When Thermal Control Meets Sensor Noise: Analysis of Noise-induced Temperature Error

Dohwan Kim*, Kyung-Joon Park*§, Yongsoon Eun*§, Sang H. Son*, and Chenyang Lu†
* Dept. of Information & Communication Engineering, DGIST, Daegu, Korea
{hoihoi6914, kjp, yeun, son}@dgist.ac.kr
† Dept. of Computer Science & Engineering, Washington University, St. Louis, MO, USA
lu@cse.wustl.edu

***Abstract*—Thermal control is critical for real-time systems as overheated processors can result in serious performance degradation or even system breakdown due to hardware throttling. The major challenges in thermal control for real-time systems are (i) the need to enforce both real-time and thermal constraints; (ii) uncertain system dynamics; and (iii) thermal sensor noise. Previous studies have resolved the first two, but the practical issue of sensor noise has not been properly addressed yet. In this paper, we introduce a novel thermal control algorithm that can appropriately handle thermal sensor noise. Our key observation is that even a small zero-mean sensor noise can induce a significant steady-state error between the target and the actual temperature of a processor. This steady-state error is contrary to our intuition that zero-mean sensor noise induces zero-mean fluctuations. We show that an intuitive attempt to resolve this unusual situation is not effective at all. By a rigorous approach, we analyze the underlying mechanism and quantify the noised-induced error in a closed form in terms of noise statistics and system parameters. Based on our analysis, we propose a simple and effective solution for eliminating the error and maintaining the desired processor temperature. Through extensive simulations, we show the advantages of our proposed algorithm, referred to as Thermal Control under Utilization Bound with Virtual Saturation (TCUB-VS).**

I. Introduction

Thermal control is crucial for many real-time systems because overheated processors can cause significant performance degradation or even system failure due to hardware throttling. The main challenges in thermal control for real-time systems are *(i)* the need to enforce both real-time and thermal constraints; *(ii)* uncertain system dynamics; *(iii)* thermal sensor noise.

Traditional approaches to thermal-aware real-time scheduling deals with *(i)*, but typically requires accurate system models. Feedback-based thermal control schemes have been further introduced to deal with uncertain system dynamics, which resolves both *(i)* and *(ii)*. However, existing feedback thermal control work does not explicitly address the thermal sensor noise, which is substantial and common on microprocessors in practice [1], [2]. In fact, 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 $10°\mathrm{C}$ [1], [2]. Consequently, we need a thermal control scheme that can appropriately deal with *not only (i)* and *(ii)*, *but also* the practical issue of *(iii)*.

In this paper, we introduce a novel thermal control algorithm that can handle sensor noise in a proper manner. Our rigorous theoretical analysis and extensive simulation results show that the proposed approach shows promising performance under various scenarios with sensor noise. In particular, our main contributions are as follows:

- We discover that even a small zero-mean sensor noise can induce a significant steady-state temperature error in thermal control for real-time systems.
- Based on the stochastic averaging theory, we rigorously analyze the noise-induced error and quantify it in a closed form in terms of noise statistics and system parameters.
- We propose a novel solution, referred to as Thermal Control under Utilization Bound with Virtual Saturation (TCUB-VS), which can eliminate the noise-induced error in thermal control.

More specifically, our key observation is that a small zero-mean sensor noise can result in a significant steady-state error in processor temperature control, which has never been studied in any related work on feedback thermal control. It should be noted that this steady-state error is completely different from typical zero-mean temperature fluctuations with measurement noise.

A non-zero steady-state error in thermal control will result in either system under-utilization, or much more seriously, processor overheating and even system breakdown: 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 despite feedback thermal control. The latter is much more critical than the former.

Furthermore, we discover that an intuitive attempt for resolving this problem without deep understanding fails to reduce the temperature error. Consequently, when feedback thermal control meets sensor noise, it is necessary to understand and analyze, in a rigorous manner, this unusual phenomenon of a non-zero temperature error. We present a rigorous analysis of the problem and propose a remedy.

The remainder of this paper is organized as follows: In Section II, as our motivation, we illustrate the effect of sensor noise in feedback thermal control. First, we show that a small

§K.-J. Park and Y. Eun are the co-corresponding authors.

978-1-4799-8603-3/15/$31.00 ©2015 IEEE

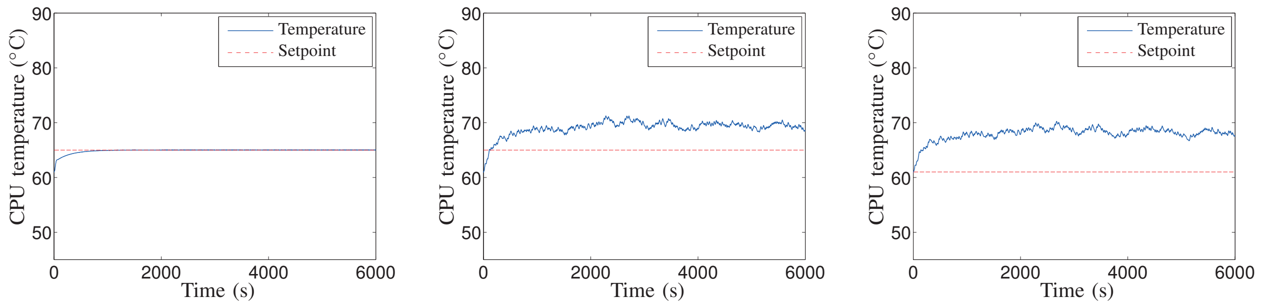


Fig. 1. Noise-induced error in feedback thermal control.

sensor noise can induce a significant error in processor temperature. Then, we demonstrate that an intuitive attempt to resolve the situation is not effective at all. We describe the system models in Section III. Section IV gives a rigorous analysis of the noise-induced error based on the stochastic averaging theory. Then, we quantify the error as a closed-form function of noise statistics and system parameters. In Section V, we introduce our solution, TCUB-VS, which can remove the noise-induced error. We evaluate the performance of TCUB-VS in Section VI. We give related work in Section VII. Our conclusion follows in Section VIII.

II. 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 [3] where utilization level is controlled based on CPU temperature feedback in order to prevent processor overheating. This method is referred to as TCUB [3]. Figure 1(a) 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 1(b) shows the control performance when zero-mean Gaussian sensor noise 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 [3]. 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 Section IV.

