal with (i) under the assumption of accurate system models. Feedback-based thermal control approaches are further proposed to take into account uncertain system dynamics, which resolves both (i) and (ii).

In the meantime, most existing studies do not explicitly consider the thermal sensor noise, which is substantial and common on microprocessors in practice (Rotem et al. 2006; Long et al. 2008). Though varied by several factors such as sensor accuracy, sensor placement, and processor architecture, the difference between sensor reading and the actual hotspot temperature can frequently be as large as 10 °C (Rotem et al. 2006; Long et al. 2008). Accordingly, we need to develop a thermal control scheme that can address *not only* (i) and (ii), *but also* the practical issue of (iii).

In this paper, we present an effective thermal control architecture for real-time systems. In particular, our key contributions are twofold.

- We rigorously analyze the noise-induced error in a general framework and quantify it in a closed form in terms of noise statistics and system parameters.
- We propose an effective thermal control architecture and *empirically* validate its performance through extensive experiments.

Our starting point is the observation that even a small zero-mean sensor noise in processor thermal control can result in a significant steady-state error. It should be noted that this steady-sate error is totally different from typical zero-mean temperature fluctuations. A non-zero steady-state error will cause either system under-utilization, or much more seriously, processor overheating: When the actual temperature is lower than the target value, the processor will be under-utilized. When the actual is higher than the target, the processor will be overheated in spite of feedback thermal control.

Furthermore, we discover that an intuitive attempt for resolving this phenomenon without fundamental understanding fails to reduce the temperature error. Hence, we need to thoroughly understand and analyze the phenomenon of a non-zero temperature error, and develop an effective solution for eliminating the error.

The remainder of this paper is organized as follows: We give related work in Sect. 2. In Sect. 3.1, as our motivation, we empirically show that a small zero-mean sensor noise can cause significant error in feedback thermal control. We further demonstrate that an intuitive effort for resolving the situation is ineffective. The system models are given in Sect. 3.2. We rigorously analyze the noise-induced temperature error and quantify it in a closed form in Sect. 3.3. In addition, we propose an effective temperature control architecture, which can eliminate the noise-induced error. Through extensive experiments, we empirically validate the proposed architecture in Sect. 4. Our conclusion follows in Sect. 5.

2 Related work

As the power density of CPU increases, thermal hotspots and severe temperature variations on the die may occur. Accordingly, thermal management mechanisms are needed to mitigate thermal problems (Kornaros and Pnevmatikatos 2013; Prakash et al. 2016; Khairnasov and Naumova 2015). In Beneventi et al. (2014), the authors present a gray box approach to system identification of thermal model to acquire more accurate thermal model. Utilization-aware power management is proposed in reliable processors that switches the power level of processors to increase energy efficiency and prevent excess of supplied power (Avirneni et al. 2016). In addition, thermal management techniques of CPU for cyber-physical systems (CPS) are recently studied in Krishna and Koren (2017).

Especially, in real-time systems, there is a need for a CPU thermal control technique that takes into account real-time constraints (Pathan 2017). To this end, thermal-aware real-time scheduling has been extensively studied (Chantem et al. 2011; Sheikh et al. 2012; Tsai et al. 2017; Zhou and Wei 2015; Zhou et al. 2016; Mohaqeqi et al. 2016; Ahmed et al. 2016). Nevertheless, these approaches typically rely on accurate system models of thermal dynamics, task execution times, power consumption, and ambient temperature. In practice, these models may not be sufficiently accurate due to various environmental changes.

There is another line of research on thermal control for real-time systems such as Saen et al. (2007) and Floyd et al. (2007), whose primary goal is to resolve the thermal issue by adjusting system parameters such as clock frequency, voltage and bandwidth allocation.

More recently, feedback thermal control methods are proposed for precluding processor overheating by tuning the task rate or CPU power. In Yue et al. (2012), a Thermal-Aware Feedback Control Scheduling (TAFCS) is proposed for soft real-time systems based on a nested feedback control loop structure called Thermal Control under Utilization Bound (TCUB) in Fu et al. (2010). The key difference between TAFCS and TCUB is that TAFCS uses a miss rate controller instead of the utilization controller in TCUB. The feedback thermal controller outputs the miss rate reference, and based on this reference value, the miss rate controller outputs the target frequency of the CPU. On the contrary, an algorithm called Temperature Regulated Capacity Bound (TRCB) is proposed in Hettiarachchi et al. (2012), whose main goal is to design thermal-resilient hard real-time systems. In Cui et al. (2017), the authors propose a highly adaptive fuzzy controller for thermal management to deal with the control quality problem according to the model uncertainty of the feedback thermal control system.

A thermal-aware service rate assignment problem in CPS is proposed in Chu et al. (2012), where a dynamic-programming based service rate assignment is introduced within a feedback control framework in order to prevent system overheating. To evaluate the proposed scheme in Chu et al. (2012), the authors introduce new performance metrics of the normalized value density, the normalized remaining power consumption, and the unfeasible ratio. A resource management for processor thermal control is proposed in Segovia et al. (2011), where the controller is based on the nested feedback controller structure that consists of the thermal controller and the resource manager, and then a service level table is defined for evaluating the proposed resource manager.

Feedback thermal control is also extended to multicore processors in Becker et al. (2014), where the authors propose two schemes of dynamic power management for thermal control of many-core real-time systems. Since neighboring CPUs can make more heat than separate ones, the proposed approaches spread the tasks in order to alleviate overheating between the cores. Feedback thermal control on multicore processors is also analyzed in Fu et al. (2012), where Real-Time Multicore Thermal Control (RT-MTC) is proposed for thermal control of multicore systems. RT-MTC controls the temperature and the utilization of multicore processors through dynamic voltage and frequency scaling (DVFS).

Another direction of research is to predict the processor temperature for improving the quality of thermal control. In Yun et al. (2013), a proactive peak temperature manager (PTM) is introduced, which periodically measures future core temperature and maintains the peak temperature of the cores below a given threshold. The goal of PTM is to make all the timing constraints of the task set satisfied in time-critical multi-core systems.

In summary, there has been no thermal control method for real-time systems that properly takes into account the impact of sensor measurement noise. One exception is the work in Kim et al. (2015), where preliminary simulation study under a limited model is reported. Here, we significantly extend the work in Kim et al. (2015) in the following two directions. First, our analysis here is carried out under a general system architecture and plant dynamics while that in Kim et al. (2015) is substantially limited to the so-called Internal Model Control (IMC) architecture with a first-order plant only. Second, we empirically validate our analytic results with a testbed while the performance evaluation in Kim et al. (2015) is carried out with simulation only.

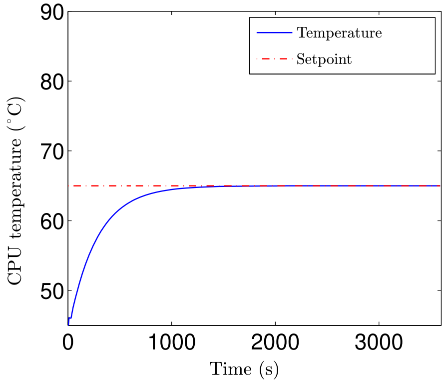
3 System modeling and analysis

3.1 Motivation: why sensor noise matters?

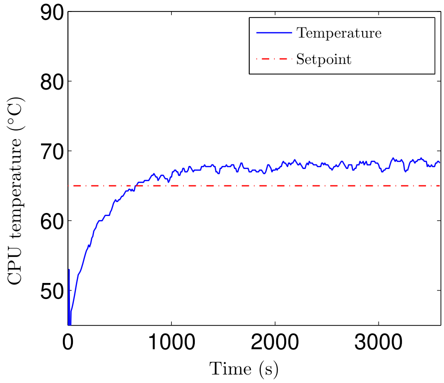
In this section, we show that random sensor noise can significantly degrade thermal control performance.

Consider the thermal feedback control of Fu et al. (2010) where utilization level is controlled based on CPU temperature feedback in order to prevent processor overheating. This method is referred to as TCUB (Fu et al. 2010). Figure 1a shows the CPU temperature from numerically simulating TCUB system when temperature sensor noise is not present. Clearly, the desired CPU temperature of 65 °C is achieved. Figure 1b shows the control performance when zero-mean Gaussian sensor noise

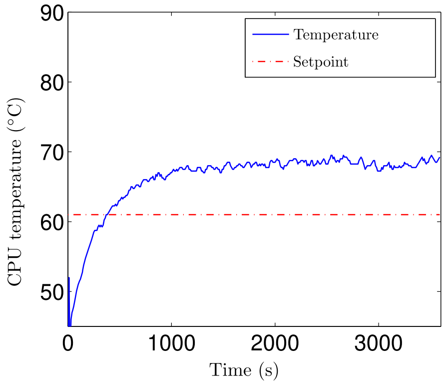
Fig. 1 Noise-induced error in feedback thermal control



(a) Thermal control in simulation



(b) Thermal control in real



(c) Thermal control response to setpoint change with variance of 1 exists in the system.[1] The CPU temperature deviates from 65 °C and fluctuates around 69 °C, which implies CPU overheating that is critical in most applications.

