

Received March 11, 2020, accepted May 6, 2020, date of publication May 18, 2020, date of current version June 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995184

A Novel Mitigation Method for Noise-Induced Temperature Error in CPU Thermal Control

JUSEUNG LEE¹, DOHWAN KIM¹, KYUNG-JOON PARK¹, (Senior Member, IEEE),
AND YONGSOON EUN¹, (Senior Member, IEEE)

Department of Information and Communication Engineering, Daegu Gyeongbuk Institute of Science and Technology (DGIST), Daegu 42988, South Korea

Corresponding author: Yongsoon Eun (yeun@dgist.ac.kr)

This work was supported in part by the Institute for Information and communications Technology Promotion (IITP) Grant funded by the Korean Government (MSIP) (Resilient Cyber-Physical Systems Research) under Grant 2014-0-00065, in part by the Global Research Laboratory Program through the National Research Foundation of Korea under Grant NRF-2013K1A1A2A02078326, and in part by the DGIST Research and Development Program of the Ministry of Science and ICT under Grant 18-ST-02 and Grant 18-EE-01.

ABSTRACT It has been reported that in the thermal control of real-time computing systems, zero-mean thermal sensor noise can induce a significant steady-state error between the target and actual temperatures of a CPU. Unlike the usual case of zero-mean sensor noise resulting in zero-mean temperature fluctuations around the target value, this noise-induced temperature error manifests in the form of a bias, i.e., the mean of the error is not zero. Existing work has analyzed the main cause of this error and produced a solution, known as TCUB-VS. However, this existing solution has a few drawbacks: the transient response is sluggish, and the exact value of the noise standard deviation is necessary in the design stage. In this paper, we propose a novel method of avoiding noise-induced temperature error while overcoming the limitations of the existing work. The proposed method uses an estimated CPU temperature for the part of the controller that is sensitive to noise while using actual measurements for the other part of the controller. In this way, our proposed method eliminates noise-induced temperature error and overcomes the drawbacks of the existing work. To show the efficacy of our proposed method, theoretical results are obtained using a stochastic averaging approach, and experimental results are presented along with simulations.

INDEX TERMS Zero-mean thermal sensor noise, noise-induced error, thermal control for real-time systems.

I. INTRODUCTION

In real-time computing systems, it is crucial to maintain the CPU temperature at a certain desired value because overheated CPUs suffer serious performance degradation, whereas if the CPU is cooler than the target temperature, this often implies resource underutilization. The goal of CPU thermal control is to maintain the temperature of a CPU at the desired value. The main challenges in CPU thermal control design for real-time systems include (i) satisfying both the real-time constraint and the thermal constraint, (ii) doing so in the face of uncertainty in the system dynamics, and (iii) overcoming the effect of thermal sensor noise. Here, the thermal constraint refers to the need to keep the CPU temperature below a given value to prevent CPU overheating. Previous studies on thermal-aware real-time scheduling have been reported by Wang and Bettati [1], [2] and Hung *et al.* [3], who

addressed problem (i) by using feedforward-based schemes. In feedforward-based schemes, an accurate system model is required to prevent CPU overheating. However, such a system model is usually subject to uncertainties in various aspects, such as thermal dynamics, power consumption, task execution time, and ambient temperature; consequently, feedforward-based schemes cannot effectively solve the CPU overheating problem. Therefore, to avoid CPU overheating (problem (i)) while addressing uncertain system dynamics (problem (ii)), feedback-based thermal control schemes have been proposed by Yue *et al.* [4], Hettiarachchi *et al.* [5], and Fu *et al.* [6]. Kim *et al.* [7] revealed that problem (iii) is also of significant importance because zero-mean noise may result in CPU overheating under utilization constraints. Thermal noise in CPU temperature measurements is very common, and the level of noise can often be substantial in practice. Indeed, the sensor noise amplitudes in the work of Rotem *et al.* [8] and Long *et al.* [9] were as large as 10°C. Kim *et al.* [7] addressed problems (i), (ii), and (iii) within the Thermal Control under

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

Utilization Bound (TCUB) framework, which was proposed by Fu *et al.* [6]. In the work of Fu *et al.* [6], the CPU utilization level is controlled based on CPU temperature feedback to maintain the temperature at a certain target. Kim *et al.* [7] showed that in TCUB systems, thermal sensor noise can induce a significant steady-state error between the desired and actual CPU temperatures. This phenomenon is referred to as noise-induced temperature error (NITE).

A similar phenomenon has been reported in toner concentration control in printing systems by Eun and Hamby [10] and Kabamba *et al.* [11], and the feedback control system structure that is conducive to this phenomenon has been generalized by Lee and Eun [12].

Fortunately, Kim *et al.* [7] proposed a method called Thermal Control under Utilization Bound with Virtual Saturation (TCUB-VS) to eliminate the NITE phenomenon. However, this method requires accurately measuring the standard deviation of the sensor noise, which varies over time and with the measurement conditions. In addition, TCUB-VS can result in a sluggish transient response of the CPU temperature.

Because NITE occurs due to measurement noise, an alternate mitigation method is to place a low-pass filter in the feedback loop. This, however, also causes a sluggish response of the control system and can affect the system stability (See Choi and Sul [13], Satici *et al.* [14], Li *et al.* [15], and Park and Kim [16].)

The main goal of this paper is to propose a novel method to address problems (i), (ii), and (iii) mentioned above while overcoming the drawbacks of TCUB-VS. The proposed method is named TCUB-NR, which stands for *Thermal Control under Utilization Bound with Noise Reduction*. According to the analysis presented by Kim *et al.* [7], Eun and Hamby [10], and Lee and Eun [12], a critical mechanism for NITE mitigation relies on a proportional control term, which is sensitive to noise, and a utilization bound in the thermal controller. Therefore, the main idea of TCUB-NR is to use an estimated CPU temperature, instead of noisy temperature measurements, for the proportional control term of the thermal controller. This estimated temperature is obtained by using a nominal thermal model of the CPU, whose input is the CPU utilization. Note that the novel feature of TCUB-NR is that to eliminate NITE, the thermal controller is considered to be divided into two parts. One part includes a proportional control term that is sensitive to noise, and the other part is not sensitive to noise. TCUB-NR uses the estimated temperature for the noise-sensitive proportional control term but uses the actual sensor output for the noise-insensitive part of the controller. In this manner, TCUB-NR eliminates the effect of sensor noise in the part of the controller that is critical for NITE mitigation while maintaining the feedback control function by using the actual sensor output for the other part of the controller. If the estimated temperature were to be used for all aspects of control, the real-time feedback nature of the controller would be lost; however, this is avoided in the TCUB-NR scheme. In turn, because the effect of noise on the NITE is reduced, the extended linear range of the virtual

saturation block used in TCUB-VS is no longer needed, thus eliminating the main cause of the sluggish response. In addition, our proposed method does not require the accurate standard deviation of the noise, which is used to set limits for the virtual saturation in TCUB-VS. Hence, the drawbacks of the existing approach are alleviated. To prove that our proposed method is effective in mitigating NITE, a theoretical analysis of NITE elimination with TCUB-NR is presented using the stochastic averaging approach of Skorokhod [17]. Additionally, a performance comparison between TCUB-VS and TCUB-NR is shown to illustrate the improvement achieved with TCUB-NR.