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 $65°C$ to $61°C$, in an attempt to lower the CPU temperature. The result of such an attempt is shown in Fig. 1(c). 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.

III. System Model and Description

A. CPU Thermal Dynamics

We adopt the system models in [3], which is introduced here for completeness. For the thermal dynamics of the processor, the well-known thermal RC model is used as follows [4], [5]:

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

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.

B. Overview of TCUB

The primary design objectives for feedback thermal control are two folds: *(i)* to prevent processor overheating, and *(ii)* to maintain desired real-time system performance. In 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 [3].

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 k-th sampling period of the outer loop, based on the measured

[1] Noise level with variance of 1 is a mild condition in practice [1], [2]. Details of the simulation environment can be found in Section VI.

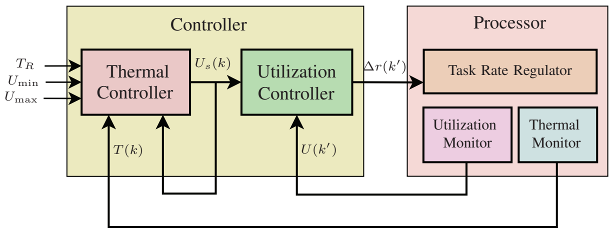


Fig. 2. Overall structure of TCUB.

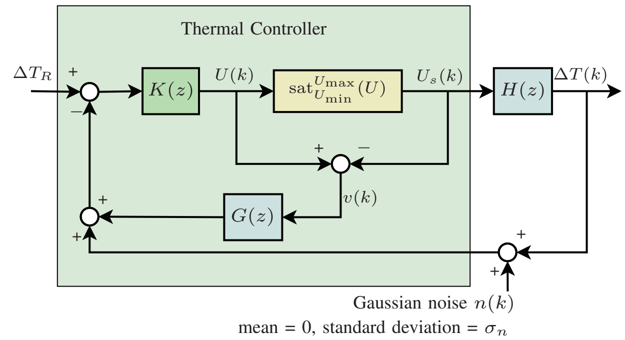


Fig. 3. Detailed view of the thermal control system with measurement noise.

temperature $T(k)$ by the thermal sensor, 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 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, $U_{\min}$, and $U_{\max}$, respectively.

C. Thermal Control System Model

Figure 3 shows the block diagram representation of TCUB feedback thermal control system [3], 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 the CPU temperature, respectively. The upper and lower bounds of the CPU utilization in TCUB is denoted by the saturation block $\text{sat}_\alpha^\beta(u)$ with the following definition:

$$\text{sat}_\alpha^\beta(u) = \begin{cases} \beta, & u > \beta \\ u, & \alpha \le u \le \beta \\ \alpha, & u < \alpha. \end{cases}$$

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 $U_{\max}$ is determined by the schedulability region within which it is feasible to meet the deadlines of the real-time tasks considered. The lower bound $U_{\min}$ 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 k-th sampling period. Then, from [3], we use $T_0 + T_{\text{idle}}$ as a reference point to define $\Delta T(k) = T(k) - (T_0 + T_{\text{idle}})$, where T_0 is the ambient temperature and T_{idle} is the idle temperature determined by $T_{\text{idle}} = R_{\text{th}} P_{\text{idle}}$ 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

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

where $\Phi = \exp(-T_s/(R_{\text{th}} C_{\text{th}}))$ with T_s being the sampling interval of the control and $\Gamma = (G_p P_a - P_{\text{idle}}) R_{\text{th}} (1 - \Phi)$ 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

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

where $K = K_i T_s (1 + \frac{w_I T_s}{2})$ and $b = \frac{2 - w_I T_s}{2 + w_I T_s}$ [3], [6]. 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 $H(z)$.

IV. Analysis of a Thermal Controller with Sensor Noise

In this section, we analyze the effect of sensor noise on feedback thermal control and provide explanation of the phenomenon presented in Section II.

It turns out that control performance degradation in the form of bias due to zero-mean sensor noise has recently been discovered and analyzed in automatic control systems community [7]. This phenomenon is referred to as Noise Induced Tracking Error (NITE). However, the analysis in [7] is limited to the systems described in continuous time. Here, we extend the work of [7] to the feedback thermal control system in discrete time.

Figure 3 shows the thermal control system with measurement noise process n. We assume $n(k)$ is a zero-mean wide-sense-stationary white Gaussian process with standard deviation of σ_n.[2] The system in Fig. 3 can be written in the state-space form as follows:

$$\Delta T(k+1) = \Phi \Delta T(k) + \Gamma \text{sat}_\alpha^\beta(U(k)), \quad (1)$$
$$y(k) = \Delta T(k) + n(k), \quad (2)$$

where $\alpha = U_{\min}$, $\beta = U_{\max}$, and $y(k)$ denotes the temperature sensor output with measurement noise. In a similar

[2]Our result can be easily extended to any non-Gaussian random noise.

manner, the dynamics for $G(z)$ can be written as

$$x(k+1) = \Phi x(k) + \Gamma \left[U(k) - \text{sat}_\alpha^\beta(U(k))\right], \qquad (3)$$

where $x(k)$ denotes the state of the anti-windup controller. In order to include the case when $G(z) \neq H(z)$ in the analysis we modify (3) to a more general form of

$$x(k+1) = \Phi_{\text{AW}} + \Gamma_{\text{AW}} \left(U(k) - \text{sat}_\alpha^\beta(U(k))\right), \qquad (4)$$

where Φ_{AW} and Γ_{AW} are parameters of anti-windup not necessarily same as plant parameters Φ and Γ. In the subsequent analysis we use (4) as the dynamics of anti-windup instead of (3).

The PI controller dynamics is given as

$$w(k+1) = w(k) + K(1-b)(\Delta T_R - y(k) - x(k)) \qquad (5)$$

$$U(k) = w(k) + (K_p + K)(\Delta T_R - y(k) - x(k)), \qquad (6)$$

where $w(k)$ is the state of the PI controller.