As it turns out, the phenomenon is not due to the particular form of TCUB (Fu et al. 2010). In fact, it takes place in any thermal control systems that achieve the target when there is no measurement noise. Detailed explanation will be given in Sect. 3.3.

It should be pointed out that this CPU temperature *bias* created by *zero-mean* sensor noise does not seem intuitively straightforward. Typically, control system outputs show zero-mean fluctuations when zero-mean sensor noise exists in the feedback measurement. Furthermore, the scenario considered in the following only emphasizes the counter-intuitive nature of the phenomenon: In response to the CPU temperature exceeding the desired value by 4 °C, one may decide to lower the setpoint, from 65 to 61 °C, in an attempt to lower the CPU temperature. The result of such an attempt is shown in Fig. 1c. The CPU temperature does not decrease as much as one would expect, and still exceeds the setpoint by more than 7 °C. This is a clear indication that an intuitive approach to remedy the situation does not work at all.

Thus, it is necessary to understand the mechanism underlying this counter-intuitive behavior in order to alleviate thermal control degradation due to sensor noise that leads to CPU overheating. In the subsequent sections, we analyze the phenomenon by modeling the thermal control systems, provide explanation and closed-form formula that predicts the level of temperature bias induced by measurement noise, and propose a solution that eliminates noise-induced temperature error. Parameters description used in the analysis of the subsequent sections can be found in Table 1.

3.2 System model and description

3.2.1 CPU thermal dynamics

We adopt the system model in Fu et al. (2010), which is introduced here for completeness. For the thermal dynamics of the processor, the well-known thermal RC model is used as follows (Ferreira et al. 2007; Bellosa et al. 2003):

$$\frac{dT(t)}{dt} = -\frac{1}{R_{\mathrm{th}}C_{\mathrm{th}}}(T(t) - T_0) + \frac{1}{C_{\mathrm{th}}}P(t), \tag{1}$$

where $T(t)$ is the temperature of the processor, T_0 is ambient temperature, $P(t)$ is the actual power consumed by the processor, C_{th} is heat capacity and R_{th} is heat resistance. Equation (1) represents equation (3.3.3) in Bellosa et al. (2003), where c_1 and c_2 are $1/C_{\mathrm{th}}$ and $1/R_{\mathrm{th}}C_{\mathrm{th}}$, respectively.

3.2.2 Overview of TCUB

The primary design objectives for feedback thermal control are twofold: (i) to prevent processor overheating, and (ii) to maintain desired real-time system performance. In

[1] Noise level with variance of 1 is a mild condition in practice (Rotem et al. 2006; Long et al. 2008).

Table 1 Parameters description

Notation	Parameters
T	Temperature of processor
T_0	Ambient temperature
P	Actual power consumed by processor
R_{th}	Thermal resistance
C_{th}	Thermal capacitance
α	Utilization lower bound
β	Utilization upper bound
U	Unfiltered utilization setpoint
U_s	Utilization setpoint
T_R	Temperature setpoint
U_R	Actual utilization of processor
R	Task rate of processor
T_s	Thermal controller sampling time
$r(k)$	kth sampled T_R
$e(k)$	kth sampled control error
$u(k)$	kth sampled control input
$d(k)$	kth sampled disturbance
$n(k)$	kth sampled sensor measurement noise

order to overcome various uncertainties in real-time systems, feedback thermal control dynamically adjusts the processor temperature so that it matches the setpoint T_R, which is specified by system requirement. In order to achieve these objectives, TCUB has been proposed based on feedback control theory (Fu et al. 2010).

The overall structure of TCUB is described in Fig. 2. TCUB has two control loops that run at different time scales. The outer loop is responsible for thermal control while the inner loop for utilization control. Since temperature dynamics is much slower than the processor utilization dynamics, it is sufficient that the outer loop runs at a lower sampling rate compared to the inner loop. At the end of the kth sampling period of the outer loop, based on the measured temperature $T(k)$ by the thermal sensor, the PI controller calculates the utilization command $U(k)$ that is checked for feasible utilization in the saturation block, and finally the thermal controller calculates the utilization setpoint $U_s(k)$ for the utilization controller of the inner loop. The role of the inner-loop utilization controller is to tune the task rates R_i's so that the actual processor utilization $U_R(k)$ converges to the setpoint $U_s(k)$ given by the thermal controller. The setpoint and the minimum and maximum utilization bounds are provided to the thermal controller at the design phase, which are denoted by T_R, α, and β, respectively.

3.2.3 Thermal control system model

Figure 3 shows the block diagram representation of TCUB feedback thermal control system (Fu et al. 2010), where $K(z)$, $G(z)$, and $H(z)$ denote the PI controller, the anti-windup controller, and the transfer function between the processor utilization and

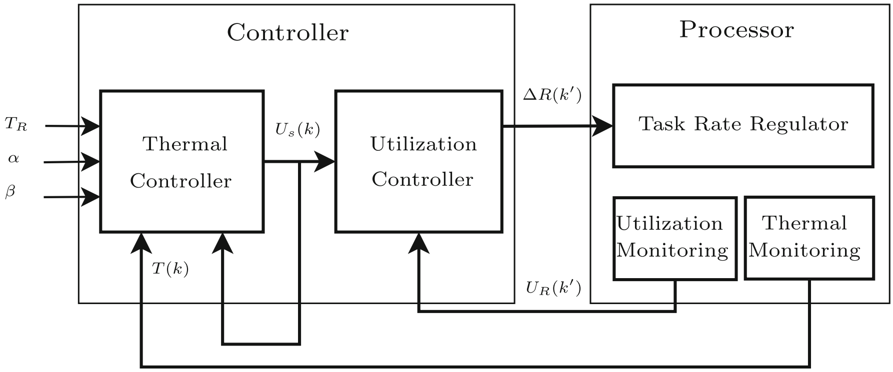


Fig. 2 Overall structure of TCUB

the CPU temperature, respectively. The upper and lower bounds of the CPU utilization in TCUB are denoted by the saturation block $\mathrm{sat}_{\alpha}^{\beta}(u)$ with the following definition:

$$\mathrm{sat}_{\alpha}^{\beta}(u) = \begin{cases} \beta, & u > \beta \\ u, & \alpha \leq u \leq \beta \\ \alpha, & u < \alpha. \end{cases} \tag{2}$$

The signals $\Delta T(k)$, ΔT_R, $U(k)$, $U_s(k)$, are, respectively, CPU temperature linearized with respect to a reference point, CPU temperature setpoint, utilization command before and after the saturation bounds. The signal $v(k)$ is the difference between $U(k)$ and $U_s(k)$ that is used as an input to the anti-windup controller $G(z)$.

The maximum utilization bound β is determined by the schedulability region within which it is feasible to meet the deadlines of the real-time tasks considered. The lower bound α can be defined as the sum of the product of each minimum achievable task execution time with the corresponding minimum allowable task rate.

The thermal controller needs to regulate the temperature of the processor to track ΔT_R by its output $U_s(k)$ under the utilization bounds. Let $T(k)$ be the temperature of the processor in the kth sampling period. Then, from Fu et al. (2010), we use $T_0 + T_{\mathrm{idle}}$ as a reference point to define $\Delta T(k) = T(k) - (T_0 + T_{\mathrm{idle}})$, where T_0 is the ambient temperature and T_{idle} is the idle temperature determined by $T_{\mathrm{idle}} = R_{\mathrm{th}} P_{\mathrm{idle}}$ with P_{idle} being the power consumption when the processor is idle and R_{th} being the heat capacity.