The outline of this paper is as follows. In Section II, we describe the thermal control problem for real-time systems and show that the NITE phenomenon occurs in TCUB. Additionally, the TCUB-VS mechanism is explained. Section III shows that TCUB-NR mitigates the NITE phenomenon while overcoming the limitations of the TCUB-VS method. Moreover, the NITE mitigation effect of TCUB-NR is analyzed using the stochastic averaging approach of Skorokhod [17]. The TCUB-NR scheme proposed in this paper for eliminating NITE is validated with both simulated and experimental results in Section IV. Finally, our conclusions are given in Section V.

II. NITE IN CPU THERMAL CONTROL SYSTEMS

A. OVERVIEW OF TCUB

TCUB is a feedback-based control approach for maintaining a desired CPU temperature by controlling CPU utilization based on measurements. In this approach, the CPU temperature is measured, and then the utilization level is adjusted based on the measured value, the desired temperature setpoint, and a model of the thermal dynamics of the unit.

The overall structure of TCUB is shown in Figure 1. It consists of a controller and a processor. The controller has two feedback loops. The inner loop, shown in red, is responsible for utilization control, and the outer loop, in blue, is responsible for thermal control. Since the processor utilization dynamics are much faster than the temperature dynamics, the outer loop runs at much lower sampling and control rates than the inner loop. The index k represents the sampling instances for the outer thermal control loop, and k' represents the sampling instances for the inner utilization loop.

The setpoint temperature and the maximum and minimum utilization bounds, denoted by T_R , β , and α , respectively, are provided as input to the thermal controller. The minimum utilization bound α may be defined as the sum of the product of each minimum achievable task execution time with the corresponding minimum allowable task rate. The maximum utilization bound β may be determined by the scheduler-dependent utilization bound at which real-time tasks can miss a deadline.

In the k -th sampling period of the outer loop, based on the measured temperature $T(k)$ provided by the thermal sensor, the thermal controller calculates the utilization setpoint $u_s(k)$

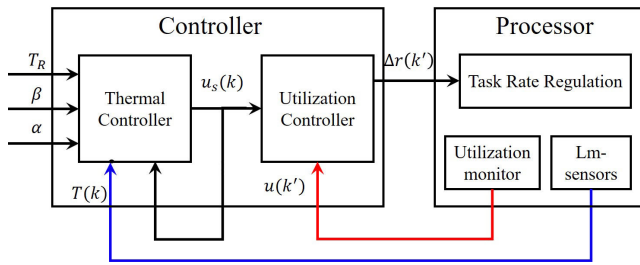


FIGURE 1. The overall structure of TCUB.

for the utilization controller in the inner loop. In the k' -th sampling period of the inner loop, based on the measured utilization $u(k')$, the utilization controller adjusts the change in the task rate $\Delta r(k')$ and then sends it to the actual processor to drive the processor utilization to converge to the setpoint $u_s(k)$ given by the thermal controller [6], [7].

For the thermal dynamics of the processor, by combining the thermal RC model [18], [19] with the relationship between CPU power and utilization [6], the model to be employed in thermal control can be obtained as follows [6]:

$$T(k + 1) = \Phi T(k) + R_{th}(1 - \Phi)(G_p P_a - P_{idle})u_s(k) + R_{th}(1 - \Phi)P_{idle} + (1 - \Phi)T_0, \quad (1)$$

where $T(k)$ is the temperature of the processor, T_0 is the ambient temperature, R_{th} is the heat resistance, G_p is the ratio between the actual active power at run time and the estimated active power P_a , and P_{idle} is the power when the CPU is idle. In addition, $\Phi = \exp(-T_s/(R_{th}C_{th}))$, where T_s is the control sampling interval and C_{th} is the heat capacity.

B. NITE IN TCUB

Figure 2 shows a block diagram of a real-time TCUB system [6], where $C(z)$, $G(z)$, and $H(z)$ denote a proportional-integral (PI) controller, an anti-windup controller, and the transfer function between the CPU utilization and CPU temperature, respectively. All transfer functions are given in the form of discrete-time transfer functions. Here, the transfer function $H(z)$ models the combined dynamics of the utilization controller (the inner loop) and the processor in Figure 1, whereas $C(z)$ and $G(z)$ are components of the thermal controller in Figure 1.

Figure 2 also shows the thermal sensor noise, denoted by $n(k)$. Here, the thermal sensor noise, along with other

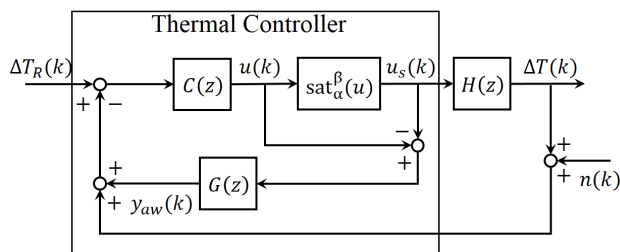


FIGURE 2. A block diagram of a thermal control system with measurement noise.

noise sources, is assumed to have a zero-mean Gaussian distribution with a standard deviation of σ_n because it is approximated as Gaussian-distributed noise [20]. The upper and lower bounds on the CPU utilization in TCUB are represented by the saturation $\text{sat}_\alpha^\beta(u)$, with the following definition:

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

where α and β denote lower and upper utilization bounds, respectively.

The signals $\Delta T(k)$, ΔT_R , and $u(k)$ are the CPU temperature deviation, the CPU temperature setpoint, and the utilization command, respectively, all obtained around a designed operating point of $T(k) = 67^\circ\text{C}$. Additionally, the signal $u_s(k)$ corresponds to $\text{sat}_\alpha^\beta(u)$. The difference between $u(k)$ and $u_s(k)$ is used as an input to the anti-windup controller. The transfer functions $H(z)$ and $G(z)$ are given by

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

where $\Gamma = (G_p P_a - P_{idle})R_{th}(1 - \Phi)$. The controller $C(z)$ is given by

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

where $K = K_i(1 + \frac{w_I T_s}{2})$ and $b = \frac{2 - w_I T_s}{2 + w_I T_s}$. The constants K_p , K_i , and w_I are the original gains of the PI controller in a continuous-time design, rather than the discrete-time design; for details, see [6], [7].

Reference [7] shows that in the thermal control feedback system shown in Figure 2, due to measurement noise, the mean value $\Delta T(k)$ of the CPU temperature deviation in the steady state deviates from the setpoint ΔT_R . The implications of this are illustrated in Figure 3, where the mean value of the CPU temperature deviation, as obtained through simulation, is plotted as a function of the CPU temperature setpoint. Clearly, in the presence of $n(k)$ with $\sigma_n = 3.5$, the CPU temperature is either higher or lower than the setpoint. For comparison, Figure 3 also shows the response when $n(k) = 0$ for all $k \geq 0$, represented by the black line. Note that no error exists in the range of setpoints from 51°C to 83°C . However, even if the noise is small, there are a few conditions under which NITE occurs (see [10] for details). The temperature deviation outside of this setpoint range is due to the limits on the utilization level imposed by α and β .