According to the averaging theory in discrete time [8], for sufficiently small sampling time T_s, the solution $\Delta T(k)$ of the system of equations (1)–(6) is approximated by the solution $\Delta\bar{T}(k)$ of the following system:

$$\Delta\bar{T}(k+1) = \Phi\Delta\bar{T}(k) + \Gamma h_\alpha^\beta(\bar{U}(k); \kappa\sigma_n), \qquad (7)$$

$$\bar{x}(k+1) = \Phi_{\text{AW}}\bar{x}(k) + \Gamma_{\text{AW}}\left[\bar{U}(k) - h_\alpha^\beta(\bar{U}(k); \kappa\sigma_n)\right], \qquad (8)$$

$$\bar{w}(k+1) = \bar{w}(k) + K(1-b)(\Delta T_R - \Delta\bar{T}(k) - \bar{x}(k)) \qquad (9)$$

$$\bar{U}(k) = \bar{w}(k) + \kappa(\Delta T_R - \Delta\bar{T}(k) - \bar{x}(k)), \qquad (10)$$

where $\kappa = K_p + K$ and the function $h_\alpha^\beta(u; \kappa\sigma_n)$ is defined as

$$h_\alpha^\beta(u; \kappa\sigma_n) = \frac{\alpha+\beta}{2} + \frac{\kappa\sigma_n}{\sqrt{2\pi}}\left(e^{-\frac{(u-\alpha)^2}{2\kappa^2\sigma_n{}^2}} - e^{-\frac{(u-\beta)^2}{2\kappa^2\sigma_n{}^2}}\right) + \frac{u-\alpha}{2}\text{erf}\left(\frac{u-\alpha}{\sqrt{2}\kappa\sigma_n}\right) - \frac{u-\beta}{2}\text{erf}\left(\frac{u-\beta}{\sqrt{2}\kappa\sigma_n}\right) \qquad (11)$$

and $\text{erf}(\xi)$ is the error function defined as $\text{erf}(\xi) = \frac{2}{\sqrt{\pi}}\int_0^\xi e^{-t^2}dt$ [7].

Equations (1)–(6) represent a dynamic system driven by random process $n(k)$ whereas equations (7)–(10) represent a dynamic system that is deterministic for given initial conditions. Thus, by applying the stochastic averaging method, the behavior of the dynamic system driven by random noise process can be studied by the behavior of the resulting deterministic system.

The effect of averaging is captured in the function $h_\alpha^\beta(\bar{U}(k); \kappa\sigma_n)$ in (7) and (8) that replaces the actuator saturation $\text{sat}_\alpha^\beta(U(k))$ in (1) and (3). The two functions are compared in Fig. 4 with the parameters $\alpha = 0.1$ and $\beta = 0.67$ for several values of $\kappa\sigma_n$. 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 $\kappa\sigma_n$ increases. When σ_n tends to zero, $h_\alpha^\beta(\bar{U}(k); \kappa\sigma_n)$ converges to $\text{sat}_\alpha^\beta(U(k))$ as reported in [7].

Let us denote the asymptotic limits of $\Delta\bar{T}(k)$, $\bar{x}(k)$, $\bar{w}(k)$,

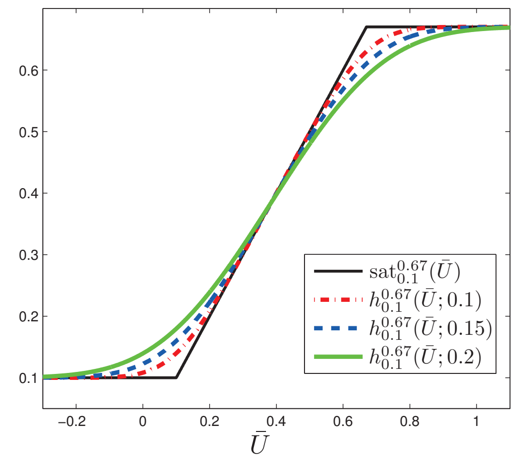


Fig. 4. Comparison between $h_\alpha^\beta(\bar{U}; \kappa\sigma_n)$ and $\text{sat}_\alpha^\beta(\bar{U})$ for several cases.

and $\bar{U}(k)$ for $k \to \infty$ by $\Delta\bar{T}$, $\bar{x}$, $\bar{w}$, and $\bar{U}$, respectively. We also denote the value of thermal control tracking error $\Delta T_R - \Delta\bar{T}$ by T_{error}. Then, we state the key result on the noise-induced thermal control error in Theorem 1.

Theorem 1. *The steady-state temperature tracking error of the system (7)–(10) is given by*

$$T_{\text{error}} = G_0\left(\bar{U} - h_\alpha^\beta(\bar{U}; \kappa\sigma_n)\right), \qquad (12)$$

where G_0 is the DC gain of the anti-windup controller and given by $\Gamma_{\text{AW}}/(1 - \Phi_{\text{AW}})$. The value of $\bar{U}$ is calculated by solving

$$\bar{U} - \frac{1-\Phi_{\text{AW}}}{\Gamma_{\text{AW}}}\Delta T_R = \left(1 - \frac{1-\Phi_{\text{AW}}}{\Gamma_{\text{AW}}}\frac{\Gamma}{1-\Phi}\right)h_\alpha^\beta(\bar{U}; \kappa\sigma_n). \qquad (13)$$

If $\sigma_n = 0$, then, (12) reduces to

$$T_{\text{error}} = G_0\left(\bar{U} - \text{sat}_\alpha^\beta(\bar{U})\right) \qquad (14)$$

Proof. From (7) and (8), we obtain

$$\Delta\bar{T} = \frac{\Gamma}{1-\Phi}h_\alpha^\beta(\bar{U}; \kappa\sigma_n), \qquad (15)$$

$$\bar{x} = G_0\left(\bar{U} - h_\alpha^\beta(\bar{U}; \kappa\sigma_n)\right),$$

From (9) we obtain $T_{\text{error}} = \bar{x}$, which proves (12).