The transfer function $H(z)$ in Fig. 3 is

$$H(z) = \frac{\Gamma}{z - \Phi}, \tag{3}$$

where $\Phi = \exp(-T_s/(R_{\mathrm{th}} C_{\mathrm{th}}))$ with T_s being the sampling interval of the control and $\Gamma = (G_p P_a - P_{\mathrm{idle}}) R_{\mathrm{th}} (1 - \Phi)$ with G_p being the ratio between the actual active power at run time and the estimated active power P_a. The transfer function $K(z)$ is a PI controller and given by

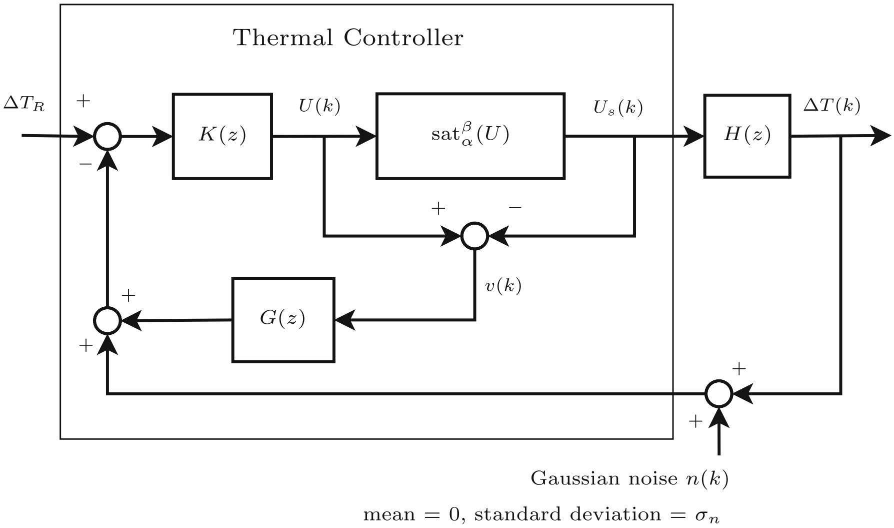


Fig. 3 Detailed view of the thermal control system with measurement noise

$$K(z) = K_p + K\frac{z-b}{z-1}, \tag{4}$$

where $K = K_i(1+\frac{w_I T_s}{2})$ and $b = \frac{2-w_I T_s}{2+w_I T_s}$ (Fu et al. 2010; Kottenstette et al. 2011). The constant K_p, K_i, and w_I are control gains from continuous dynamics. The anti-windup controller $G(z)$ is selected to be identical to $H(z)$.

3.3 Analysis of a thermal controller with sensor noise

It turns out that control performance degradation in the form of bias at the system output due to zero-mean sensor noise was first reported in Eun and Hamby (2014) for PI controlled systems with anti-windup, where the phenomenon was referred to as Noise Induced Tracking Error (NITE). The work of Kim et al. (2015) discovered that NITE also occurs in thermal control systems which are designed using Internal Model Control (IMC) structure known to have an intrinsic anti-windup mechanism. In this section we provide a NITE analysis for a larger class of feedback systems. Specifically, the novelty of the current analysis compared to those of Eun and Hamby (2014) and Kim et al. (2015) are as follows. The analysis in Eun and Hamby (2014) is limited to a PI controlled systems with specific type of anti-windup given by continuous-time dynamics. The analysis of Kim et al. (2015), on the other hand, is formulated in discrete time, and deals with an IMC type controller with only first order plant dynamics. Analysis in current work, also given in discrete time, extends the work of Kim et al. (2015) to control system architectures not limited to IMC type, and to the plant dynamics not limited to first order.

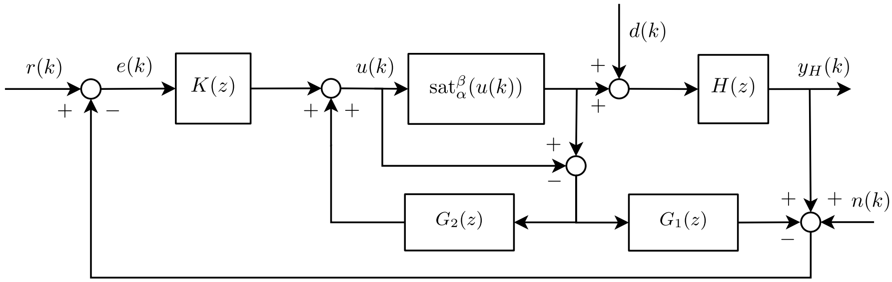


Fig. 4 A block diagram of a PI controlled system with an anti-windup with measurement noise

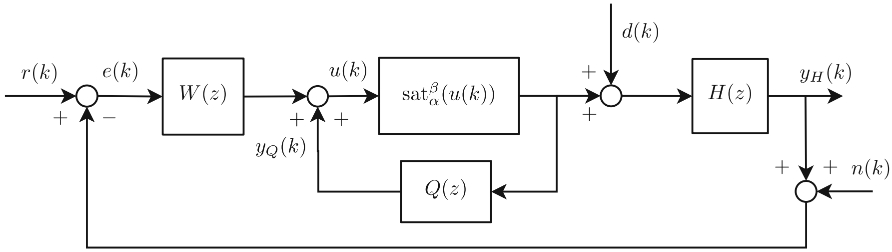


Fig. 5 A block diagram of the systems equivalent to that of Fig. 4

Figure 4 shows a PI controlled system with an anti-windup. The anti-windup consists of two transfer functions $G_1(z)$ and $G_2(z)$, and this structure is general enough to include any known anti-windup mechanism. Measurement noise process is denoted by n. The noise assumed to have zero-mean Gaussian distribution with standard deviation denoted by σ_n. The signals $r(k)$, $n(k)$, $d(k)$, $e(k)$, $u(k)$, and $y_H(k)$ are, respectively, reference, measurement noise, disturbance, tracking error, control input, and output. We notice that the thermal control systems in this paper can also be represented by the systems of Fig. 4. This can be done by setting the transfer function $G_2(z) = 0$ and $G_1(z) = G(z)$.

By manipulating the systems of Fig. 4, we can obtain the systems of Fig. 5, equivalent to that of Fig. 4 when $Q(z) = \frac{K(z)G_1(z)+G_2(z)}{1+K(z)G_1(z)+G_2(z)}$ and $W(z) = \frac{K(z)}{1+K(z)G_1(z)+G_2(z)}$.

The systems of Fig. 5 provide simplicity in terms of analysis of NITE phenomenon. Therefore, by using the systems of Fig. 5, we provide the analysis of NITE phenomenon. Here, the plant $H(z)$ and the block $Q(z)$ are strictly proper transfer functions, while the block $W(z)$ is proper one. The outputs of $H(z)$ and $Q(z)$ are denoted by $y_H(z)$ and $y_Q(z)$.

We assume the followings. The reference r and the disturbance d are assumed to satisfy

$$\alpha \le \frac{W(1)r}{1 - Q(1) + W(1)H(1)} - d \le \beta, \tag{5}$$

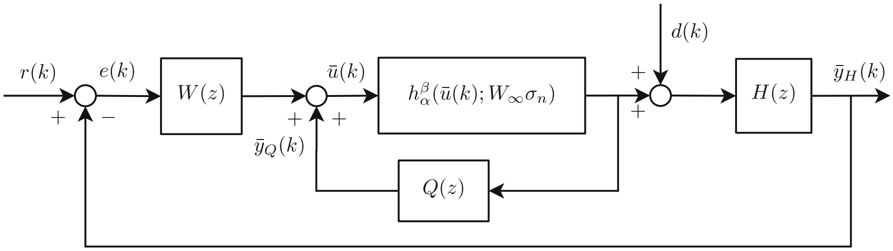


Fig. 6 An averaged version of the systems in Fig. 5

where $W(1)$, $Q(1)$, and $H(1)$ are, respectively, the dc gains of $W(z)$, $Q(z)$, and $H(z)$. Equation (5) is introduced to exclude tracking loss purely due to input saturation, and ensure that r and d are in the ranges where no steady-state tracking error occurs due to saturation (Eun and Hamby 2014). If Eq. (5) does not hold, there will be a non-zero steady-state error even without any sensor noise. This assumption is to ensure that the control input in the steady state remains in the linear region of the actuator if $n = 0$. We also assume $H(1) \neq 0$, $W(1) \neq 0$, $Q(1) \neq \infty$. Most systems physically satisfy these conditions.

According to the stochastic averaging theory (Bai et al. 1988), we apply stochastic averaging to the systems of Fig. 5, and obtain averaged systems of Fig. 6 that are driven only by deterministic signals. This approximation yields high accuracy when the noise bandwidth is high compared to that of the system. In the averaged systems, the random noise n disappears, and $\mathrm{sat}_\alpha^\beta(u(k))$ is replaced by

$$\begin{aligned} h_\alpha^\beta(\bar{u}(k); W_\infty\sigma_n) = {} & \frac{\alpha+\beta}{2} + \frac{\bar{u}(k)-\alpha}{2}\mathrm{erf}\left(\frac{\bar{u}(k)-\alpha}{\sqrt{2}W_\infty\sigma_n}\right) \\ & - \frac{\bar{u}(k)-\beta}{2}\mathrm{erf}\left(\frac{\bar{u}(k)-\beta}{\sqrt{2}W_\infty\sigma_n}\right) \\ & + \frac{W_\infty\sigma_n}{\sqrt{2\pi}}\left(e^{-\frac{(\bar{u}(k)-\alpha)^2}{2W_\infty^2\sigma_n^2}} - e^{-\frac{(\bar{u}(k)-\beta)^2}{2W_\infty^2\sigma_n^2}}\right), \end{aligned} \tag{6}$$

where

$$\mathrm{erf}(\xi) = \frac{2}{\sqrt{\pi}}\int_0^\xi \exp(-t^2)\,dt. \tag{7}$$

The parameter W_∞ means $\lim_{z\to\infty} W(z)$. As shown in Fig. 6, all signals are denoted with bar.