C. TCUB-VS

The analysis presented in [7] reveals that NITE is caused by undesired persistent triggering of $G(z)$ due to measurement noise amplified by the proportional term of the controller. Hence, the analysis in [7] predicts that eliminating $G(z)$ should eliminate NITE. However, since $G(z)$ is necessary for stability and for the transient response (see [21]–[26] for a discussion of the adverse effect without $G(z)$), an alternative solution has been suggested that uses a virtual saturation

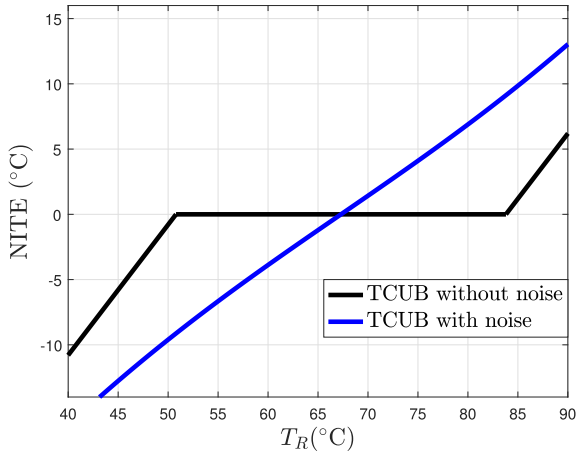


FIGURE 3. Comparison of TCUB without noise and TCUB with noise for $\alpha = 0.1$, $\beta = 0.67$, $G(1) = 57.92$, $(K + K_p) = 0.1107$, and $\sigma_n = 3.5$.

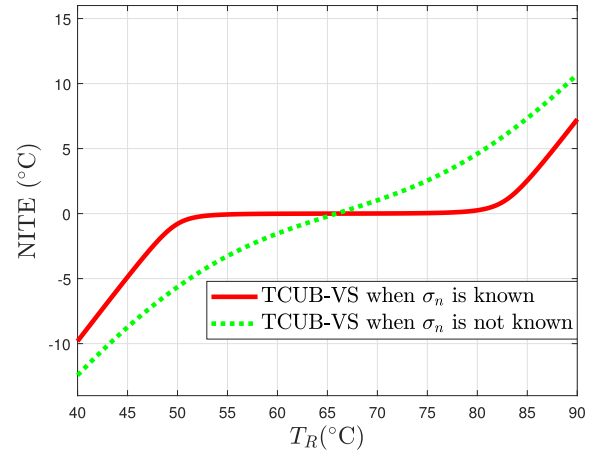


FIGURE 5. NITE in TCUB-VS with $\alpha = 0.1$, $\beta = 0.67$, $G(1) = 57.92$, $(K + K_p) = 0.1107$ when the standard deviation σ_n is not exactly known.

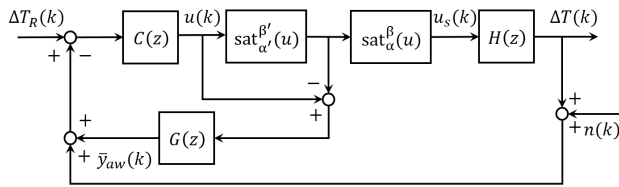


FIGURE 4. A block diagram of a thermal control system with virtual saturation.

block with a linear range that is wider than the actually allowed utilization range. The proposed scheme is illustrated in Figure 4, where all transfer functions, gains, and signals are defined similarly to those in Figure 2. To avoid the continuous triggering of $G(z)$ by noise, the limits of the virtual saturation block are set to

$$\alpha' = \alpha - 3(K + K_p)\sigma_n, \quad \beta' = \beta + 3(K + K_p)\sigma_n.$$

In this way, 99% of the undesired activation of $G(z)$ is avoided because $n(k)$ follows a Gaussian distribution with the standard deviation of σ_n . Note that for the above setup, an accurate estimate of σ_n is required to compute the new limits.

Figure 5 shows the mean CPU temperature deviation as a function of the setpoint. In contrast to Figure 3, the CPU temperature is now controlled without errors. Figure 5 also shows the response when there is a mismatch between the actual noise standard deviation and the estimate. The green line shows the CPU temperature error for the case in which α' and β' are chosen using a value of $\sigma_n = 3.5$ when the actual value is 6. Clearly, some error in the CPU temperature reappears. Therefore, the standard deviation of the measurement noise must be exactly known for TCUB-VS. However, this requirement might be difficult and impractical to fulfill because the sensor characteristics vary over time and with the measurement conditions. This is one of the disadvantages of TCUB-VS. Another disadvantage is that TCUB-VS may result in a sluggish transient response for avoiding controller windup because the virtual saturation delays the activation of anti-windup as a result of the linear range being wider than the actually allowed range of utilization.

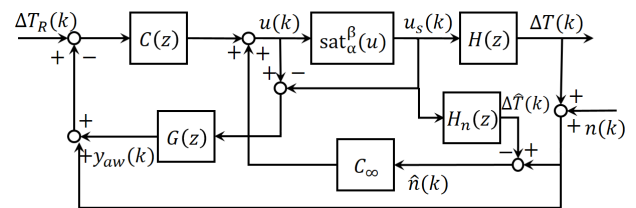


FIGURE 6. A block diagram of a thermal control system with the proposed noise reduction method.

A novel NITE mitigation method that avoids the disadvantages of TCUB-VS is the topic of the next section.

III. A NOVEL METHOD FOR ELIMINATING NITE

We propose an effective method for avoiding NITE in feedback-based thermal control systems. Figure 6 shows the block diagram of a system with the proposed noise reduction method, referred to as Thermal Control under Utilization Bound with Noise Reduction (TCUB-NR). Most signals and transfer functions are defined similarly to those in Figure 2. In addition, $H_n(z)$ is the transfer function of the nominal thermal dynamics of the CPU, which is assumed to be identical to the plant $H(z)$. An estimated CPU temperature $\Delta \hat{T}(k)$, which is not affected by the noise, is obtained from $H_n(z)$, and C_∞ is equal to $C(\infty)$.

TCUB-NR is implemented by using the estimated CPU temperature for the proportional control part of the thermal controller and the actual measured CPU temperature for the other part of the controller. As noted in [7], the NITE is due to undesired triggering of anti-windup caused by the fluctuation of the utilization level induced by the action of measurement noise through the proportional control term. Therefore, TCUB-NR uses an estimated temperature for proportional control, which is sensitive to noise, but uses the actual sensor output for the other part of the controller, which is not sensitive to noise. In this way, TCUB-NR is expected to avoid NITE. Note that the noise-insensitive part of the controller uses the actual temperature measurement in order to maintain the feedback control function. Consequently,

TCUB-NR does not lose its real-time feedback nature while effectively mitigating NITE.