Using (15) we write

$$T_{\text{error}} = \Delta T_R - \Delta\bar{T} = \Delta T_R - \frac{\Gamma}{1-\Phi}h_\alpha^\beta(\bar{U}; \kappa\sigma_n). \qquad (16)$$

Then combining (12) and (16) yields (13). Equation (14) can be shown since $h_\alpha^\beta(\bar{U}(k); \kappa\sigma_n)$ converges to $\text{sat}_\alpha^\beta(U(k))$ as $(K_p + K)\sigma_n$ tends to zero. This completes the proof. □

Equation (13) always has a solution, which can be found by either graphically or using bisection algorithm. In particular, for $G(z) = H(z)$, the solution of (13) is given by

$$\bar{U} = \frac{(1-\Phi)\Delta T_R}{\Gamma}.$$

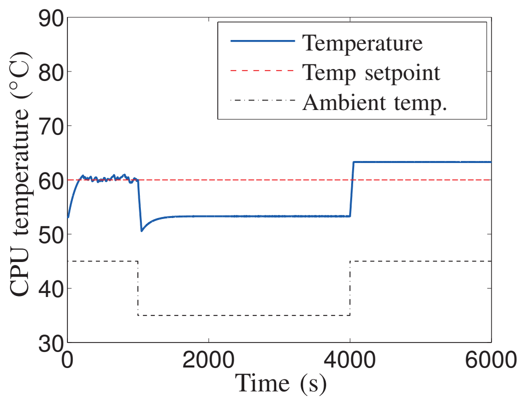


Fig. 5. Response of the PI thermal control system without anti-windup under ambient temperature change.

Theorem 1 implies that the thermal control system with measurement noise yields CPU temperature tracking error, T_{error}, to the level defined by (12). In contrast, when there is no measurement noise, the temperature error is given by (14), which is zero for $\alpha \le \bar{U} \le \beta$. This inequality condition on $\bar{U}$ means that the steady-state utilization level is between the minimum and maximum utilization levels determined a priori, which is a necessary condition for the CPU temperature to be controlled to the desired setpoint T_R. Moreover, Theorem 1 shows that the error would appear for any first order anti-windup compensator, not necessarily same as $H(z)$.

The temperature error in (12) is proportional to the difference between $\bar{U}$ and $h_\alpha^\beta(\bar{U}, \kappa\sigma_n)$. As one can see in Fig. 4, 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 σ_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.

It should be pointed out that the noise-induced error is not due to the particular form of TCUB [3]. For any thermal control systems, due to its dynamics, PI control is needed to ensure zero steady state error and fast response. Also, the bounded actuation is inevitable, which necessitates the anti-windup. Thus, any thermal control systems would involve a certain form of PI control with anti-windup. Therefore, the noise-induced error takes place in any reasonably designed thermal control systems.

Since the temperature error is the product of the d.c. gain G_0 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 [9].

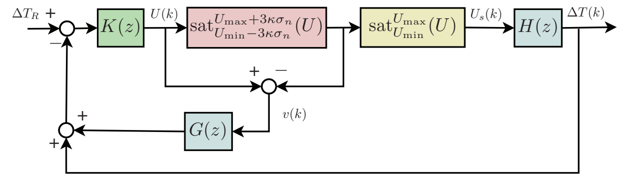


Fig. 6. Thermal control system with virtual saturation.

In fact, Fig. 5 shows the response of the PI thermal control system without anti-windup when ambient temperature dropped between 1,000 and 4,000 seconds due to a sudden operating environment change. From 0 to 1,000 seconds, noise-induced temperature error does not take place as predicted by Theorem 1. However, controller windup phenomenon occurs at 4,000 seconds, which makes the CPU temperature exceed the setpoint for a prolonged period of time. The desired target temperature is not achieved from 1,000 to 4,000 seconds although CPU is running tasks at the maximum utilization level, due to lower ambient 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.

V. 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.

The proposed scheme, referred to as Thermal Control under Utilization Bound with Virtual Saturation (TCUB-VS), is illustrated in Fig. 6. In order to make $U(k)$ lie inside the linear region, TCUB-VS introduces a virtual saturation block with $\text{sat}_{\alpha'}^{\beta'}(U(k))$, where $\alpha' = \alpha - 3\kappa\sigma_n$ and $\beta' = \beta + 3\kappa\sigma_n$. The margin of $3\kappa\sigma_n$[3] 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. 7. 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 $[0.1, 0.67]$.

[3]We have chosen three standard deviations in an arbitrary manner. We can use a different value if needed.

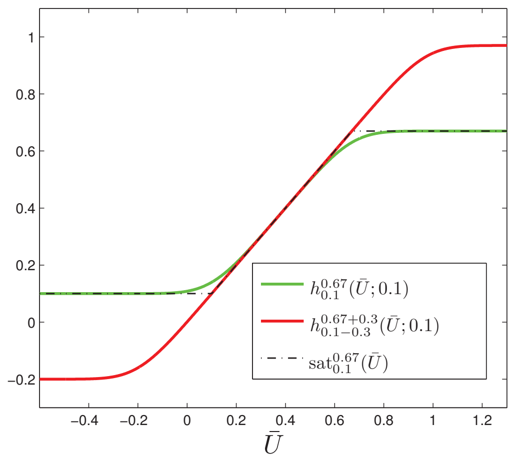


Fig. 7. Comparison of $\mathrm{sat}_{\alpha}^{\beta}(\bar{U})$, $h_{\alpha}^{\beta}(\bar{U};\kappa\sigma_n)$, and $h_{\alpha'}^{\beta'}(\bar{U};\kappa\sigma_n)$ with $\alpha = 0.1$, $\beta = 0.67$, $\kappa\sigma_n = 0.1$, $\alpha' = 0.1-0.3$, and $\beta' = 0.67+0.3$.

As illustrated in Fig. 7, for the steady-state utilization level $\bar{U}$ that satisfies $\alpha \leq \bar{U} \leq \beta$, we obtain $h_{\alpha'}^{\beta'}(\bar{U}(k);\kappa\sigma_n) \approx \bar{U}$. Therefore, from (8) 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.

VI. Performance Evaluation

In order to evaluate the performance of the proposed method, we use the simulator introduced in [3]. The simulation environment consists of two components which are an utilization controller and a thermal control system. The former is implemented in C++, the latter is in Matlab/Simulink, and the two are connected through TCP connection.

In our simulation there exist 10 periodic soft real-time tasks. The Rate Monotonic (RM) scheduling algorithm [10] is used to handle all these tasks in the utilization controller. First, each task T_i has a period randomly generated in the range of $[100\ \mathrm{ms}, 200\ \mathrm{ms}]$. Next, the simulator chooses the execution time of all tasks when each task has nearly equal utilization and the whole utilization does not exceed the utilization bound. The deadline of each task equals its period.

