ABSTRACT
Many DTM schemes rely heavily on the accurate knowledge of the chip’s dynamic thermal state to make optimal performance/temperature trade-off decisions. This information is typically generated using a combination of thermal sensor inputs and various estimation schemes such as Kalman filter. A basic assumption used by such schemes is that the statistical characteristics of the power consumption do not change. This is problematic since such characteristics are heavily application dependent. In this paper, we first present autonomous schemes for detecting the change in the statistical characteristics of power and then propose adaptive schemes for capturing such new statistical parameters dynamically. This could enable accurate temperature estimation during runtime given dynamically changing power statistical states. Our schemes use a combination of hypothesis testing and residual whitening methods and can improve the accuracy by 67% as compared to the traditional non-adaptive schemes.

Categories and Subject Descriptors
B.7.2 [Integrated Circuits]: Design Aids
General Terms
Algorithms, Design, Reliability
Keywords
Temperature tracking, Sensor, Power characteristics, Adaptive, Statistical

1. INTRODUCTION
Most of today’s high performance multi-core processors suffer from the heavy power/thermal stress. The operating frequency of each processor core is fundamentally limited due to reliability issues caused by their operating temperature. To help alleviate the problem many dynamic thermal management (DTM) techniques are developed to maintain the chip temperature within reasonable range. Such DTM techniques [7, 3, 1] rely on online information of the chip’s thermal state estimated using thermal sensors. Thermal sensors have their own power and area overhead and therefore cannot be placed everywhere. Sensors are also subject to fabrication variability and hence are prone to noise. Estimating the entire thermal profile of the chip at runtime using only a few noisy thermal sensors has begun to attract significant attention. For example, Cochran et al. used spectral methods [2] to estimate the thermal profile of the chip. Zhang et al. introduced methods [10] from estimation theory to estimate the thermal profile by exploiting the logical/spatial correlation in temperature. Sharifi et al. introduced Kalman filter based method [6] to track the dynamic system temperature change. Jung et al. used a similar technique to come up with thermal estimates and generate alerts for the potential hot spots [5]. All these methods are capable of estimating the thermal profile of the chip based on limited sensor observations. One of the most promising approaches is based on the theory of Kalman filtering [6, 5] which is capable of exploiting the statistical properties of power consumption along with sensor observations to estimate the temperatures at all chip locations during runtime.

If the statistical characteristics of the power dissipation profile does not change in time, Kalman filter based approach can generate optimal thermal estimates using sensor observations [4]. However, in reality, not only the power profile is highly random (due to unpredictable workloads, variability in data, scheduling policy and etc), but its statistical characteristics (mean, variance etc) vary from time to time as well due to the changing workload characteristics and the specific setting of dynamic frequency/voltage scaling at the time. In such a scenario, since the traditional Kalman filtering based approach does not incorporate the change in statistical characteristics of the power, it can not produce satisfactory estimation accuracy. In this paper, we investigate the problem of adaptive temperature tracking at runtime by considering the dynamic changes in the statistical characteristics of the power profile. Using Kalman filter as the core estimation engine, we develop schemes for autonomously detecting changes in statistical characteristics of power and adaptive techniques for estimating the new statistical parameters. Such awareness of the system power state can be incorporated into Kalman filter to generate accurate thermal tracking results. We develop two schemes in this paper. One is based on the residual whitening of the error autocorrelation and the other is based on hypothesis testing. The first one is more suitable for the situation where the system is relatively stable (the statistical property in power does not change too quickly) whereas the second scheme is designed for the case where the system switches among different applications very frequently. Experimental results show that our adaptive method is capable of achieving 67% accuracy improvement on average over the non-adaptive Kalman filter based approaches which do not autonomously detect the system power state change. The implementation overhead is very small. For the residual whitening based method only a set of linear operations are involved. For the hypothesis testing based approach it can be implemented using table lookup strategy which means most of the overhead can be hidden by carrying out the computation off-line.