The effect of averaging is captured in the function $h_\alpha^\beta(\bar{u}(k); W_\infty\sigma_n)$ that replaces the actuator saturation $\mathrm{sat}_\alpha^\beta(u(k))$. The two functions are compared in Fig. 7 with the parameters $\alpha = 0.1$ and $\beta = 0.67$ for several values of $W_\infty\sigma_n$. As shown in the figure, the effect of the measurement noise manifests itself as a distortion from the saturation function and the difference between the two increases as $W_\infty\sigma_n$ increases. When σ_n

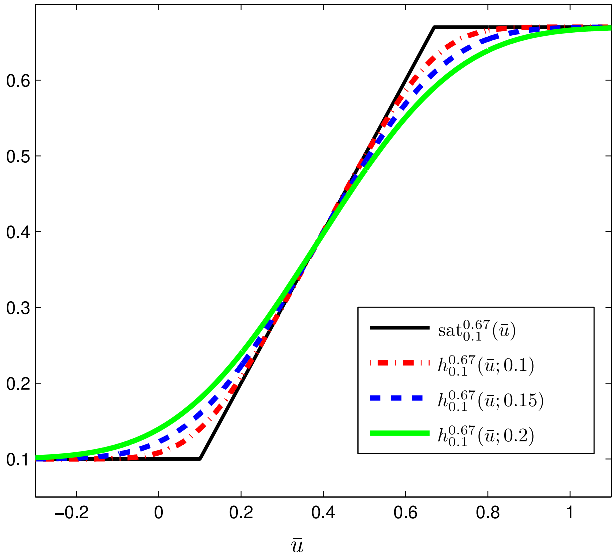


Fig. 7 Comparison between $h_\alpha^\beta(\bar{u}; W_\infty\sigma_n)$ and $\mathrm{sat}_\alpha^\beta(\bar{u})$ for several cases

tends to zero, $h_\alpha^\beta(\bar{u}(k); W_\infty\sigma_n)$ converges to $\mathrm{sat}_\alpha^\beta(u(k))$ as reported in Eun and Hamby (2014).

Let us denote the asymptotic limits of $r(k)$, $d(k)$, $\bar{y}_H(k)$, $\bar{y}_Q(k)$, $\bar{e}(k)$, and $\bar{u}(k)$ are, respectively, denoted by r, d, $\bar{y}_H$, $\bar{y}_Q$, $\bar{e}$, and $\bar{u}$ for $k \to \infty$ (e.g. $\bar{e} = \lim_{k\to\infty} \bar{e}(k)$). In steady state, as it follows from Fig. 6, $\bar{y}_Q$ and $\bar{u}$ satisfies

$$\bar{u} = W(1)\,(r - \bar{y}_H) + Q(1)h_\alpha^\beta(\bar{u}; W_\infty\sigma_n), \tag{8}$$

$$\bar{y}_Q = Q(1)h_\alpha^\beta(\bar{u}; W_\infty\sigma_n), \tag{9}$$

To account for cases where $W(1) = \infty$, we rewrite Eq. (8) as

$$\frac{\bar{u}}{W(1)} = (r - \bar{y}_H) + \frac{Q(1)}{W(1)} h_\alpha^\beta(\bar{u}; W_\infty\sigma_n), \tag{10}$$

We denote the value of noise induced tracking error $r - \bar{y}_H$ by $\bar{e}$. Then, we state the key result on the noise-induced tracking error in Theorem 1.

Theorem 1 *The steady state tracking error of the system of Fig. 6 is given by*

$$\bar{e} = \frac{\bar{u} - Q(1)h_\alpha^\beta(\bar{u}; W_\infty\sigma_n)}{W(1)}. \tag{11}$$

Proof The value of $\bar{u}$ is calculated by solving

$$\frac{\bar{u} - Q(1)h_\alpha^\beta(\bar{u}; W_\infty\sigma_n)}{H(1)W(1)} = \frac{r}{H(1)} - \left(h_\alpha^\beta(\bar{u}; W_\infty\sigma_n) + d\right). \tag{12}$$

if $\sigma_n = 0$, then (11) reduces to

$$\bar{e} = \frac{\bar{u} - Q(1)\bar{u}}{W(1)}. \tag{13}$$

if the plant $H(z)$ has a pole at $z = 1$, for an equilibrium to exist, the input to the plant $H(z)$ must be zero in the steady state, that is,

$$h_{\alpha}^{\beta}(\bar{u};\, W_{\infty}\sigma_n) + d = 0. \tag{14}$$

Therefore, we obtain the steady state tracking error

$$\bar{e} = \frac{\bar{u} + Q(1)d}{W(1)}. \tag{15}$$

□

Theorem 1 shows the steady state tracking error in the systems of Fig. 6. Since the thermal control systems of Fig. 4 can also be expressed as that of Fig. 5, NITE in this thermal control system can be analyzed through Theorem 1 as well. In case of the thermal control system, from (12), $Q(1)$ is equal to 1, and r and $W(1)$ correspond to ΔT_R and $\frac{1}{G(1)}$, respectively. Also, W_{∞} and $\bar{u}$ of $K(z)$ are denoted by κ and $\bar{U}$. Theorem 1 implies that the thermal control system with measurement noise yields CPU temperature tracking error, T_{error}, to the level defined by $\bar{e}$ in (11).

The temperature error in (11) is proportional to the difference between $\bar{U}$ and $h_{\alpha}^{\beta}(\bar{U}, \kappa\sigma_n)$. As one can see in Fig. 7, this difference increases as $\bar{U}$ approaches to the boundary α or β. Also, for given $\bar{U}$, the difference increases as $\kappa\sigma_n$ increases. Therefore, larger $\kappa\sigma_n$ yields higher level of temperature error. The difference is also a function of κ which is given by the sum of the controller gains, $K_p + K$. Thus, larger proportional control gain, or larger integral control gain increases the temperature error as well.

Since the temperature error is the product of the d.c. gain $G(1)$ of the anti-windup controller $G(z)$ and $\bar{U} - h_{\alpha}^{\beta}(\bar{U}, \kappa\sigma_n)$, removing the anti-windup controller eliminates the noise-induced temperature error. Indeed, the noise-induced temperature error is a phenomenon that uniquely takes place when a PI controlled system with anti-windup is operating with measurement noise. Although removing anti-windup eliminates the temperature error, this is not a desirable solution to deal with the effect of measurement noise in thermal control as many cases had been reported regarding adverse effect of controller windup (Goodwin et al. 2000).

In fact, Fig. 8 shows the response of the PI thermal control system without anti-windup when ambient temperature dropped between 1000 and 4000 s due to a sudden operating environment change. From 0 to 1000 s, noise-induced temperature error does not take place as predicted by Theorem 1. However, controller windup phenomenon occurs at 4000 s, which makes the CPU temperature exceed the setpoint for a prolonged period of time. The desired target temperature is not achieved from 1000 to 4000 s although CPU is running tasks at the maximum utilization level, due to lower ambient

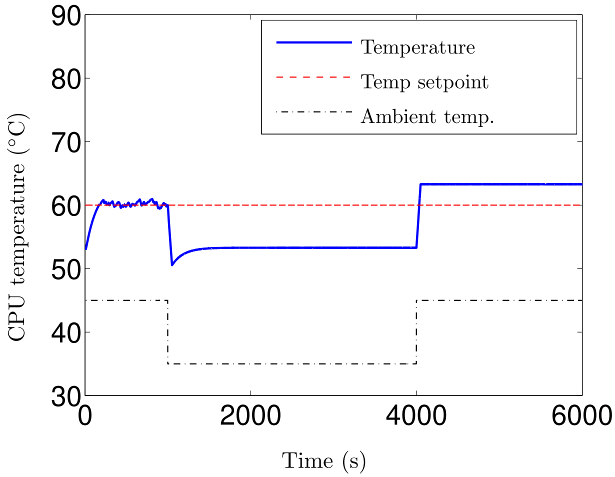


Fig. 8 Response of the PI thermal control system without anti-windup under ambient temperature change

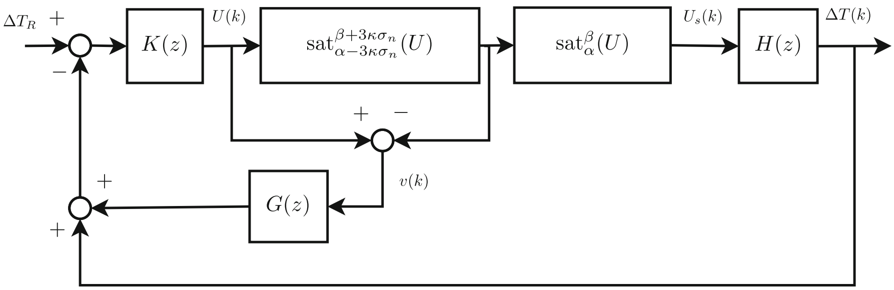


Fig. 9 Thermal control system with virtual saturation

temperature, which is inevitable. As illustrated in this case, removing anti-windup is not a desirable solution. It is necessary to develop a method that suppresses noise-induced error while retaining anti-windup function.