Table I shows thermal related parameters that are given in [3]. All parameters except thermal capacitance in Table I are obtained from Intel technical specification [11]. The thermal capacitance is obtained by simulating Pentium 4 processor on Hotspot [12], which is an architecture-level simulator. Table II shows the controller parameters of the simulator.

In this section, we evaluate the performance of TCUB and TCUB-VS in various scenarios that include operation under nominal condition, operation with different noise characteristics, a case when mismatch exists between actual power and estimated power, and a case when Alpha 21264 CPU is used.

A. Experiment I: Motivating Scenarios Revisited

In Section II, TCUB cannot achieve the desired CPU temperature when temperature sensor noise is present. This

TABLE I
Thermal Related Parameters

Parameter	Notation	Value
Nominal ambient temperature	T_0	$45°\mathrm{C}$
Estimated Active power	P_a	$51.9W$
Idle power	P_i	$13.3W$
Thermal Capacitance	C_{th}	$295.7J/K$
Thermal Resistance	R_{th}	$0.467K/W$

TABLE II
Controller Parameters

Parameters	Notation	Value
Thermal sensor error standard deviation	σ	1
Control system gains	K_p	0.0523
	K	0.5329
	κ	0.8143
Utilization upper bound	$U_{\max}$	0.67
Utilization lower bound	$U_{\min}$	0.1
Thermal controller sampling time	T_s	10s
Utilization controller sampling time	T_u	1s

result is obtained from the simulation with thermal resistance being twice the nominal value, i.e., $R_{\text{th}} = 0.934K/W$. Higher thermal resistance can be caused by CPU fan failure or blocking of the case air vent.

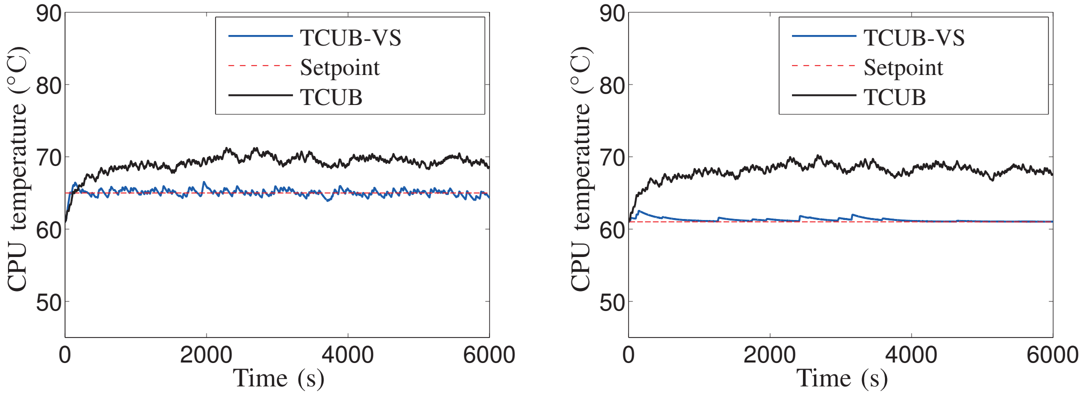
Figure 8 shows the CPU temperature responses of TCUB-VS and TCUB under the same experimental condition. Figure 8(a) shows the case of $T_R = 65°\mathrm{C}$. Clearly, response of TCUB-VS does not show any measurement noise induced temperature error while that of TCUB exhibits severe bias. Figure 8(b) shows the case of $T_R = 61°\mathrm{C}$, which again confirms that noise induced error is eliminated in TCUB-VS.

B. Experiment II: Controller Windup Revisited

In Section IV, we showed that removing anti-windup eliminated noise-induced temperature error, but suffered from controller windup phenomenon. The simulation scenario included changes in ambient temperature due to a sudden operating environment shift. We evaluate the performance of TCUB-VS under the same scenario. Figure 9 shows the responses of both TCUB without anti-windup and TCUB-VS. The two responses are almost identical up to 4,000 seconds and does not show any noise-induced temperature error. However, after 4,000 seconds, TCUB-VS does not exhibit controller windup and quickly achieves the desired temperature while TCUB without anti-windup fails to do so.

C. Experiment III: Various Noise Characteristics

As the major goal of thermal control is to prevent CPU overheating, we analyze the effect of noise standard deviation



(a) Thermal control response with $T_R = 65°\text{C}$ (b) Thermal control response with $T_R = 61°\text{C}$

Fig. 8. Motivation scenario revisited.

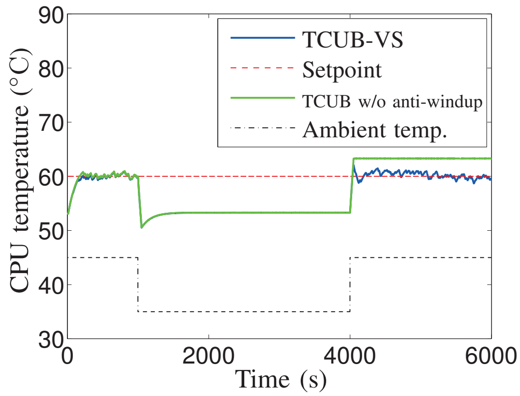


Fig. 9. Controller windup revisited.

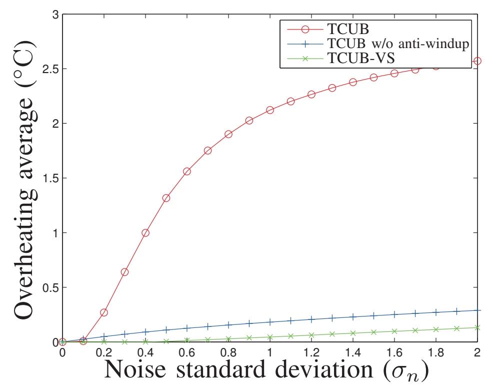


Fig. 10. Effect of noise σ_n on CPU overheating.

on CPU overheating following using the metric introduced in [2]. The metric, that we refer to as overheating average, is the average of the temperature that exceeds the desired target. Since the overheating average takes into account only the portion of the temperature that is higher than the desired, it quantifies the degree of CPU overheating better than the usual average does.