Mathematical support for the above argument is presented as follows. The utilization command $u(k)$ in the TCUB scheme illustrated in Figure 2 is written as

$$u(k) = C(z) (\Delta T_R(k) - y_{aw}(k) - \Delta T(k) - n(k)). \quad (5)$$

We decompose $C(z)$ into the proportional control part, C_∞ , and the remaining part, denoted by $C_1(z)$. In other words, we write $C(z) = C_\infty + C_1(z)$. For the case of the controller given in (4), $C_\infty = (K + K_p)$. Equation (5) shows that in TCUB, both parts of the controller use the actual measured CPU temperature as input: both are multiplied by $\Delta T(k) + n(k)$. Specifically, the term $(K_p + K)n(k)$ represents the sensor noise amplified by the proportional control term, which results in the occurrence of NITE [7].

In contrast, the utilization command $u(k)$ in the TCUB-NR scheme illustrated in Figure 6 is written as

$$\begin{aligned} u(k) = & C_\infty (\Delta T_R(k) - y_{aw}(k) - \Delta T(k) - n(k)) \\ & + C_1(z) (\Delta T_R(k) - y_{aw}(k) - \Delta T(k) - n(k)) \\ & + C_\infty (\Delta T(k) + n(k) - \hat{\Delta T}(k)). \end{aligned} \quad (6)$$

Thus, we obtain

$$\begin{aligned} u(k) = & C(\infty) (\Delta T_R(k) - y_{aw}(k) - \hat{\Delta T}(k)) \\ & + C_1(z) (\Delta T_R(k) - y_{aw}(k) - \Delta T(k) - n(k)). \end{aligned} \quad (7)$$

Comparing (5) and (7) shows that in TCUB-NR, the noise $n(k)$ is not multiplied by C_∞ but instead is multiplied only by $C_1(z)$. Hence, this scheme eliminates the effect of the noise in the proportional control term, i.e., the gain of $(K + K_p)$, but still retains the feedback on the actual measured CPU temperature for the other part of the controller, $C_1(z)$.

Similarly to what is done in [7], [10], [12], the system in Figure 6 is analyzed below using stochastic averaging to mathematically demonstrate the elimination of the NITE. Applying stochastic averaging to the system in Figure 6 with respect to the measurement noise yields the system depicted in Figure 7. The symbols used for the signals involved follow the same notation used in Figure 6 but with an upper bar, which indicates that they represent mean values of the corresponding signals with respect to the randomness caused by noise. Additionally, note that the zero-mean measurement noise $n(k)$ is averaged away and does not appear in the system of Figure 7. The function $h_\alpha^\beta(\bar{u}(k); \kappa\sigma_n)$ is given by

$$\begin{aligned} & h_\alpha^\beta(\bar{u}(k); \kappa\sigma_n) \\ &= \frac{\alpha + \beta}{2} + \frac{\kappa\sigma_n}{\sqrt{2\pi}} \left(e^{-\frac{(\bar{u}(k)-\alpha)^2}{2\kappa^2\sigma_n^2}} - e^{-\frac{(\bar{u}(k)-\beta)^2}{2\kappa^2\sigma_n^2}} \right) \\ & \quad - \frac{\bar{u}(k) - \beta}{2} \operatorname{erf} \left(\frac{\bar{u}(k) - \beta}{\sqrt{2}\kappa\sigma_n} \right) + \frac{\bar{u}(k) - \alpha}{2} \operatorname{erf} \left(\frac{\bar{u}(k) - \alpha}{\sqrt{2}\kappa\sigma_n} \right), \end{aligned} \quad (8)$$

where

$$\operatorname{erf}(\xi) = \frac{2}{\sqrt{\pi}} \int_0^\xi \exp(-t^2) dt. \quad (9)$$

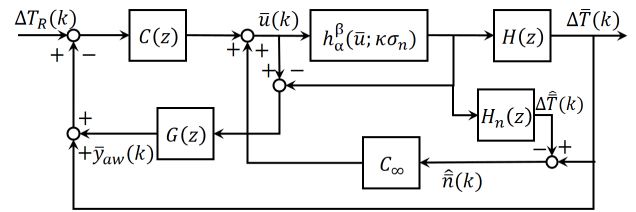


FIGURE 7. A block diagram of the average dynamics of the TCUB-NR system.

The parameter κ represents the part of $u(k)$ that is directly proportional to $n(k)$. In particular, the temperature tracking error signal is denoted by \bar{e} . The NITE in TCUB-NR (the system depicted in Figure 6) is evaluated by analyzing \bar{e} in the system depicted in Figure 7. The accuracy of this approximation has been proven to be high [7], [10], [12]. The NITE in TCUB-NR is given by

$$\bar{e} = G(1) (\bar{u} - h_\alpha^\beta(\bar{u}; \kappa\sigma_n)), \quad (10)$$

where $G(1)$ is the dc gain of the transfer function $G(z)$. The derivation of (10) is given in the Appendix.

According to equation (8) and the controller $C(z)$ given in (4), the function $h_\alpha^\beta(\bar{u}; \kappa\sigma_n)$ is the same as $\operatorname{sat}_\alpha^\beta(\bar{u})$ since $\kappa = C_\infty - (K + K_p) = 0$. Therefore, when the steady-state utilization command \bar{u} is in the linear region, the term $\bar{u} - h_\alpha^\beta(\bar{u}; \kappa\sigma_n)$ in (10) is zero; accordingly, $\bar{e} = 0$ in Figure 7. Therefore, this analysis of (10) reveals that TCUB-NR eliminates the NITE.

According to (10), unlike TCUB-VS, TCUB-NR eliminates the NITE without requiring the exact standard deviation of the noise. In addition, the utilization bounds of TCUB-NR are the same as those of TCUB. Therefore, since there is no delay in the activation of the anti-windup mechanism in TCUB-NR, it is expected that the transient response of the thermal control system should not be degraded compared to the original design.

Regarding system stability, the stability of the system shown in Figure 6 is investigated as follows: When $H(z) = H_n(z) = G(z)$, the stability of the system in Figure 6 is identical to that in Figure 2, which was discussed in [6]. Thus, the system depicted in Figure 6 remains stable.

IV. EXPERIMENTAL VALIDATION

A. EXPERIMENTAL SETUP

For validation, we have implemented TCUB in a computing system. The hardware platform is a Samsung DB-P205 desktop with an Intel i7-870 CPU @2.93 GHz, and the operating system is Linux kernel 4.13 with Ubuntu 16.04.3 LTS. The implemented setup is shown in Figure 8. To implement TCUB, we divided it into three components: the temperature monitoring module, the utilization controller, and the thermal controller. Here, the CPU temperature is measured with lm-sensors. The lm-sensors package provides tools for monitoring CPU temperatures using digital thermal sensors on CPU cores.

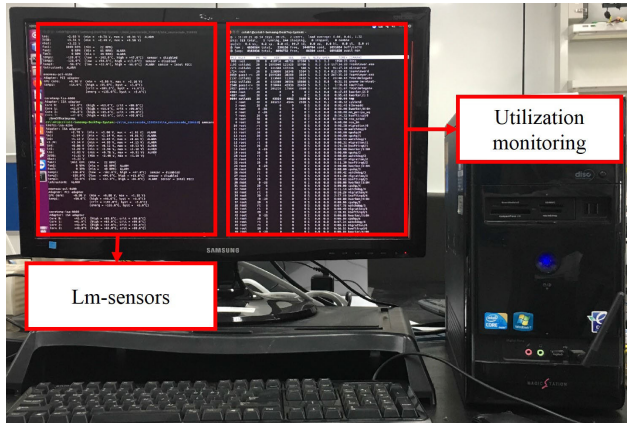


FIGURE 8. CPU thermal control system testbed.