3.4 Design of thermal control with virtual saturation

Now, we design an effective method for eliminating the noise-induced error in feedback thermal control. The mechanism of the noise-induced temperature error is such that the measurement noise is amplified by the controller gain, acts through the system and persistently triggers the anti-windup controller. In order to prevent anti-windup controller activation by the measurement noise, we introduce a virtual saturation block which has a wider linear range than the actually allowed utilization range.

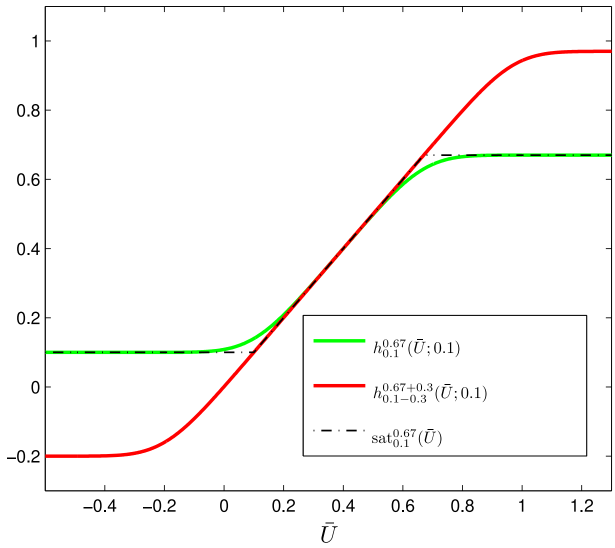


Fig. 10 Comparison of $\mathrm{sat}_{\alpha}^{\beta}(\bar{U})$, $h_{\alpha}^{\beta}(\bar{U};\kappa\sigma_n)$, and $h_{\alpha'}^{\beta'}(\bar{U};\kappa\sigma_n)$ with $\alpha = 0.1$, $\beta = 0.67$, $\kappa\sigma_n = 0.1$, $\alpha' = 0.1 - 0.3$, and $\beta' = 0.67 + 0.3$

The proposed scheme, referred to as Thermal Control under Utilization Bound with Virtual Saturation (TCUB-VS), is illustrated in Fig. 9. In order to make $U(k)$ lie inside the linear region, TCUB-VS introduces a virtual saturation block with $\mathrm{sat}_{\alpha'}^{\beta'}(U(k))$, where $\alpha' = \alpha - 3\kappa\sigma_n$ and $\beta' = \beta + 3\kappa\sigma_n$. The margin of three standard deviations[2] is determined using the Gaussian distribution of the noise process and is expected to prevent anti-windup activation from noise about 99.7% of the time. In this manner, we can eliminate the noise-induced activation of anti-windup while the true windup behavior due to unexpected environment change would still be prevented.

The function $h_{\alpha'}^{\beta'}(\bar{U}(k);\kappa\sigma_n)$ is depicted in Fig. 10. Clearly this gives an enlarged linear region. Thus, the difference between $\bar{U}(k)$ and $h_{\alpha'}^{\beta'}(\bar{U}(k);\kappa\sigma_n)$ is negligible in the original linear region of [0.1, 0.67].

As illustrated in Fig. 10, for the steady-state utilization level $\bar{U}$ that satisfies $\alpha \le \bar{U} \le \beta$, we obtain $h_{\alpha'}^{\beta'}(\bar{U}(k);\kappa\sigma_n) \approx \bar{U}$. Therefore, from (11) in the averaged dynamics, we arrive at $T_{\text{error}} \approx 0$. In this manner, TCUB-VS significantly suppress the noise-induced error while maintaining anti-windup functionality.

It should be noted that a low-pass filter is one of alternative candidates to mitigate the NITE phenomenon. However, a low-pass filter is sensitive in terms of system stability unless carefully designed while the proposed approach, TCUB-VS, can be easily implemented by simply enlarging the saturation bound without any significant loss in stability (Fu et al. 2010).

[2] We have chosen three standard deviations in an arbitrary manner. We can use a different value if needed. However, use of too high a margin value slows down the transient of temperature control. Three standard deviations can eliminate NITE substantially while at the same time causing little degradation of the transient performance of the temperature control.

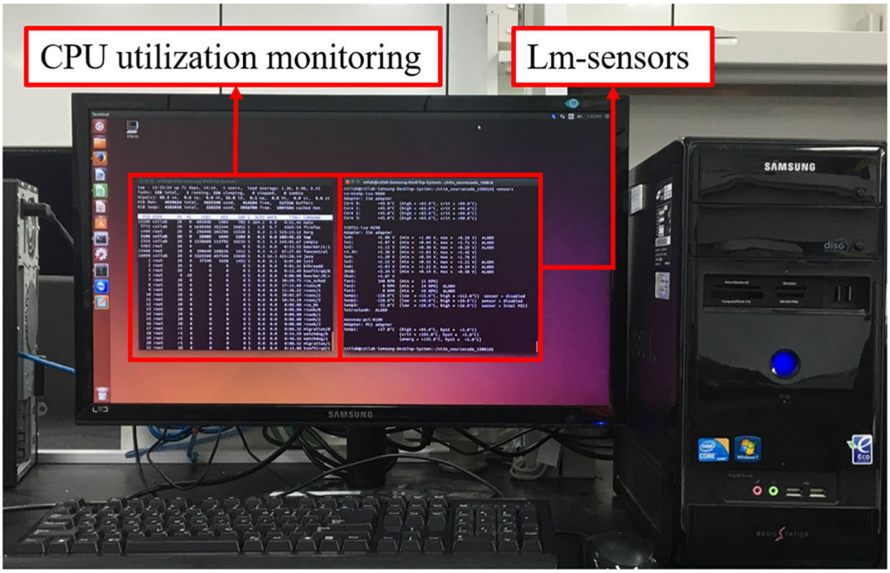


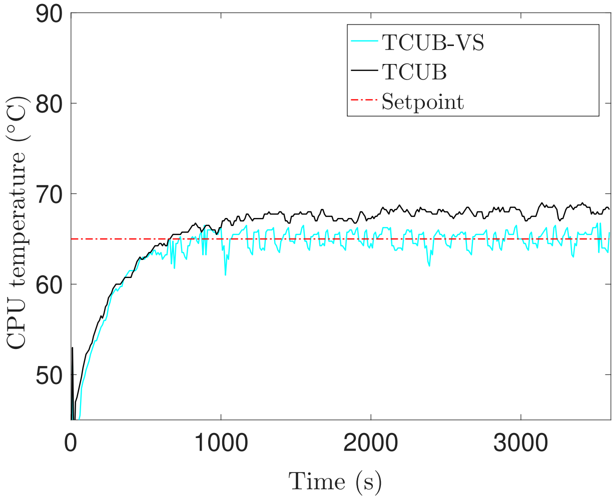
Fig. 11 The hardware platform used in the experiment

Table 2 Thermal related parameters for experiment

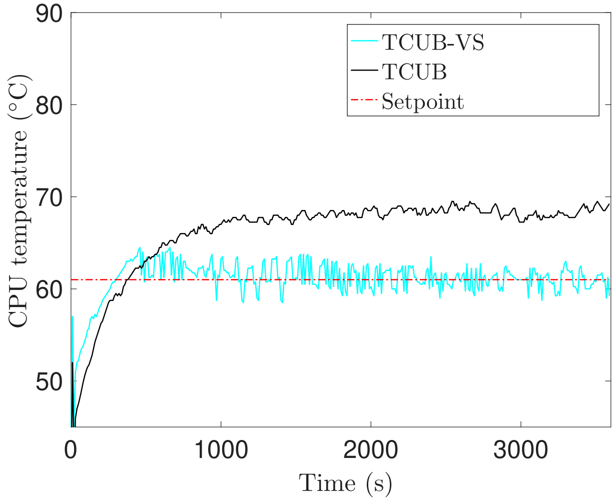
Parameters	Notation	Value
Thermal resistance	R_{th}	0.883K/W
Thermal capacitance	C_{th}	309.02J/K
Control system gains	K_p	0.6066
	K_i	0.001
	Φ	0.964
	Γ	3.0186
Utilization upper bound	β	0.67
Utilization lower bound	α	0.1
Thermal controller sampling time	T_s	10 s

4 Experiments

Here, we evaluate the performance of TCUB-VS through extensive empirical studies by implementing the proposed architecture. The hardware platform used for the experiment is a Samsung DM700T2A-ASC7 desktop with an Intel i5-760 @ 2.793 GHz and the Linux kernel 3.13 with Ubuntu 14.04.2 LTS, as shown in Fig. 11.