Figure 10 shows the overheating average as a function of noise standard deviation when the temperature setpoint is $55°\text{C}$. As the noise standard deviation increases, the overheating average of TCUB increases, whereas those of TCUB-VS and TCUB without anti-windup remain nearly unchanged. As shown in this experiment, measurement noise can severely degrade thermal control performance and lead to CPU overheating, if not properly handled. The proposed solution of TCUB-VS effectively prevents such situations.

Although the analysis presented in the paper is valid for Gaussian noise, we further investigate this phenomenon for zero mean uniform random noise. The target temperature is $55°\text{C}$ and other simulation parameters are given in Tables I and II. Virtual saturation level is enlarged by $3\kappa\sigma_n$. Figure 11 shows the response of thermal control system when $n(k)$ is zero-mean uniform random noise with standard deviation of 1. Clearly, TCUB shows noise-induced error, which indicates that the phenomenon takes place not only with Gaussian but also with other types of zero-mean noise. Notice that TCUB-VS does not show any noise-induced error, which indicates that it is an effective solution for noise other than Gaussian characteristics as well.

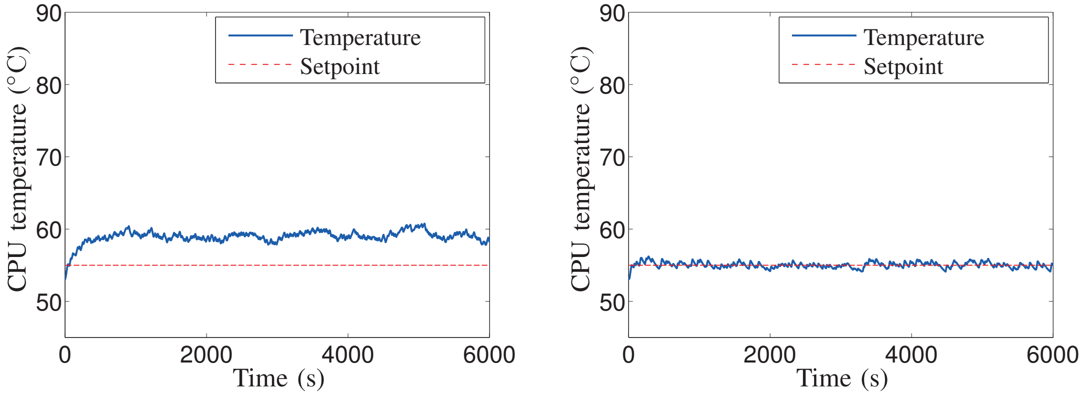
D. Experiment IV: Elevated Ambient Temperature

We consider the scenario where ambient temperature is elevated for a period of time. The ambient temperature increases from $45°\text{C}$ to $50°\text{C}$ at 1,000 seconds, and then decrease to $45°\text{C}$ at 4,000 seconds. The responses are shown in Fig. 12.

Response of TCUB shows noise-induced error, indicating CPU overheating between 1,000 and 4,000 seconds, and CPU under utilization from 0–1,000 seconds and 4,000–6,000 seconds, which are clearly undesirable. On the other hand, the proposed method of TCUB-VS succeeds to regulate CPU temperature at the desired value, avoiding noise induced error.

E. Experiment V: Increased Power Ratio

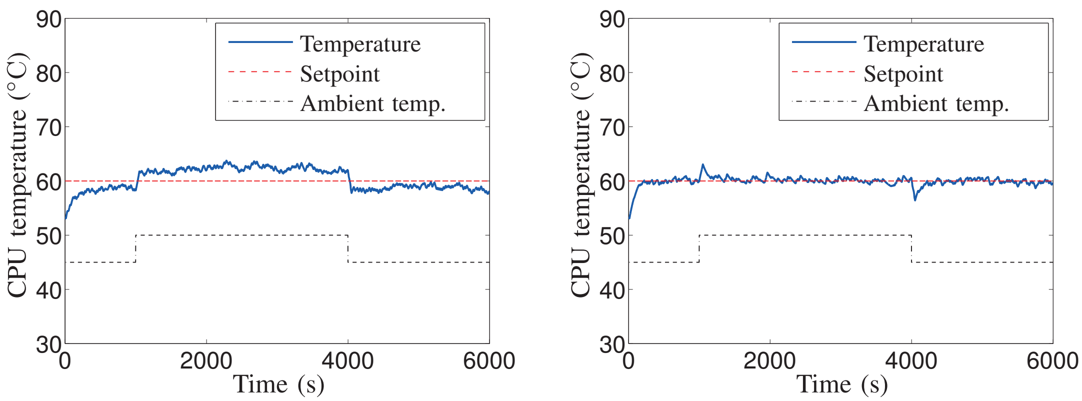
The processor's actual active power P_a may be different from the estimate [13]. The ratio between the actual power and the estimated is called power ratio. In this experiment,



(a) Response of TCUB

(b) Response of TCUB-VS

Fig. 11. Thermal control response with uniform random noise.



(a) Response of TCUB

(b) Response of TCUB-VS

Fig. 12. Thermal control response under elevated ambient temperature.

we consider the scenario where the power ratio is given by 2. Figure 13 represents the CPU temperature responses of TCUB and TCUB-VS. The response of TCUB achieves the setpoint of $65°C$ when the ambient temperature is $45°C$. However, when the ambient temperature is changed to $55°C$, the noise-induced error appears. The CPU temperature exceeds the target by $7°C$. Again, TCUB-VS achieves the temperature setpoint despite the increased power ratio.

F. Experiment VI: Alpha 21264 Processor

Here, we simulated thermal control for another processor Alpha 21264. For this processor, the active power P_a is $72W$, the idle power P_{idle} is $6W$, and thermal resistance R_{th} is $0.4K/W$, all of which are found in [15]. Thermal capacitance C_{th} is $769.6J/K$ which is computed by simulating Alpha 21264 on Hotspot [14]. The temperature responses of TCUB and TCUB-VS are shown in Fig. 14. Similar to other cases, the ambient temperature changes between $45°C$ to $55°C$. The performance of TCUB is poor due to severe level of noise induced error, which leads to both substantial CPU under-utilization and overheating. As can be seen in Fig. 14, TCUB-VS shows much improved performance as it maintains the target temperature.