2. PRELIMINARY
The thermal profile of a chip can be highly dynamic and random due to unpredictable workload characteristics, environmental variability (supply voltage fluctuation and ambient temperature change etc.), and a wide range of heat sink con-
2.1 System Dynamics

![Figure 1: Equivalent thermal-RC model of the chip with on-chip thermal sensors](image)

To model the thermal dynamics of the chip system we use the popular thermal-RC model as shown in figure 1 (see [8, 7]). The nodes in this circuit represent various chip locations which are of thermal interests. In this model, the temperature $T_i$ at each node $i$ is represented by a voltage and the power generated at each node locally is represented by an input current. Each node $i$ has an associated thermal capacitance $C_i$. Between any neighboring nodes $i$ and $j$ there is an equivalent thermal resistance $R_{ij}$ (or conductance $G_{ij} = 1/R_{ij}$). Given a power dissipation profile of the chip, the dynamic thermal behavior can be captured by the following differential equation [8, 7]:

$$ \sum_{j \in N_i} G_{ij}(T_j(t) - T_i(t)) + C_i \frac{dT_i(t)}{dt} - P_i(t) = 0 $$

where $P_i(t)$ is the power dissipated at node $i$ at time $t$ and $N_i$ is the set of all neighbors of such a node. This differential equation can be approximated by a difference equation ([8, 6]) which can be written in its vector form as follows:

$$ T[n] = AT[n-1] + BP[n-1] $$

Here $T[n]$ and $P[n]$ are vectors representing the temperature and power values for all nodes at time $n$. The coefficients matrices $A$ and $B$ depend upon the parameters of the thermal-RC circuit as well as the time interval between consecutive discrete time steps. Essentially this difference equation captures how the current temperature $T[n]$ depends on the temperature history $T[n-1]$ and the power dissipation $P[n-1]$.

Due to unpredictability of the power dissipation (vector $P[n]$ is unknown until runtime) and random device & environmental characteristics, the exact value of $T[n]$ at runtime can be very hard to predict. This has motivated the designers to place sensors at various chip locations to provide online thermal input. Now let us assume we have a set of on-chip thermal sensors (see figure 1) which can constantly monitor the temperatures at their respective nodes and provide us an observation vector $S[n]$. Essentially $S[n]$ reports the value of a subset of $T[n]$.

$$ S[n] = HT[n] + v[n] $$

where $H$ is simply a transformation matrix determined by the sensor placement. It has dimension $s \times m$ if vectors $S[n]$ and $T[n]$ each has $s$ and $m$ elements respectively. It is noteworthy that sensors can be quite noisy due to fabrication and environmental variabilities. $v[n]$ is a Gaussian random vector with zero mean that represents the sensor noise [6]. Sensors come with a power/area cost as well. This means their number and placement are highly constrained. The problem of estimating the entire thermal profile at any time $n$ (vector $T[n]$) based on only a few sensor observations ($S[n]$) is rather complex.

Recently many techniques has been proposed to solve the above problem [2, 10, 6, 5]. Among these techniques Kalman filter based method is a very promising research direction; it is based on the linear model shown in equation (2) and it is capable of generating accurate thermal estimates for all chip locations based on limited sensor observations. It also exploits the statistical characteristics of the system very effectively which enables high estimation accuracy. In the next section we briefly introduce the fundamentals of Kalman filter.

2.2 Kalman Filter Based Thermal Tracking

Typically, the power dissipation profile $P[n]$ can be viewed as a random quantity which depends on the statistical characteristics of application, data and task scheduling etc. Assuming we know the mean $\mu_p$ and variance $Q$ of this random vector $P[n]$ (this could be generated using extensive simulation or experiments), we would like to estimate the temperature vector $T[n]$ given the sensor observations $S[n]$. When $P[n]$ and sensor noise $v[n]$ are Gaussian, a Kalman filter based approach generates optimal estimates [4]. In Kalman filter two sequential steps are carried out repetitively: predict & correct. In the predict stage the filter tries to use the mean power dissipation $\mu_p$ and the temperature history to predict the temperature in the future. As soon as a new sensor observation becomes available, the filter updates its prediction in the correct stage using the new sensor input. These two stages are carried out iteratively as the time progresses, thus continuously tracking the dynamic temperature change of the system. The Kalman filtering process can be described by equations (4) to (8) below:

**predict:**

$$ T[n|n-1] = AT[n-1] + BP[n-1] $$

**correct:**

$$ T[n|n] = T[n|n-1] + K[n](S[n] - HT[n|n-1]) $$

where $H$ is the covariance matrix of the sensor noise vector $v[n]$, $C[n|n-1]$ and $C[n|n]$ are the error covariance matrices associated with $T[n|n-1]$ and $T[n|n]$ respectively. $I$ is the identity matrix. $K[n]$ is the Kalman gain which is chosen to minimize the expected error of our thermal estimate $T[n|n]$ at each time step. The Kalman filter works in the following way: first equation (4) is used to generate the a prediction ($T[n|n-1]$) for the temperature at time $n$ based on the temperature history up to time $n-1$. Then equation (6) is used to adjust the prediction ($T[n|n-1]$) based on the new sensor measurement $S[n]$ which comes in at time $n$. The filter also updates the error covariance matrices $C[n|n-1]$ and $C[n|n]$ associated with $T[n|n-1]$ and $T[n|n]$. This helps us to keep track of the potential error in our thermal estimates. Such error covariance matrices are computed based on the covariance matrix of power ($Q$) and sensor noise ($R$).

Note that usually $R$ (the covariance of sensor noise) does not change for a relatively long period of time. It is also noteworthy that if the system is only running a specific set of applications the statistical characteristics of $P$ stay relatively stable as well. In such a scenario the Kalman filter will quickly stabilize which means $C[n|n-1]$, $C[n|n]$ and $K[n]$ will all converge to static values. This is called the steady-state of Kalman filter. During the steady state, even though our thermal estimates may change time to time, the error covariance $C[n|n]$ stays the same. This is a desirable property since we do not want the estimation error to increase beyond an acceptable bound.
3. PROBLEM DEFINITION AND CHALLENGES

As highlighted earlier, if the statistical characteristics (mean \( \mu \) and variance \( Q \)) of the power dissipation profile \( P[n] \) does not change, Kalman filtering based approach is capable of generating optimal thermal estimates using sensor observations. However in reality, not only \( P[n] \) is highly unpredictable, the statistical characteristics varies from time to time as well due to the change in application characteristics, data being processed at the time and dynamic power management (DPM) policies etc. In such a scenario, the filtering approach presented earlier breaks down. Let us assume we have \( K \) different statistical power states, each of which essentially represents the behavior of a certain cluster of similar applications etc. The probabilistic distribution of \( P[n] \) in state \( k \) can be modeled as Gaussian with mean \( \mu_k \) and variance \( Q_k \) (the power statistical state \( k \) is hereafter referred to as a hypothesis \( H_k \)). Now given a certain sensor observation vector \( S[n] \) at time \( n \), we would like to predict whether the underlying power state has changed and what the new state might be. Once we know the new power state we can apply the correct statistical information (\( \mu_k \) and \( Q_k \)) in equations 4 to 8, hence updating the Kalman filter to adapt to the new system state. Our problem can be formulated below:

Given: \( K \) hypotheses of the system's statistical power state and the prior probability of each hypothesis:

\[
\begin{align*}
H_1 &\sim \mathcal{N}(\mu_{k_1}, Q_1) : p_1 \\
H_2 &\sim \mathcal{N}(\mu_{k_2}, Q_2) : p_2 \\
H_K &\sim \mathcal{N}(\mu_{k_K}, Q_K) : p_K
\end{align*}
\]

1) Detect if any change in power state has occurred and 2) Predict the new power state and update Kalman filter.

The statistical characteristics of each potential power state could be captured by simulating or experimenting with all potential applications sets (integer vs floating point, scientific vs multimedia and etc.) and various power management settings. The prior probability \( p_i \) for each state \( H_i \) could be computed based on the our experience of the typical workload distribution. In this paper we assume all such statistical information pertaining to each hypothesis is given. We focus on developing an adaptive and autonomous scheme for detecting and estimating the new power state of the system during runtime. Earlier works on this topic include the ARMA model [3], but their approach assumes a linear model for updating \( H_i \) and estimating the new power state of the system during runtime. Our system's statistical characteristics stay the same within each power state it can be shown that the error covariance matrices \( \{C[n|n-1]\} \) and Kalman gain \( K[n] \) will stabilize rather quickly and converge to fixed values which we denote by \( C_s \) and \( K_s \). For all \( K \) power states, we can easily compute the steady state \( C_s \) and \( K_s \) using equations 4 to 8 (see [4] for details). This notion of stability is an important property that we will exploit in our methods. Now let us suppose the system has switched the power state and converged to the steady state. In such a scenario, to explain our method let us first introduce the concept of the residual process \( e[n] \):