(a) Thermal control response with $T_R = 65°C$



(b) Thermal control response with $T_R = 61°C$

Fig. 12 Motivation scenario revisited

TCUB-VS monitors the CPU temperature and allows the CPU to operate with the utilization value calculated by the thermal controller. In order to perform these operations, TCUB-VS consists of the following three components; temperature monitoring, thermal controller, and utilization controller.

Temperature monitoring These days, a processor typically has a digital thermal sensor (DTS) for each core, which can be supported by the operating system or third-party packages. In Linux, lm-sensors which is a free and open-source application, provides

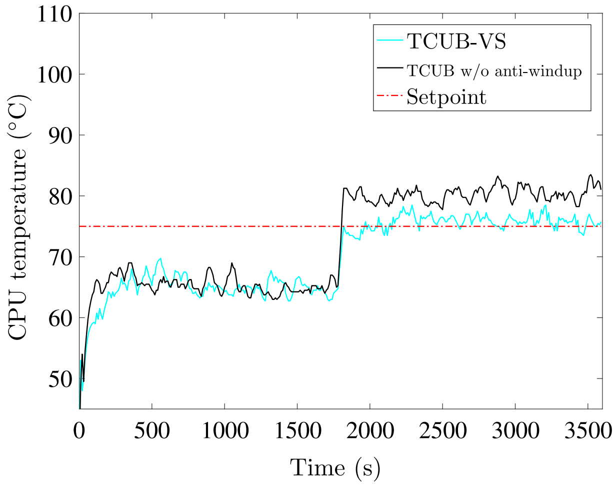


Fig. 13 Controller windup revisited

tools and drivers for monitoring temperatures of CPU cores. We use lm-sensors to get the temperature of CPU cores for our experiment.

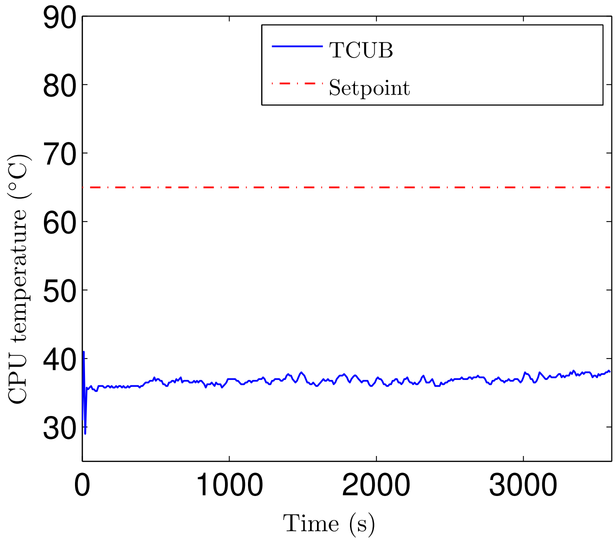
Thermal controller In order to properly design the parameters of TCUB-VS, we obtain the CPU thermal parameters as follows. First, the thermal resistance can be obtained using linear regression from the relationship between CPU utilization and temperature. In addition, thermal capacitance is acquired by using the discrete thermal RC model (Ferreira et al. 2007; Bellosa et al. 2003), CPU power consumption (Intel 2009), and temperature transient graph empirically obtained from our platform. Thermal capacitance and thermal resistance of the CPU, and the parameters of TCUB-VS are summarized in Table 2.

Utilization controller To change CPU utilization, we generate workload by using CRC benchmark. For each core, we run a thread that controls CPU utilization by regulating the work time and the sleep time. We use the system tick counter for the work time. We implement the sleep time using *usleep* function.

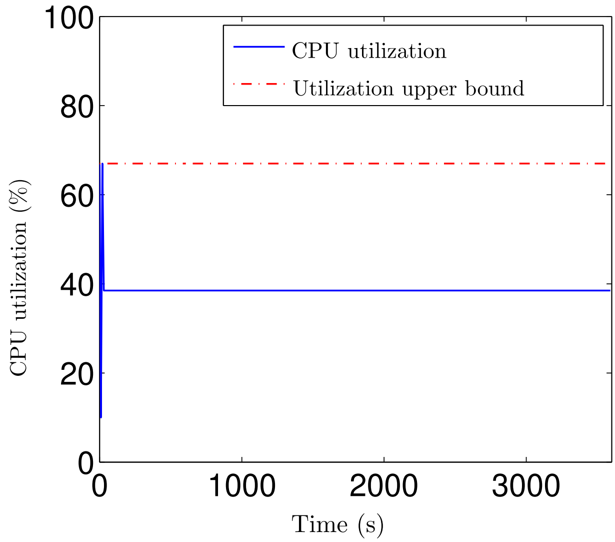
Here, we study five scenarios. The first two revisit the problems in Figs. 1 and 8, respectively. Then, we discuss the CPU under-utilization problem and execution time variation of CPU. Finally, we show that the noise-induced temperature error exists with a different hardware platform. In all of the scenarios, we confirm that the proposed TCUB-VS can effectively eliminate the error.

4.1 Experiment I: motivating scenario revisited

In Fig. 1, we show that TCUB fails to achieve the desired CPU temperature due to sensor measurement noise even when the intuitive attempt to eliminate the temperature error lowers the target temperature. Figure 12 shows the temperature responses of TCUB-VS and TCUB under the same experimental condition. In particular, Fig. 12a shows the case of $T_R = 65\,°C$, where TCUB-VS can achieve the desired CPU temper-



(a) Thermal control response with $T_R = 65°C$



(b) Utilization outputs

Fig. 14 Under-utilization problem of TCUB

ature while TCUB shows noise-induced steady state error. In the case of $T_R = 61$ °C, Fig. 12b shows that TCUB-VS eliminates the noise-induced error as expected while the steady-state error of TCUB becomes even larger than that in Fig. 12a. These results indicate that TCUB-VS can eliminate the noise-induced steady state error regardless of the target temperature.

4.2 Experiment II: windup scenario revisited

As shown in Fig. 8, we can eliminate noise-induced error by removing the anti-windup function, while the controller suffers from the windup phenomenon. Figure 13 shows the responses of TCUB-VS and TCUB without anti-windup. In the scenario, the initial target temperature is 75 °C, and the ambient temperature abruptly increases due to the change of the operating environment, e.g., from indoor to outdoor. The two temperature responses in Fig. 13 are quite similar until 1800 s. In this interval, the temperatures of both cases are lower than the target value due to low ambient temperature. After 1800 s, the ambient temperature increases by 15 °C that is sufficient to achieve the target temperature. Then, TCUB without anti-windup suffers from controller windup, which results in failure of temperature control. The proposed TCUB-VS can prevent windup phenomenon and tightly controls the temperature around the target value.

4.3 Experiment III: under-utilization problem

In the experiment, we compare TCUB-VS and TCUB under the situation when the target CPU temperature cannot be achieved even if the CPU utilization reaches its maximum-possible value. Here, the CPU fan speed is set to 2935 RPM which is higher than that in other experimental scenarios. The higher fan speed reduces the thermal resistance that leads to lower the CPU temperature with the same CPU utilization.

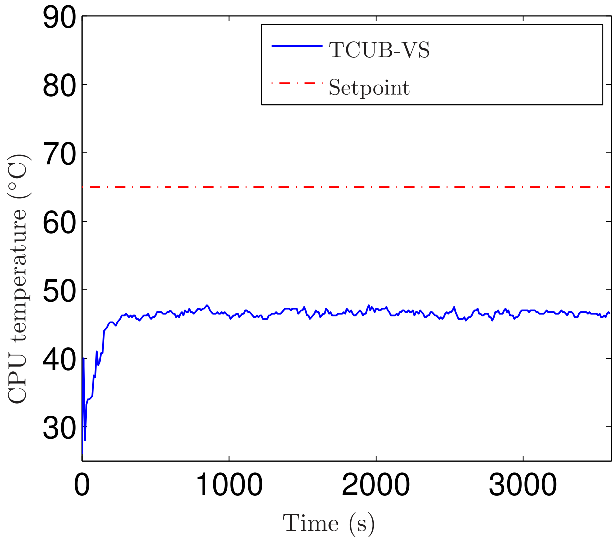
Figure 14 shows that TCUB shows under-utilization. Both TCUB in Fig. 14a and TCUB-VS in Fig. 15a do not achieve the target temperature due to low thermal resistance. Nevertheless, TCUB achieves only 40% of CPU utilization while the upper bound is 66%, which is a significant under-utilization of CPU resources. This under-utilization also results from the NITE phenomenon.

When the operating point of utilization is lower than the midpoint of the range, the NITE results in CPU overheating as shown in the motivating example in Sect. 3.1. In the meantime, when the operating point is higher than the midpoint, the NITE decreases the CPU temperature, which results in under-utilization. TCUB-VS can resolve this under-utilization as shown in Fig. 15b, where the CPU utilization reaches its upper bound.