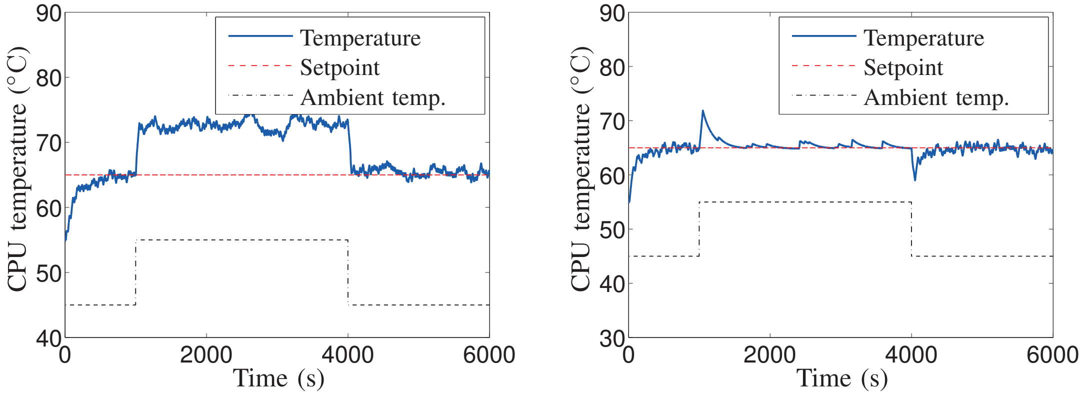
VII. Related Work

As the power density of CPU increases, thermal hotspots and severe temperature variations on the die may occur. Consequently, thermal monitor and management mechanisms are needed to mitigate thermal problems [16].

To this end, thermal-aware real-time scheduling has been substantially studied [17]–[24]. However, these approaches typically rely on accurate models on the system characteristics such as thermal dynamics, task execution times, power consumption, and ambient temperature, which can significantly vary in practice.

There also exist studies on thermal control for real-time systems such as [25]–[27], whose main goal is to resolve the thermal issue by tuning system parameters such as clock frequency, voltage and bandwidth allocation.

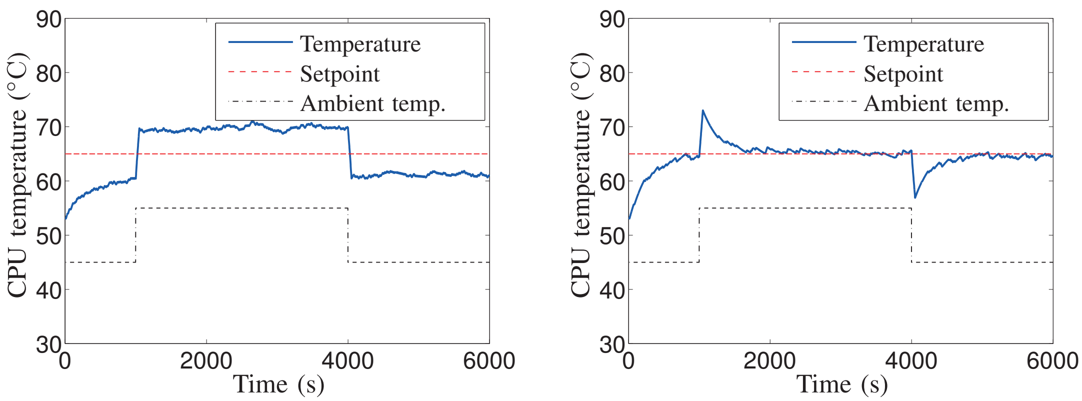
More recently, feedback thermal control methods are proposed for preventing processor overheating by adjusting the task rate or CPU power. In [28], a Thermal-Aware Feedback Control Scheduling (TAFCS) is proposed for soft real-time systems based on a nested feedback control loop structure like TCUB in [3]. The difference between TAFCS and TCUB is that TAFCS uses a miss rate controller instead of the utilization



(a) Response of TCUB

(b) Response of TCUB-VS

Fig. 13. Thermal control response with power ratio equal to 2.



(a) Response of TCUB

(b) Response of TCUB-VS

Fig. 14. Thermal control response with Alpha 21264 processor.

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 CPU. On the contrary, an algorithm called Temperature Regulated Capacity Bound (TRCB) is proposed in [29], whose main goal is to design thermal-resilient hard real-time systems.

A thermal-aware service rate assignment problem in cyber-physical systems is proposed in [30], where a dynamic-programming based service rate assignment is introduced within a feedback control framework in order to prevent system overheating. In order to evaluate the proposed scheme in [30], the author 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 [31], where the controller is based on the nested feedback controller structure that consists of the thermal controller and the resource manager, and a service level table is defined in order to evaluate the proposed resource manager.

Feedback thermal control is also extended to multicore processors. In [32], the authors propose two approaches 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 redistribute the tasks in order to alleviate overheating between the cores. Feedback thermal control on multicore processors is also studied in [33], 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 [34], 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, unlike previous thermal control schemes, TCUB-VS can enforce real-time and thermal constraints under *not only* uncertain system dynamics, *but also* thermal sensor noise, which always exists in practice due to various issues such as sensor accuracy, sensor placement, and processor architecture. To the best of our knowledge, there has not been any thermal control method that takes into account the impact of sensor noise.

VIII. Conclusion

We have incorporated the important practical issue of thermal sensor noise into problem formulation. We have discovered that even a small zero-mean sensor noise can induce a significant steady-state error in processor temperature control. Based on rigorous analysis, we have quantified the error as a closed-form function of noise statistics and system parameters. Then, we have proposed a novel algorithm, TCUB-VS (stands for Thermal Control under Utilization Bounds with Virtual Saturation), which can eliminate the noise-induced error and maintain the desired processor temperature for real-time performance. Our extensive simulation results have shown the key benefits of TCUB-VS.

Acknowledgment

This work was supported by the ICT R&D program of MSIP/IITP (14-824-09-013, Resilient Cyber-Physical Systems Research). The authors greatly appreciate Dr. Y. Fu for the invaluable help on performance evaluation.

References

[1] E. Rotem, J. Hermerding, A. Cohen, and H. Cain, "Temperature measurement in the intel® coretm duo processor," in *Proc. International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, September 2006.