\[
\]

where \( x[n] = (T[n|n-1] - T_{real}[n]) \) and it represent the difference in the predicted and the real temperature vectors. The autocorrelation function of this residual process \( e[n] \) is defined as \( ac_i \equiv E(e[n] \cdot e[n-i]^T) \). Basically this is the autocorrelation between the error at time \( n \) and \( n-i \). Based on equation 9 it has the following form:

\[
ac_i = \begin{cases} 
HE(e[n] \cdot e[n-i]^T)H^T - HE(e[n] \cdot v[n-i]^T) & i > 0 \\
HC_sH^T + R & i = 0
\end{cases}
\]

(11)

Note that we use \( C_s \) instead of \( C[n|n-1] \) since we assume the system has reached the steady state. To obtain the analytical form of autocorrelation function \( ac_i \) we can first derive the recursive equation for \( x[n] \) based on equations 2, 3, 4 and 6 (for details see [4]):

\[
x[n] = A(I - K_sH)x[n-1] + AK_s(e[n-1] - B[P[n-1] - \mu^T])
\]

(12)

Once again we use steady state \( K_s \) as parameter. Carrying equation 12 i steps back we have the relationship between \( x[n] \) and \( x[n-i] \). Then the following equation can be easily computed:

\[
E(x[n] \cdot x[n-i]^T) = [A(I - K_sH)]^iC_s
\]

(13)

\[
E(x[n] \cdot e[n-i]^T) = [A(I - K_sH)]^{i-1}AK_sR
\]

(14)

Substituting the above two equations into 11, the autocorrelation of the residual process \( e[n] \) is given by:

\[
ac_i = \begin{cases} 
H[A(I - K_sH)]^{i-1}A[C_sH^T - K_sac_0] & i > 0 \\
HC_sH^T + R & i = 0
\end{cases}
\]

(15)

If we know the current power state we know its steady state \( K_s \) and \( C_s \) as explained earlier. The steady state Kalman gain will follow the filter equation 7 which capture its relationship with other parameters \( (C_s \) and etc.). Now if we reliable and accurate thermal estimates. Note that the power consumption in different states can have wildly different values (for example: full-speed vs sleep state).

3. Speed vs accuracy - The methods for autonomous detection and adaptive estimation need to be applicable during runtime. They should be aware of the rate at which system switches states and balance the accuracy and detection speed accordingly.

Next we present two schemes for adaptive and autonomous thermal tracking when the statistical characteristics of the application change with time.
plug in steady state Kalman gain $K_i$ in 15 we get $ac_0 = 0$ for $\forall i > 0$. Hence if we know the correct $\mu^p$ and $Q$ and therefore know the correct steady state gain $K_i$, then equation 15 should evaluate to 0 for $\forall i > 0$. This is called the whitening of the residual process. Now, we can compute the autocorrelation using the most recent $N$ sensor data samples as follows:

$$\hat{ac}_i = \frac{1}{N} \sum_{n=1}^{N} [e[n] \cdot e[n - i]^T]$$

(16)

Basically, we evaluate error between observation and prediction and estimate the autocorrelation. Now the prediction was performed using Kalman filtering equations which are based on our belief of what the current power state is. If the belief is incorrect then the computed autocorrelation using equation 16 will not evaluate to zero. Thus a good way to detect the switch of the system power state is to calculate the autocorrelation using equation 16 based on real sensor data and then check if the result is close to zero.

Based on the above theory of residual whitening we can design our adaptive Kalman filter in the following way: The filter stores the most recent $N$ data samples and keeps updating the autocorrelation function $ac_i$ based on equation 16. As soon as the value of $ac_i$ exceeds a certain threshold $ac_s$ the filter knows that a power state change has happened and hence can react accordingly. This approach does not require any external interference and detects the change in power state autonomously.