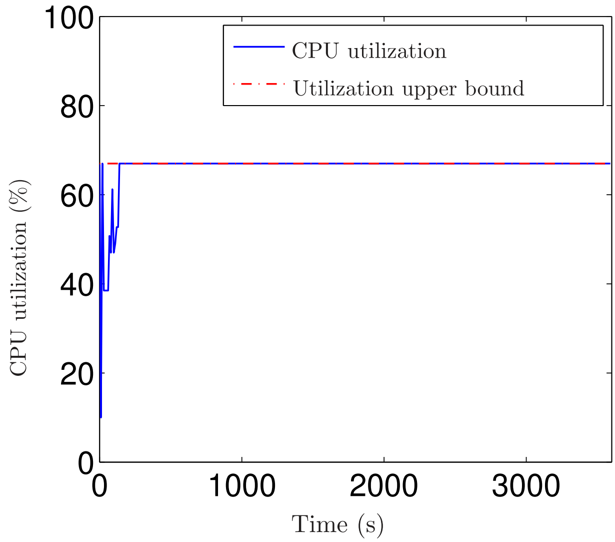
4.4 Experiment IV: execution time variation

In this scenario, we compare TCUB-VS and TCUB when there exists uncertainty in task execution times. The actual task execution time can be different from the estimated time due to various system uncertainties. We introduce the execution time factor (*etf*), which is the ratio between the actual and the estimated task execution times.

Figure 16 shows the thermal control response with $etf = 2$ and $etf = 0.5$, respectively. When $etf = 2$, the actual task execution time is twice the estimated time, and $etf = 0.5$, in contrast, means that the actual task execution time is half of the estimated time. In both cases, TCUB-VS can tightly control the CPU temperature while TCUB shows



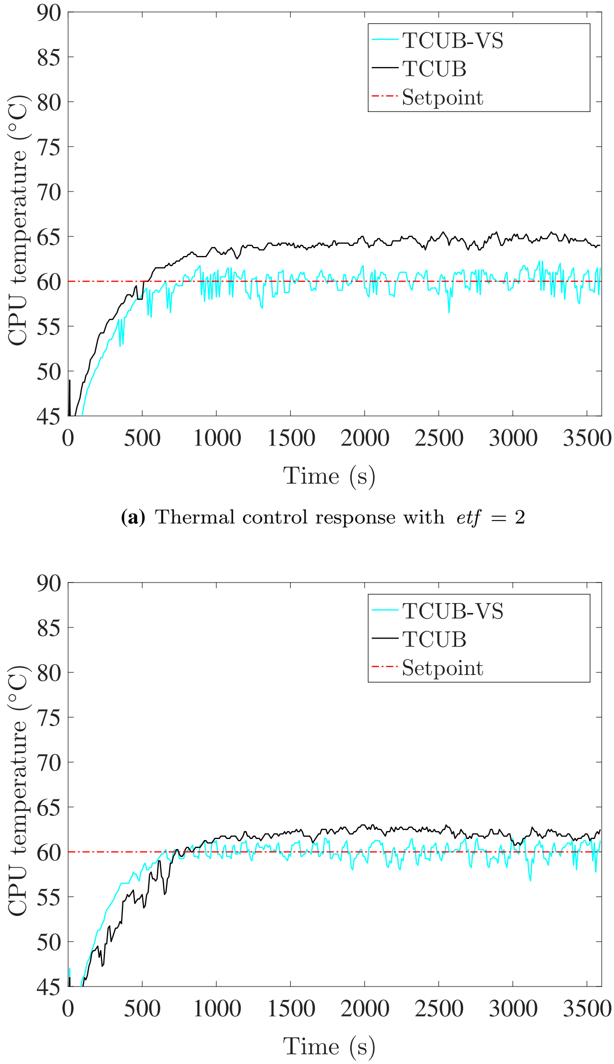
(a) Thermal control response with $T_R = 65°\mathrm{C}$



(b) Utilization outputs

Fig. 15 TCUB-VS for solving under-utilization problem

CPU overheating. It should be noted that TCUB can properly control the temperature under execution time variation when there is no sensor measurement noise, as reported in Fu et al. (2010).



(a) Thermal control response with $etf = 2$

(b) Thermal control response with $etf = 0.5$

Fig. 16 Two responses with execution time variation

4.5 Experiment V: different hardware platform

In order to evaluate performance with a different hardware platform, we carry out the experiment with a different laptop (Toshiba PORTEGE R830). Here, we set the target temperature of 60 °C. As shown in Fig. 17, TCUB-VS does not show any noise-induced temperature error while the response of TCUB exhibits severe bias. As expected from our analysis, noise-induced temperature error still appears with a different hardware platform, and TCUB-VS can effectively eliminate the error.

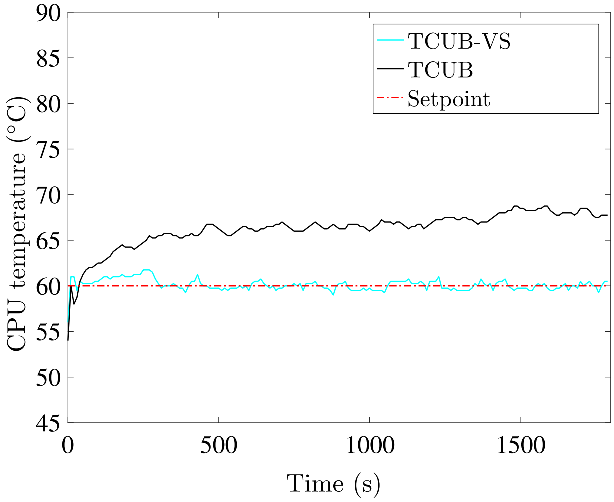


Fig. 17 Thermal control responses with a different processor

5 Conclusion

We have studied the important practical issue of sensor noise in processor thermal control for real-time systems. In particular, we have discovered that a small zero-mean sensor noise can induce a significant steady-state error in processor thermal control. First, we have extended the previous preliminary analysis by considering a general system architecture. Then, based on the general model, we have quantified the error in a closed-form expression in terms of noise statistics and system parameters. Then, we have proposed an efficient thermal control algorithm, TCUB-VS, which can eliminate the noise-induced error. In order to evaluate the performance of the proposed algorithm, we have conducted extensive experiments with our testbed. Our empirical results have confirmed that the proposed scheme can remove the noise-induced error and maintain the desired processor temperature for real-time performance.

Acknowledgements This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (2014-0-00065, Resilient Cyber-Physical Systems Research), the DGIST R&D Program of the Ministry of Science and ICT (18-EE-01), and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2016R1C1B2007899).

References

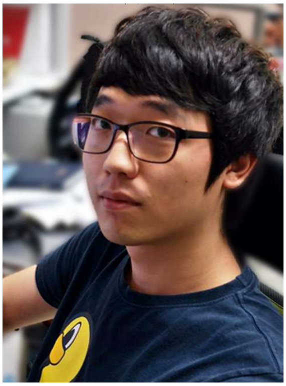
Ahmed R, Ramanathan P, Saluja KK (2016) Necessary and sufficient conditions for thermal schedulability of periodic real-time tasks under fluid scheduling model. ACM Trans Embed Comput Syst (TECS) 15(3):49

Avirneni NDP, Ramesh PK, Somani AK (2016) Utilization aware power management in reliable and aggressive chip multi processors. IEEE Trans Comput 65(3):979–991

Bai EW, Fu LC, Sastry SS (1988) Averaging analysis for discrete time and sampled data adaptive systems. IEEE Trans Circuits Syst 35(2):137–148