[2] J. Long, S. O. Memik, G. Memik, and R. Mukherjee, "Thermal monitoring mechanisms for chip multiprocessors," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 5, no. 2, p. 9, August 2008.

[3] Y. Fu, N. Kottenstette, Y. Chen, C. Lu, X. D. Koutsoukos, and H. Wang, "Feedback thermal control for real-time systems," in *Proc. Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2010.

[4] A. P. Ferreira, D. Mosse, and J. C. Oh, "Thermal faults modeling using a RC model with an application to web farms," in *Proc. Euromicro Conference on Real-Time Systems (ECRTS)*, July 2007.

[5] F. Bellosa, S. Kellner, M. Waitz, and A. Weissel, "Event-driven energy accounting of dynamic thermal management," in *Proc. Workshop on Compilers and Operating Systems for Low Power (COLP)*, September 2003.

[6] N. Kottenstette, J. Hall, X. Koutsoukos, P. Antsaklis, and J. Sztipanovits, "Digital control of multiple discrete passive plants over networks," *International Journal of Systems, Control and Communications (IJSCC)*, vol. 3, no. 2, pp. 194–228, April 2011.

[7] Y. Eun and E. S. Hamby, "Noise induced loss of tracking in systems with saturating actuators and anti-windup," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 5, p054501 (6 pages), September 2014.

[8] E.-W. Bai, L.-C. Fu, and S. S. Sastry, "Averaging analysis for discrete time and sampled data adaptive systems," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 2, pp. 137–148, February 1988.

[9] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*. Prentice Hall, 2000.

[10] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM (JACM)*, vol. 20, no. 1, pp. 46–61, January 1973.

[11] Intel Corp., "Intel pentium 4 processor in the 423-pin package. thermal design guidelines," Tech. Rep., November 2000.

[12] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam, "Compact thermal modeling for temperature-aware design," in *Proc. Design Automation Conference (DAC)*, June 2004.

[13] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proc. International Symposium on Microarchitecture (MICRO)*, December 2003.

[14] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "Hotspot: A compact thermal modeling methodology for early-stage vlsi design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 501–513, May 2006.

[15] M. K. Gowan, L. L. Biro, and D. B. Jackson, "Power considerations in the design of the alpha 21264 microprocessor," in *Proc. Design Automation Conference (DAC)*, June 1998.

[16] G. Kornaros and D. Pnevmatikatos, "A survey and taxonomy of on-chip monitoring of multicore systems-on-chip," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 18, no. 2, pp. 17:1–17:38, March 2013.

[17] S. Wang and R. Bettati, "Reactive speed control in temperature-constrained real-time systems," *Real-Time Systems*, vol. 39, no. 1–3, pp. 73–95, August 2008.

[18] ——, "Delay analysis in temperature-constrained hard real-time systems with general task arrivals," in *Proc. Real-Time Systems Symposium (RTSS)*, December 2006.

[19] W.-L. Hung, Y. Xie, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, "Thermal-aware task allocation and scheduling for embedded systems," in *Proc. Design, Automation and Test in Europe (DATE)*, March 2005.

[20] J.-J. Chen, S. Wang, and L. Thiele, "Proactive speed scheduling for real-time tasks under thermal constraints," in *Proc. Real-Time and Embedded Technology Applications Symposium (RTAS)*, April 2009.

[21] L. Yuan, S. Leventhal, and G. Qu, "Temperature-aware leakage minimization technique for real-time systems," in *Proc. International Conference on Computer-Aided Design (ICCAD)*, November 2006.

[22] T. Chantem, X. S. Hu, and R. P. Dick, "Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 10, pp. 1884–1897, October 2011.

[23] N. Fisher, J.-J. Chen, S. Wang, and L. Thiele, "Thermal-aware global real-time scheduling on multicore systems," in *Proc. Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2009.

[24] H. F. Sheikh, I. Ahmad, Z. Wang, and S. Ranka, "An overview and classification of thermal-aware scheduling techniques for multi-core processing systems," *Sustainable Computing: Informatics and Systems*, vol. 2, no. 3, pp. 151–169, September 2012.

[25] R. McGowen, C. A. Poirier, C. Bostak, J. Ignowski, M. Millican, W. H. Parks, and S. Naffziger, "Power and temperature control on a 90-nm itanium family processor," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 229–237, January 2006.

[26] M. Saen *et al.*, "Embedded SoC resource manager to control temperature and data bandwidth," in *Proc. International Solid-State Circuits Conference (ISSCC)*, February 2007.

[27] M. S. Floyd, S. Ghiasi, T. W. Keller, K. Rajamani, F. Rawson, J. C. Rubio, and M. S. Ware, "System power management support in the IBM POWER6 microprocessor," *IBM Journal of Research and Development*, vol. 51, no. 6, pp. 733–746, November 2007.

[28] J. Yue, T. Zhang, Y. Liu, B. Quan, and C. Tianzhou, "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)*, June 2012.

[29] P. M. Hettiarachchi, N. Fisher, M. Ahmed, L. Y. Wang, S. Wang, and W. Shi, "The design and analysis of thermal-resilient hard-real-time systems," in *Proc. Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2012.

[30] H.-H. Chu, T.-H. Tsai, and Y.-S. Chen, "Thermal-aware service rate adjustment for cyber-physical systems," in *Proc. International Conference on Anti-Counterfeiting, Security and Identification (ASID)*, August 2012.

[31] V. R. Segovia, M. Kralmark, M. Lindberg, and K.-E. Årzén, "Processor thermal control using adaptive bandwidth resource management," in *Proc. World Congress of the International Federation of Automatic Control (IFAC)*, August 28–September 2, 2011.

[32] M. Becker, K. Sandström, M. Behnam, and T. Nolte, "Dynamic power management for thermal control of many-core real-time systems," in *Proc. Workshop on Adaptive and Reconfigurable Embedded Systems (APRES)*, April 2014.

[33] Y. Fu, N. Kottenstette, C. Lu, and X. D. Koutsoukos, "Feedback thermal control of real-time systems on multicore processors," in *Proc. International Conference on Embedded Software (EMSOFT)*, October 2012.

[34] B. Yun, K. G. Shin, and S. Wang, "Predicting thermal behavior for temperature management in time-critical multicore systems," in *Proc. Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2013.