### 4.2 Adaptive Tracking Algorithms

Once our adaptive filter detects the power state change, there are two things that need to be done: 1) the new power state needs to be predicted and 2) the filter parameters $\{\mu^p$ and $Q\}$ need to be updated to their correct values for the new state. To predict the new power state we could do the following: for each potential state/hypothesis $k$, we compute the autocorrelation function $ac_k$ of the residual process assuming hypothesis $k$ is true. Then we could simply choose state $q$ as our predicted state as follows:

$$q = \arg\min_k \{ac_k - 0\}$$

(17)

In the simplest case $i$ can be chosen as 1. There is one more complication though. If we only know that a switch in power state has occurred but do not know where exactly it happened in time. We would not be able to compute the correct values of the residual process $e^k[n]$ for each hypothesis $k$, let alone the autocorrelation function $ac_k$. On the other hand if we can detect the exact switching time $t_{sw}$, for the power state change we can rewind the Kalman filter to this switching point and use ($\mu^p_k$, $Q_k$) (associate with each hypothesis $k$) to carry out Kalman filtering starting from time $t_{sw}$. Then we can easily obtain the residual process $e^k[n]$ and autocorrelation function $ac_k$. Once we have all $ac_k$’s we can use equation 17 to predict the new power state. Essentially this scheme chooses the state that minimizes $ac_k$ (the one closest to zero) as our predicted new state.

Next we give a divide-and-conquer algorithm for detecting the switching point of the power state change (see algorithm 1). Given the most recent $N$ data samples (within which the state switching has been detected) and the previous system state $l$ before the switch happened, the algorithm predicts the switching point to fall within the first half or the second half of the total data samples based on the autocorrelation computed using each half respectively. Then we recursively divide this sub-range in half till the switching point is predicted within a reasonable accuracy (minlength in the algorithm). Algorithm 2 gives the overall adaptive thermal tracking procedure. Note that for these algorithms we assume Kalman filter is running underneath and generating thermal estimates ($T_i[n-1]$ and $T_i[n]$, etc) at each time step $n$. In addition the filter always keeps track of the most recent $N$ data samples.

**Algorithm 1**

**Input:** Sequential data samples $\{S[n]\}$ and $\{T_i[n-1]\}$ for $n = t_i$ to $t_f$. Acceptable detection error minlength

**Output:** The approximate switching point $t_{sw}$ as output

1. $t_m \leftarrow \lfloor \frac{t_i + t_f}{2} \rfloor$ // compute the middle point
2. /* minlength: acceptable detection accuracy */
3. if $t_f - t_i < minlength$ then
4. return $t_{sw} = t_f$
5. else
6. for $n = t_i$ to $t_m$ do
7. compute residual $e[n] = S[n] - HT_i[n-1]$
8. estimate $\hat{ac}_n = \sum_{i=1}^{minlength} [e[n] \cdot e[n - i]^T]$ for the first half data samples
9. if $ac_s^2 > threshold$ then
10. /* $ac_s^2 \approx 0$: the switching point falls in $[t_i, t_m]$ */
11. return $sw_{detect}(t_i, t_m)$
12. else
13. /* $ac_s^2 \approx 0$: the switching point falls in $[t_m + 1, t_f]$ */
14. return $sw_{detect}(t_{m+1}, t_f)$
15. end if
16. end if

**Algorithm 2**

**Adaptive Tracking Algorithm Based on Residual whitening**

**Input:** The sequential sensor inputs $S[n]$ and the potential power statistics ($\mu^p_k$, $Q_k$) for $k = 1$ to $K$

**Output:** The adaptive Kalman filter output

1. initialize $\mu^p \leftarrow \mu^p$, $Q \leftarrow Q_k$ with any $k$
2. while (1) do
3. advance Kalman filter to the next time step ($n \leftarrow n + 1$)
4. if ($n\%N = 0$) then
5. compute $e[i]$ for the most recent $N$ samples
6. compute $ac_i = \frac{1}{N} \sum_{n=1}^{N} [e[i] \cdot e[i-1]^T]$
7. if $|ac_i| < threshold$ then
8. continue /* no power state change */
9. else
10. $t_{sw} = sw_{detect}(n - N + 1, n)$ // switching time
11. /* predict the new system state below */
12. for $k = 1$ to $K$ do
13. $\mu^p \leftarrow \mu^p_k$, $Q \leftarrow Q_k$
14. compute $ac_k$, assuming state $k$ is true
15. end for
16. $q = \arg\min_k \{ac_k - 0\}$
17. update filter parameters: $\mu^p \leftarrow \mu^p_q$, $Q \leftarrow Q_q$
18. end if
19. end while

### 5. ADAPTIVE TRACKING BASED ON HYPOTHESIS TESTING

The previous approach was based on computing the autocorrelation value for the residue process for all the potential power states and choosing the one which was closest to 0.