TABLE 1. Parameters related to thermal behavior.

Parameter	Notation	Value
Estimated Active power	P_a	95 W
Idle power	P_{idle}	26.8 W
Thermal Capacitance	C_{th}	309.9 J/K
Thermal Resistance	R_{th}	0.4195 K/W
Thermal Controller	ω_i	0.0036
	K_p	0.0549
	K	0.0558

We generate a workload to vary the CPU utilization by using a cyclic redundancy check (CRC) benchmark. For each core, we run a thread that controls the CPU utilization by regulating the work time and sleep time. We use the system tick counter for the work time and implement the sleep time using the usleep function.

We acquire the thermal resistance from a linear regression of the temperature output with respect to the CPU utilization. Next, the thermal capacitance is obtained from the discrete thermal RC model [18], [19], the CPU power consumption [27], and the transient temperature graph. Table 1 shows the parameters related to the thermal behavior in the experimental setup. Based on Table 1 and the TCUB method as presented in [6], [7], the thermal control system parameters are obtained as follows:

$$H(z) = \frac{4.255}{z - 0.926}, \quad \alpha = 0.1, \beta = 0.67, \\ K_p = 0.0549, \quad K = 0.0558, T_s = 10s. \quad (11)$$

The nominal plant $H_n(z)$ for TCUB-NR is assumed to be identical to $H(z)$.

B. SIMULATION RESULTS

By running Simulink with the transfer functions and parameters given in (11), we obtain MATLAB simulation results for the thermal control systems shown in Figures 2, 4, and 6 for various scenarios.

Under the assumption that the standard deviation σ_n of the measurement noise is 3.5 and the ambient temperature

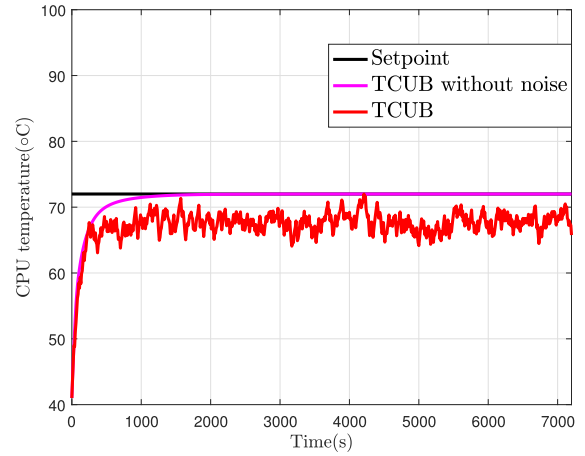


FIGURE 9. Responses of the thermal control system in Figure 2 with $T_R = 72^\circ\text{C}$.

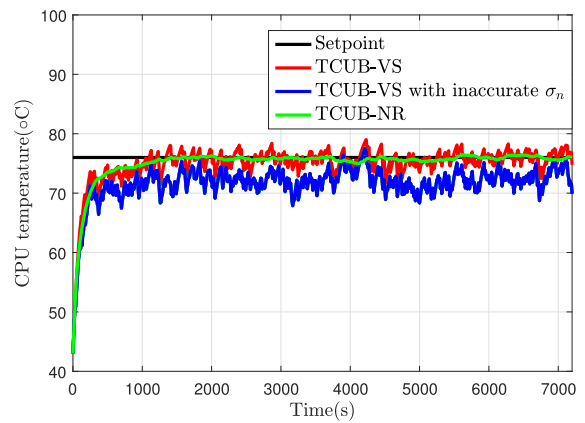


FIGURE 10. Responses of the thermal control systems in Figures 4 and 6 with $T_R = 76^\circ\text{C}$.

is 15°C , Figure 9 shows the response of the thermal control system of Figure 2. Clearly, the CPU temperature shown deviates from the target temperature setpoint. The CPU is underutilized relative to the desired level due to the effect of the noise. In addition, the response of TCUB in the absence of measurement noise is shown for comparison, from which it is clear that the deviation results from the zero-mean measurement noise.

Figure 10 shows the responses of the thermal control systems of Figure 4 (TCUB-VS) and Figure 6 (TCUB-NR). The response of TCUB-VS, shown in red, exhibits no bias, as expected. However, if the estimate of the standard deviation σ_n is off from the true value, bias appears again, as shown in blue. The response of TCUB-NR is shown in green; here, no bias exists, and the fluctuation is much less. Hence, TCUB-NR eliminates the NITE without requiring knowledge of the noise characteristics.

In the next simulation scenario, the CPU temperature setpoint is 70°C , and the ambient temperature changes due to a sudden shift in the operating environment. After 1800 seconds, the ambient temperature rises. Figure 11 shows the responses of TCUB-VS and TCUB-NR in this scenario.

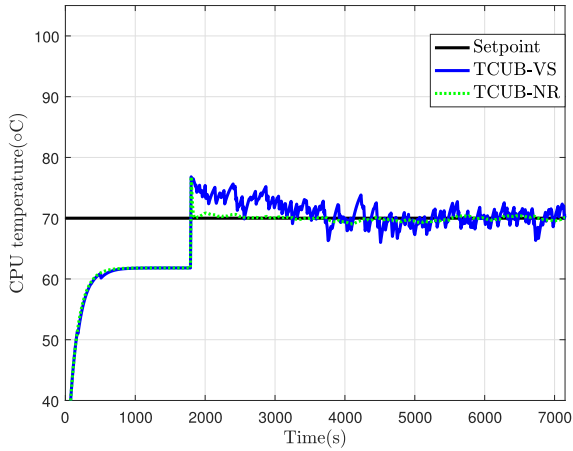


FIGURE 11. Responses of the thermal control systems in Figures 4 and 6 to a sudden shift in the operating environment.

For the first 1800 seconds, neither control system can achieve the desired CPU temperature because of the upper limit on the utilization. During this time, controller windup occurs. At 1800 seconds, the environment changes such that the desired temperature can be achieved. Note that TCUB-VS, as shown in blue, takes thousands of seconds to recover from the windup. This is because the anti-windup activation is delayed due to the wide linear range of the virtual saturation block. On the other hand, TCUB-NR, as shown in green, recovers immediately. Hence, it is shown that TCUB-NR overcomes the sluggish transient response of TCUB-VS, although neither TCUB-VS nor TCUB-NR exhibits any NITE.

Note that NITE occurs in TCUB when CPU power is increased by twice [7]. Also, [7], [30] show the occurrence of NITE in various CPUs (Alpha 21264, Pentium 4, i7-2620M). Therefore, NITE can occur regardless of CPU types or power. In addition, the combined outcome of all sources of noise including power spike etc., is approximated as thermal sensor noise n . Therefore, TCUB-NR is effective to avoid NITE although power spike become larger as CPU power P_a increases. For validation, Figure 12 shows the responses of TCUB and TCUB-NR as the existing CPU power increases from 95W to 123 or 205W by overclocking the CPU in accordance with [31], [32]. Specifically, responses of TCUB and TCUB-NR are, respectively, shown in solid and dotted green when the CPU power P_a is 123W. When the CPU power P_a is 205W, responses of TCUB and TCUB-NR are shown in solid and dotted red, respectively. As shown in Figure 12, it has been confirmed that NITE is mitigated in TCUB-NR regardless of CPU power.