Becker M, Sandström K, Behnam M, Nolte T (2014) Dynamic power management for thermal control of many-core real-time systems. In: Proc. workshop on adaptive and reconfigurable embedded systems (APRES)
Bellosa F, Weissel A, Waitz M, Kellner S (2003) Event-driven energy accounting for dynamic thermal management. In: Proc. workshop on compilers and operating systems for low power (COLP)
Beneventi F, Bartolini A, Tilli A, Benini L (2014) An effective gray-box identification procedure for multicore thermal modeling. IEEE Trans Comput 63(5):1097–1110
Chantem T, Hu XS, Dick RP (2011) Temperature-aware scheduling and assignment for hard real-time applications on mpsocs. IEEE Trans Very Large Scale Integr VLSI Syst 19(10):1884–1897
Chu HH, Tsai TH, Chen YS (2012) Thermal-aware service rate adjustment for cyber-physical systems. In: Proc. international conference on anti-counterfeiting, security and identification (ASID)
Cui Y, Zhang W, He B (2017) A variation-aware adaptive fuzzy control system for thermal management of microprocessors. IEEE Trans Very Large Scale Integr VLSI Syst 25(2):683–695
Eun Y, Hamby ES (2014) Noise induced loss of tracking in systems with saturating actuators and anti-windup. ASME J Dyn Syst Meas Control 136(5):p054501 (6 pages)
Ferreira AP, Mosse D, Oh JC (2007) Thermal faults modeling using a rc model with an application to web farms. In: Proc. Euromicro conference on real-time systems (ECRTS)
Floyd MS, Ghiasi S, Keller TW, Rajamani K, Rawson F, Rubio JC, Ware MS (2007) System power management support in the ibm power6 microprocessor. IBM J Res Dev 51(6):733–746
Fu Y, Kottenstette N, Chen Y, Lu C, Koutsoukos XD, Wang H (2010) Feedback thermal control for real-time systems. In: Proc. real-time and embedded technology and applications symposium (RTAS)
Fu Y, Kottenstette N, Lu C, Koutsoukos XD (2012) Feedback thermal control of real-time systems on multicore processors. In: Proc. international conference on embedded software (EMSOFT)
Goodwin GC, Graebe SF, Salgado ME (2000) Control system design. Prentice Hall, Englewood Cliffs
Hettiarachchi PM, Fisher N, Ahmed M, Wang LY, Wang S, Shi W (2012) The design and analysis of thermal-resilient hard-real-time systems. In: Proc. real-time and embedded technology and applications symposium (RTAS)
Intel (2009) i7-800 and i5-700 desktop processor series and lga1156 socket: thermal/mechanical specifications and design guidelines
Khairnasov S, Naumova A (2015) Heat pipes application in electronics thermal control systems. Front Heat Pipes (FHP)
Kim D, Park KJ, Eun Y, Son SH, Lu C (2015) When thermal control meets sensor noise: analysis of noise-induced temperature error. In: Proc. real-time and embedded technology and applications symposium (RTAS), pp 98–107
Kornaros G, Pnevmatikatos D (2013) A survey and taxonomy of on-chip monitoring of multicore systems-on-chip. ACM Trans Des Autom Electron Syst (TODAES) 18(2):17:1–17:38
Kottenstette N, Hall JF III, Koutsoukos X, Antsaklis P (2011) Digital control of multiple discrete passive plants over networks. Int J Syst Control Commun 3(2):194–228
Krishna C, Koren I (2017) Thermal-aware management techniques for cyber-physical systems. Sustain Comput 15:39–51
Long J, Memik SO, Memik G, Mukherjee R (2008) Thermal monitoring mechanisms for chip multiprocessors. ACM Trans Archit Code Optim (TACO) 5(2):9
Mohaqeqi M, Kargahi M, Fouladi K (2016) Stochastic thermal control of a multicore real-time system. In: Proc. 24th Euromicro international conference on parallel, distributed, and network-based processing (PDP), pp 208–215
Pathan RM (2017) Real-time scheduling algorithm for safety-critical systems on faulty multicore environments. Real-Time Syst 53(1):45–81
Prakash A, Amrouch H, Shafique M, Mitra T, Henkel J (2016) Improving mobile gaming performance through cooperative cpu-gpu thermal management. In: Proceedings of the 53rd annual design automation conference
Rotem E, Hermerding J, Cohen A, Cain H (2006) Temperature measurement in the intel® coretm duo processor. In: Proc. international workshop on thermal investigations of ICs and systems (THERMINIC)
Saen M, Osada K, Misaka S, Yamada T, Tsujimoto Y, Kondoh Y, Kamei T, Yoshida Y, Nagahama E, Nitta Y, et al (2007) Embedded soc resource manager to control temperature and data bandwidth. In: Proc. international solid-state circuits conference (ISSCC)

Segovia VR, Kralmark M, Lindberg M, Årzén KE (2011) Processor thermal control using adaptive bandwidth resource management. In: Proc. World Congress of the international federation of automatic control (IFAC)
Sheikh HF, Ahmad I, Wang Z, Ranka S (2012) An overview and classification of thermal-aware scheduling techniques for multi-core processing systems. Sustain Comput 2(3):151–169
Tsai TH, Chen YS, He XX, Li CY (2017) Stem: a thermal-constrained real-time scheduling for 3d heterogeneous-isa multicore processors. IEEE Trans Comput
Yue J, Zhang T, Liu Y, Quan B, Tianzhou C (2012) Thermal-aware feedback control scheduling for soft real-time systems. In: Proc. international conference on high performance computing and communications & international conference on embedded software and systems (HPCC-ICESS)
Yun B, Shin KG, Wang S (2013) Predicting thermal behavior for temperature management in time-critical multicore systems. In: Proc. real-time and embedded technology and applications symposium (RTAS)
Zhou J, Wei T (2015) Stochastic thermal-aware real-time task scheduling with considerations of soft errors. J Syst Softw 102:123–133
Zhou J, Wei T, Chen M, Yan J, Hu XS, Ma Y (2016) Thermal-aware task scheduling for energy minimization in heterogeneous real-time mpsoc systems. IEEE Trans Comput Aided Des Integr Circuits Syst 35(8):1269–1282



Dohwan Kim received the B.S. degree in computer science and engineering from Kyungpook National University, South Korea, in 2013. He is currently pursuing the Ph.D. degree with the Cyber-Physical Systems Integration Laboratory, Daegu Gyeongbuk Institute of Science and Technology (DGIST). His research interests include resilient cyber-physical system, communication network, and real-time systems.

Juseung Lee received the B.S.E. degrees in electrical engineering from Chungbuk National University, Korea, in 2013, and M.S. degrees in information and communication engineering from Daegu Gyeongbuk Institute of Science and Technology (DGIST), Korea, in 2015. Currently, he is a Ph.D. candidate in the Information and Communication Engineering at DGIST. His research interests include quasi-linear control systems, control systems with nonlinear sensors and actuators, and resilient cyber-physical systems.

Kyung-Joon Park (M'05) received the B.S. and M.S. degrees in electrical engineering from the School of Electrical Engineering, and the Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 1998, 2000, and 2005, respectively. From 2005 to 2006, he was a Senior Engineer with the Samsung Electronics, Suwon, Korea. From 2006 to 2010, he was a Postdoctoral Research Associate with the Department of Computer Science, University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA. He is currently a Professor with the Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu, Korea. His research interests include resilient cyber-physical systems and smart production systems.

Yongsoon Eun (M'03) received the B.A. degree in mathematics, and the B.S. and M.S.E. degrees in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1992, 1994, and 1997, respectively, and the Ph.D. degree in electrical engineering and computer science from the University of Michigan, Ann Arbor, MI, USA, in 2003. From 2003 to 2012, he was a Research Scientist with the Xerox Innovation Group, Webster, NY, USA, where he worked on a number of subsystem technologies in the xerographic marking process and image registration method in production inkjet printers. Since 2012, he is a Professor with the Department of Information and Communication Engineering, DGIST, Korea. His research interests include control systems with nonlinear sensors and actuators, geometric control of quadrotors, communication network, production systems, and resilient cyber-physical systems.

Sang Hyuk Son (M'85-SM'98-F'13) received the B.S. degree in electronics engineering from Seoul National University, the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), and the Ph.D. degree in computer science from the University of Maryland, College Park, MD, USA. He has been a Professor with the Department of Computer Science, University of Virginia, and WCU Chair Professor with Sogang University. He has been a Visiting Professor at KAIST, the City University of Hong Kong, Ecole Centrale de Lille in France, Linkoping University, and the University of Skövde, Sweden. He is currently the President of the Daegu Gyeongbuk Institute of Science and Technology. His research has been funded by the Korean Government, National Research Foundation, National Science Foundation, DARPA, Office of Naval Research, Department of Energy, National Security Agency, and IBM. He has authored or co-authored over 340 papers and edited/authored four books in his research areas. His research interests include cyber physical systems, realtime and embedded systems, database and data services, and wireless sensor networks. Prof. Son is a member of both the Korean Academy of Science & Technology and the National Academy of Engineering of Korea. He is a Founding Member of the ACM/IEEE CPSWeek, and serving as a member of the steering committee for the IEEE RTCSA and Cyber Physical Systems Week. He received the Outstanding Contribution Award from the Cyber Physical Systems Week in 2012. He has served on the Editorial Board of the ACM Transactions on Cyber Physical Systems, the IEEE Transactions on Computers, the IEEE Transactions on Parallel and Distributed Systems, and Real-Time Systems Journal.

Chenyang Lu (F'15) received the B.S. degree from the University of Science and Technology of China in 1995, the M.S. degree from the Chinese Academy of Sciences in 1997, and the Ph.D. degree from University of Virginia in 2001. He is the Fullgraf Professor with the Department of Computer Science and Engineering, Washington University in St. Louis. He has authored and co-authored over 150 research papers with over 16,000 citations with an H-index of 56. His research interests include real-time systems, wireless sensor networks, cyber-physical systems, and Internet of Things. He is the Editor-in-Chief of the ACM Transactions on Sensor Networks, an Area Editor of the IEEE Internet of Things Journal, and an Associate Editor of the ACM Transactions on Cyber-Physical Systems and Real-Time Systems Journal. He also chaired premier conferences, such as IEEE Real-Time Systems Symposium, ACM/IEEE International Conference on Cyber-Physical Systems, and ACM Conference on Embedded Networked Sensor Systems.