In this section we present an alternative hypothesis testing based approach. As described in section 3 we suppose each power state $H_k$ of the $K$ potential states (which we also call hypotheses) has a probability $p_k$ of occurrence. In a hypothesis testing framework, we always check the probability of each underlying power state using the most recent sensor observations to see if the previous power state is obsolete. Essentially we would like to choose the state of the highest probability of occurrence given the current set of sensor observations. Let us suppose we make a sensor observation at time $n$: $S[n]$. The posterior probability for each hypothesis $H_k$ is defined as $prob(H_k | S[n])$. In this framework, we would like to choose the power state or hypothesis with maximum posterior probability.

$$H_p = \delta(S[n]) = \arg\max_{h_k \in \{1, N\}} \{prob(H_k | S[n])\}$$

(18)

Here $\delta(S[n])$ is our decision rule and it is a function of the sensor observation $S[n]$ only. Given a certain $S[n]$, we evaluate $prob(H_k | S[n])$ for each potential $H_k$ and then choose the one with the highest probability. The problem is really how
we can obtain the value of \( \prob[H_k|S[n]] \). Note that according to Bayes’ theorem we have the following relationship:

\[
\prob[H_k|S[n]] = \frac{\prob(H_k, S[n])}{\prob(S[n])} = \frac{\prob(S[n]|H_k) \times \prob(H_k)}{\prob(S[n])} \tag{19}
\]

where \( \prob(H_k) \) is the prior probability of \( H_k \) (\( \prob(H_k) = p_k \)). Note that the denominator in the above equation stays the same for each different hypothesis \( H_k \). This means we can focus on the numerator only and simplify our decision rule to the following:

\[
H_p = \delta(S[n]) = \arg \max_k \left( \prob(S[n]|H_k) \times p_k \right) \tag{20}
\]

where \( \delta(S[n]) = \{ H_1, \ldots, H_K \} \). Now we can use the following process to compute the value of \( \prob(S[n]|H_k) \). Let us assume that until time \( n - 1 \) we knew exactly which power state we were in. Hence we know \( T[n-1|n-1] \) and \( C[n-1|n-1] \) accurately. At time \( n \), we detect a change in power state and have an associated new sensor sample \( S[n] \). Since the power state has changed we do not know \( T[n|n-1] \) and \( C[n|n-1] \) accurately. For each hypothesis \( k \), we can use equations 4 to 8 in time step \( n \) to compute new potential \( T_k[n|n-1] \) and \( C_k[n|n-1] \) for each hypothesis \( k \). Now let the error \( x_k[n] = T[n] - T_k[n|n-1] \) for each hypothesis \( k \). Thus \( C_k[n|n-1] \) is the covariance matrix of \( x_k[n] \).

Based on equation 3 we have the following:

\[
S[n] = HT[n] + v[n] = H(x_k[n] + T_k[n|n-1]) + v[n] \tag{21}
\]

where \( v[n] \) is the sensor noise (normally distributed with zero mean and covariance \( R \)). Since \( S[n] \) is a linear combination of Gaussian random variables, it should be normally distributed as well whose mean and covariance can be derived as follows (assuming \( x_k[n] \) and \( v[n] \) are uncorrelated):

\[
\mu^S_k = E[H(x_k[n] + T_k[n|n-1]) + v[n]] = HT_k[n|n-1] \tag{22}
\]

\[
\Sigma^S_k = H\text{cov}(x_k[n])HT^T + \text{cov}(v[n]) = HC_k[n|n-1]HT^T + R \tag{23}
\]