C. EXPERIMENTAL RESULTS

First, Figure 13 shows experimental results for thermal control systems implementing the TCUB, TCUB-VS, and TCUB-NR schemes, respectively. For these experimental results, the target temperature is 65°C. In Figure 13, the response in red shows that TCUB exhibits NITE. The

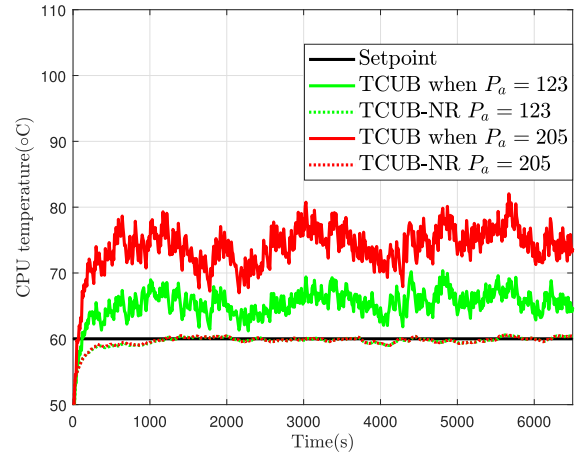


FIGURE 12. Responses of the thermal control systems in Figures 4 and 6 as CPU power P_a is increased.

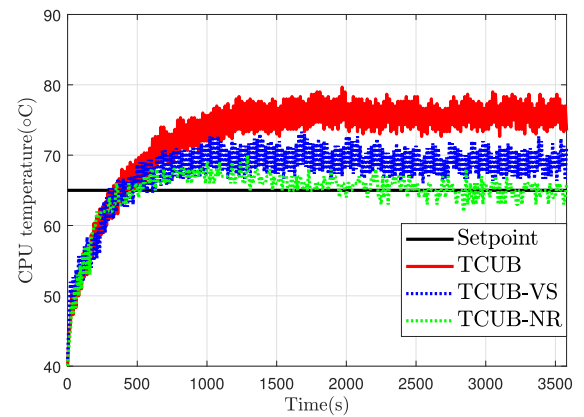


FIGURE 13. Experimental responses of thermal control systems with the system parameters given in (11).

response in blue is TCUB-VS designed with the estimate of σ_n to be 1 while the actual value is about 4.5. Clearly, NITE is not well mitigated. By contrast, the response in green shows that TCUB-NR, our proposed method, effectively eliminates NITE and noise standard deviation is not necessary.

Similar to the simulation results in Figure 11, Figure 14 includes a change in the ambient temperature due to a sudden shift in the operating environment. This experimental setup is to demonstrate an improved transient response by TCUB-NR. For this reason, we use TCUB-VS design with accurate σ_n . The three responses in Figure 14 are almost the same until 1800 seconds, before which they cannot achieve the desired CPU temperature due to the low ambient temperature.

Sluggish transient response means it takes longer for CPU temperature to reach the set-point. The reason for this is that it takes time for controller states to come out of windup [21]. In Figure 14 (experimental results), TCUB-VS (blue line) takes about 1500 seconds (3300-1800) to recover from windup and reach the setpoint while TCUB-NR (green dotted line) takes only about 700 seconds (2500-1800). Note that TCUB (red line) never reaches the setpoint due to NITE.

Notably, the fluctuation with TCUB-NR is also much less than that with TCUB or TCUB-VS due to the noise

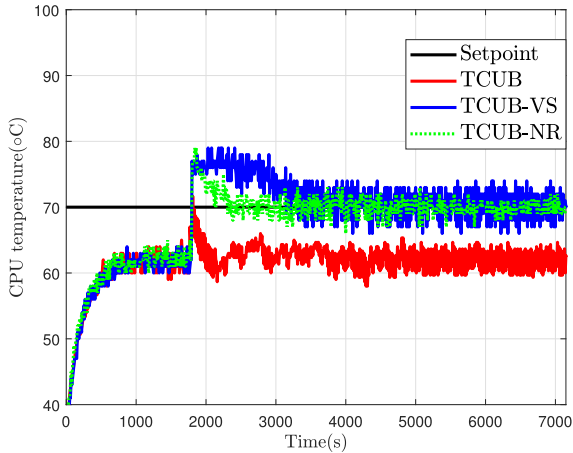


FIGURE 14. Experimental responses of thermal control systems with the system parameters given in (11) in the case of a sudden shift in the operating environment.

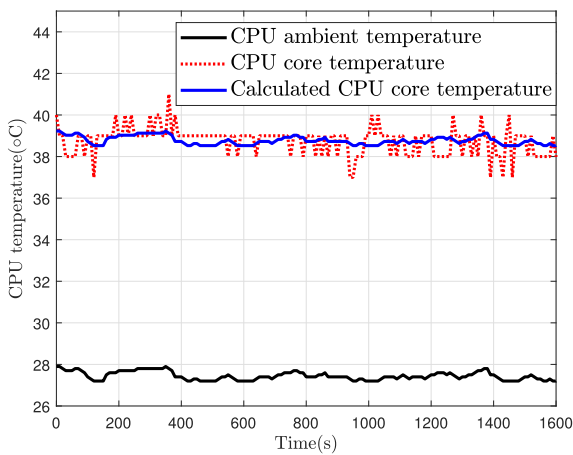


FIGURE 15. CPU core temperature and calculated CPU core temperature.

reduction mechanism of TCUB-NR. Both the simulations and the experimental results show that the proposed TCUB-NR method of thermal control effectively eliminates NITE. Additionally, the experimental validation and the simulation results support the claim that TCUB-NR overcomes the two main drawbacks of TCUB-VS.

In order to verify the accuracy of the CPU core temperature measurement using the existing thermal sensor (1m sensor package), we conducted an experiment using a thermocouple device. According to [6], the CPU core temperature T can be calculated by measuring CPU ambient temperature T_0 , that is, $T = R_{th}P_{th} + T_0$. Hence, we placed the thermocouple device in a close proximity of the CPU case to measure T .

Figure 15 show CPU core temperature obtained from the 1m sensor and that calculated using the above equation and measurement of T_0 , when CPU is idle. In Figure 15, the thermocouple measured CPU ambient temperature is shown in black. Based on the ambient temperature, the calculated CPU core temperature in blue is obtained. Finally, the CPU core temperature obtained from the existing 1m sensor is shown in red. Clearly, the CPU core temperature from the existing

thermal sensor (red line) is very close to the calculated CPU core temperature (blue line).

V. CONCLUSION

In thermal control for real-time computing systems, noise-induced temperature error occurs. In this paper, a new approach called Thermal Control under Utilization Bound with Noise Reduction (TCUB-NR) is proposed and shown to effectively eliminate this noise-induced bias while simultaneously overcoming a few drawbacks of an existing solution. The proposed method achieves CPU thermal control in the presence of model uncertainty and measurement noise. It is conjectured that similar errors induced by noise will occur for other types of noise distributions. Therefore, for mitigating the error induced by non-Gaussian-distributed noise, the potential of existing methods, such as minimum entropy [28], [29], is an open problem that could be investigated in future work.

APPENDIX

A. ANALYSIS OF NITE IN TCUB-NR

Let the state-space realization of the thermal control system illustrated in Figure 6 be

$$\begin{aligned}
 x(k+1) &= \Phi x(k) + \Gamma \text{sat}_\alpha^\beta(u(k)), \\
 x_n(k+1) &= \Phi x_n(k) + \Gamma \text{sat}_\alpha^\beta(u(k)), \\
 x_{aw}(k+1) &= \Phi x_{aw}(k) + \Gamma(u(k) - \text{sat}_\alpha^\beta(u(k))), \\
 x_i(k+1) &= x_i(k) + K(1-b)(\Delta T_R(k) - y_{aw}(k)) \\
 &\quad - K(1-b)(\Delta T(k) + n(k)), \\
 \Delta T(k) &= x(k), \\
 \Delta \hat{T}(k) &= x_n(k), \\
 y_{aw}(k) &= x_{aw}(k), \\
 u(k) &= (K_p + K) (\Delta T_R(k) - \Delta \hat{T}(k)) - \kappa n(k) \\
 &\quad + x_i(k), \tag{12}
 \end{aligned}$$

where $x(k)$, $x_n(k)$, $x_i(k)$, and $x_{aw}(k)$ are the states of the plant, nominal plant, PI controller, and anti-windup controller, respectively, in the system of Figure 6. The signals $y_{aw}(k)$, $\Delta \hat{T}(k)$, and $\Delta T(k)$ are the outputs of the anti-windup controller, the nominal plant, and the actual plant, respectively. Additionally, κ is equal to $C_\infty - (K + K_p)$. By using the stochastic averaging approach, the state-space realization of the system depicted in Figure 7 can be obtained as follows:

$$\begin{aligned}
 \bar{x}(k+1) &= \Phi \bar{x}(k) + \Gamma h_\alpha^\beta(\bar{u}(k); \kappa \sigma_n), \\
 \bar{x}_n(k+1) &= \Phi \bar{x}_n(k) + \Gamma h_\alpha^\beta(\bar{u}(k); \kappa \sigma_n), \\
 \bar{x}_{aw}(k+1) &= \Phi \bar{x}_{aw}(k) + \Gamma \bar{u}(k) \\
 &\quad - \Gamma h_\alpha^\beta(\bar{u}(k); \kappa \sigma_n),
 \end{aligned}$$

$$\begin{aligned}
\bar{x}_i(k+1) &= \bar{x}_i(k) + K(1-b)(\Delta T_R(k) - \bar{y}_{aw}(k)) \\
&\quad - K(1-b)\Delta\bar{T}(k), \\
\Delta\bar{T}(k) &= \bar{x}(k), \\
\Delta\hat{T}(k) &= \bar{x}_n(k), \\
\bar{y}_{aw}(k) &= \bar{x}_{aw}(k), \\
\bar{u}(k) &= (K_p + K)\left(\Delta T_R(k) - \Delta\hat{T}(k)\right) \\
&\quad + \bar{x}_i(k), \tag{13}
\end{aligned}$$

where all gains, signals, states, and parameters are defined similarly to those in (12) except with upper bars. Let us denote the asymptotic limits of $\Delta T_R(k)$, $\bar{y}_{aw}(k)$, $\Delta\bar{T}(k)$, $\bar{x}(k)$, $\bar{x}_{aw}(k)$, $\bar{x}_i(k)$, and $\bar{u}(k)$ for $k \rightarrow \infty$ by T_R , \bar{y}_{aw} , $\Delta\bar{T}$, \bar{x}_p , \bar{x}_{aw} , \bar{x}_i , and \bar{u} , respectively. Accordingly, the steady state of the system in Figure 7 is defined by

$$\begin{aligned}
\bar{x} &= \Phi\bar{x} + \Gamma h_\alpha^\beta(\bar{u}; \kappa\sigma_n), \\
\bar{x}_n &= \Phi\bar{x}_n + \Gamma h_\alpha^\beta(\bar{u}; \kappa\sigma_n), \\
\bar{x}_{aw} &= \Phi\bar{x}_{aw} + \Gamma(\bar{u} - h_\alpha^\beta(\bar{u}; \kappa\sigma_n)), \\
\bar{x}_i &= \bar{x}_i + K(1-b)(\Delta T_R - \bar{y}_{aw} - \Delta\bar{T}), \\
\Delta\bar{T} &= \bar{x}, \\
\Delta\hat{T} &= \bar{x}_n, \\
\bar{y}_{aw} &= \bar{x}_{aw}, \\
\bar{u} &= (K_p + K)(\Delta T_R - \bar{y}_{aw} - \Delta\bar{T}) + \bar{x}_i, \tag{14}
\end{aligned}$$

Replacing \bar{x}_{aw} in the seventh equation in (14) with the solution obtained from the third equation in (14) yields

$$\bar{y}_{aw} = (1 - \Phi)^{-1} \Gamma(\bar{u} - h_\alpha^\beta(\bar{u}(k); \kappa\sigma_n)). \tag{15}$$

Here, $(1 - \Phi)^{-1} \Gamma$ is the same as $G(1)$. Therefore, from the fourth equation in (14) and equation in (15), the following equation must be satisfied:

$$\begin{aligned}
K(1-b)(T_R - \Delta\bar{T}) \\
= K(1-b)G(1)(\bar{u} - h_\alpha^\beta(\bar{u}(k); \kappa\sigma_n)) \tag{16}
\end{aligned}$$

Accordingly, we obtain

$$\begin{aligned}
\bar{e} &= T_R - \Delta\bar{T} \\
&= G(1)(\bar{u} - h_\alpha^\beta(\bar{u}(k); \kappa\sigma_n)). \tag{17}
\end{aligned}$$

REFERENCES

- [1] S. Wang and R. Bettati, "Reactive speed control in temperature constrained real-time systems," *Real Time Syst.*, vol. 39, nos. 1–3, pp. 73–95, Aug. 2013.
- [2] S. Wang and R. Bettati, "Delay analysis in temperature-constrained hard real-time systems with general task arrivals," in *Proc. 27th IEEE Int. Real-Time Syst. Symp. (RTSS)*, Dec. 2006, pp. 323–334.
- [3] 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, Autom. Test Eur.*, 2005, pp. 898–899.
- [4] J. Yue, T. Zhang, Y. Liu, B. Quan, and C. Tianzhou, "Thermal-aware feedback control scheduling for soft real-time systems," in *Proc. IEEE 14th Int. Conf. High Perform. Comput. Commun. IEEE 9th Int. Conf. Embedded Softw. Syst.*, Jun. 2012, pp. 1479–1486.
- [5] 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. IEEE Real Time Embed. Technol. Appl. Symp. (RTAS)*, Apr. 2012, pp. 67–76.
- [6] Y. Fu, N. Kottenstette, Y. Chen, C. Lu, X. D. Koutsoukos, and H. Wang, "Feedback thermal control for real-time systems," in *Proc. 16th IEEE Real-Time Embedded Technol. Appl. Symp.*, Apr. 2010, pp. 111–120.
- [7] D. Kim, K.-J. Park, Y. Eun, S. H. Son, and C. Lu, "When thermal control meets sensor noise: Analysis of noise-induced temperature error," in *Proc. 21st IEEE Real-Time Embedded Technol. Appl. Symp.*, Apr. 2015, pp. 98–107.
- [8] E. Rotem, J. Hermerding, A. Cohen, and H. Cain, "Temperature measurement in the intel (R) coretm duo processor," in *Proc. Int. Workshop Thermal Invest. Syst.*, Sep. 2006, pp. 1–5.
- [9] J. Long, S. O. Memik, G. Memik, and R. Mukherjee, "Thermal monitoring mechanisms for chip multiprocessors," *ACM Trans. Archit. Code Optim.*, vol. 5, no. 2, pp. 1–33, Aug. 2008.
- [10] Y. Eun and E. S. Hamby, "Noise induced loss of tracking in systems with saturating actuators and antiwindup," *J. Dyn. Syst., Meas., Control*, vol. 136, no. 5, Sep. 2014, Art. no. 054501.
- [11] P. T. Kabamba, S. M. Meerkov, and H. R. Ossareh, "Stochastic linearisation approach to performance analysis of feedback systems with asymmetric nonlinear actuators and sensors," *Int. J. Control*, vol. 88, no. 1, pp. 65–79, Jan. 2015.
- [12] J. Lee and Y. Eun, "Analysis of noise-induced tracking loss in PI controlled systems with anti-windup," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 5461–5466.
- [13] J.-W. Choi and S.-K. Sul, "Inverter output voltage synthesis using novel dead time compensation," *IEEE Trans. Power Electron.*, vol. 11, no. 2, pp. 221–227, Mar. 1996.
- [14] A. C. Satici, H. Poonawala, and M. W. Spong, "Robust optimal control of quadrotor UAVs," *IEEE Access*, vol. 1, pp. 79–93, 2013.
- [15] Y. Li, D. Wang, W. Han, S. Tan, and X. Guo, "Performance improvement of Quasi-Type-1 PLL by using a complex notch filter," *IEEE Access*, vol. 4, pp. 6272–6282, 2016.
- [16] M.-H. Park and K.-S. Kim, "Chattering reduction in the position control of induction motor using the sliding mode," *IEEE Trans. Power Electron.*, vol. 6, no. 3, pp. 317–325, Jul. 1991.
- [17] A. V. Skorokhod, *Asymptotic Methods in the Theory of Stochastic Differential Equations*. Providence, RI, USA: AMS, 2009.
- [18] A. P. Ferreira, D. Mosse, and J. C. Oh, "Thermal faults modeling using a RC model with an application to Web farms," in *Proc. 19th Euromicro Conf. Real-Time Syst. (ECRTS)*, Jul. 2007, pp. 113–124.
- [19] K. Skadron, T. Abdelzaher, and M. R. Stan, "Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management," in *Proc. 8th Int. Symp. High Perform. Comput. Archit.*, 2002, pp. 17–28.
- [20] Y. Zhang and A. Srivastava, "Accurate temperature estimation using noisy thermal sensors," in *Proc. 46th Annu. Design Autom. Conf. (DAC)*, 2009, pp. 472–477.
- [21] K. J. Astrom and L. Rundqwist, "Integrator windup and how to avoid it," in *Proc. Amer. Control Conf.*, Jun. 1989, pp. 1693–1698.
- [22] A. Scottedward Hodel and C. E. Hall, "Variable-structure PID control to prevent integrator windup," *IEEE Trans. Ind. Electron.*, vol. 48, no. 2, pp. 442–451, Apr. 2001.
- [23] N. Bottrell and T. C. Green, "Comparison of current-limiting strategies during fault ride-through of inverters to prevent latch-up and wind-up," *IEEE Trans. Power Electron.*, vol. 29, no. 7, pp. 3786–3797, Jul. 2014.
- [24] A. D. Paquette and D. M. Divan, "Virtual impedance current limiting for inverters in microgrids with synchronous generators," *IEEE Trans. Ind. Appl.*, vol. 51, no. 2, pp. 1630–1638, Mar. 2015.
- [25] Y. Wang, S. Jiang, B. Chen, and H. Wu, "Trajectory tracking control of underwater vehicle-manipulator system using discrete time delay estimation," *IEEE Access*, vol. 5, pp. 7435–7443, 2017.
- [26] H.-B. Shin and J.-G. Park, "Anti-windup PID controller with integral state predictor for variable-speed motor drives," *IEEE Trans. Ind. Electron.*, vol. 59, no. 3, pp. 1509–1516, Mar. 2012.
- [27] *I7-800 and i5-700 Desktop Processor Series and Lga1156 Socket: Thermal/Mechanical Specifications and Design Guidelines*, Intel, Santa Clara, CA, USA, Sep. 2009.
- [28] H. Yue and H. Wang, "Minimum entropy control of closed-loop tracking errors for dynamic stochastic systems," *IEEE Trans. Autom. Control*, vol. 48, no. 1, pp. 118–122, Jan. 2003.

- [29] Y. Zhou, Q. Zhang, H. Wang, P. Zhou, and T. Chai, "EKF-based enhanced performance controller design for nonlinear stochastic systems," *IEEE Trans. Autom. Control*, vol. 63, no. 4, pp. 1155–1162, Apr. 2018.
- [30] D. Kim, J. Lee, K.-J. Park, Y. Eun, S. H. Son, and C. Lu, "Analysis and elimination of noise-induced temperature error in processor thermal control," *Real-Time Syst.*, vol. 56, no. 1, pp. 1–27, Jan. 2020.
- [31] Thomas Soderstrom. *Core i7-870 Overclocking And Fixing Blown P55-Based Boards*. Accessed: Nov. 3, 2009. [Online]. Available: <https://www.tomshardware.com/reviews/p55-motherboard-overclock,2460.html>
- [32] M. Wolf, *Computers as Components: Principles of Embedded Computing System Design*. Amsterdam, The Netherlands: Elsevier, 2012.



actuators, and resilient cyber-physical systems.

JUSEUNG LEE received the B.S. degree in electrical engineering from Chungbuk National University, South Korea, in 2013, and the M.S. degree in information and communication engineering from the Daegu Gyeongbuk Institute of Science and Technology (DGIST), South Korea, in 2015, where he is currently pursuing the Ph.D. degree in information and communication engineering. His research interests include quasi-linear control systems, control system with nonlinear sensor and actuators, and resilient cyber-physical systems.



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 systems, communication networks, and real-time systems.



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



YONGSOON EUN (Senior Member, IEEE) 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, South 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 technologies in the xerographic marking process and production inkjet printers. Since 2012, he has been with DGIST, South Korea. He is currently a Professor with the Department of Information and Communication Engineering and also the Director of the DGIST Resilient Cyber Physical Systems Research Center. His research interests include control systems with nonlinear sensors and actuators, control of quadrotors, communication networks, industry 4.0 production systems, railroad vehicle platooning, and resilient cyber-physical systems.

• • •