Here \( \text{cov}(\cdot) \) represents the covariance of a certain random vector. Now that (given a certain hypothesis \( H_k \)) we know \( S[n] \) has Gaussian distribution with mean \( \mu^S_k \) and covariance \( \Sigma^S_k \), the probability \( \prob(S[n]|H_k) \) can be easily obtained based on the following multivariate Gaussian density function:

\[
p(S[n]|H_k) = \frac{1}{(2\pi)^{\frac{N}{2}}|\Sigma^S_k|^{1/2}} \exp \left\{ -\frac{1}{2} \left( S[n] - \mu^S_k \right)^T (\Sigma^S_k)^{-1} \left( S[n] - \mu^S_k \right) \right\} \tag{24}
\]

Once we have \( \prob(S[n]|H_k) \) the decision rule in equation 20 can be easily evaluated (note that \( \prob(H_k) = p_k \) is the prior probability).

Though the above described method is very accurate, its accuracy depends heavily on the amount of noise in the system and also how accurately we can determine the switching time of the power states. This is because the above scheme relies too much on \( S[n] \); the information sampled at a single time instance. It tends to make mistakes if such a \( S[n] \) is erroneous due to noise. We can improve the accuracy of this method by considering \( m \) sequential sensor observations \( \{ S[n], S[n-1], \ldots, S[n-m+1] \} \). Essentially by exploiting the information sampled at multiple time instances the noise associated with each one gets canceled out and a more accurate prediction can be achieved. In such a scenario our decision rule is similar:

\[
H_p = \delta(S[n], S[n-1], \ldots, S[n-m+1]) = \arg \max_{H_k \in \delta} \left( \prob(H_k | S[n], \ldots, S[n-m+1]) \right) \tag{25}
\]

\[
= \arg \max_{H_k \in \delta} \left( \prob(S[n], \ldots, S[n-m+1] | H_k) \times \prob(H_k) \right) \tag{26}
\]

Note equation 26 is obtained by applying the Bayes’ theorem and noting the fact that the denominator stays the same for each hypothesis \( H_k \). To evaluate this decision rule we can expand the conditional probability as follows:

\[
\prob(S[n], S[n-1], \ldots, S[n-m+1] | H_k) = \prob(S[n-m+1]|H_k) \times \prob(S[n-m+2]|S[n-m+1], H_k) \times \cdots \times \prob(S[n]|S[n-1], \ldots, S[n-m+1], H_k) \tag{27}
\]

\[
= \prod_{i:n-m+1} \prob(S[i] | T_k[i|1], C_k[i|1], H_k) \tag{28}
\]

Note that at a certain time \( i \), Kalman filter uses all previous sensor observations up to time \( i-1 \), combined with the statistical information \( H_k \sim \mathcal{N}(\mu^p_k, Q_k) \) to generate \( T_k[i|1] \) and \( C_k[i|1] \), which is then used to derive the conditional probability of \( S[i] \) (see equations 22, 23 and 24). This leads to the simplification in the last step. Note that each term in 28 can be computed in exactly the same way as we showed earlier when computing the conditional probability of a single observation \( \prob(S[i]|H_k) \). Once we know how to compute \( \prob(S[n], S[n-1], \ldots, S[n-m+1]|H_k) \), the decision rule for multiple sequential sensor observations (equation 26) can be easily obtained.

6. QUALITATIVE COMPARISON

In general the hypothesis-testing based scheme operates at a much finer granularity in time than the residual-whitening based method. The latter method usually has lower overhead; it is activated every \( N \) steps and only tries to predict the new system state if an actual change has been detected and also higher accuracy (more sensor observations are used to test the whiteness of the residual process: \( N >> m \)). However it is constrained to the case where the system stays in each state for relatively long period of time so that Kalman filter has entered steady-state. The hypothesis-testing based method has the flexibility of changing the number of sequential observations exploited (the value \( m \)). Thus it can adapt to any switching frequency of the system. However it has higher implementation overhead; the probability in equation 26 needs to be updated and decision rule needs to be evaluated at each time step. We are providing both methods here so that in practice the most appropriate scheme can be selected based on the nature of the application field (yet another level of adaptivity).

7. EXPERIMENTAL RESULTS

In this section we show some of our experimental results using both the residual whitening and the hypothesis testing based methods. To obtain the statistical characteristics (mean \( \mu^p_k \) and covariance \( Q_k \)) for each potential power state \( k \), we simulated a high performance aggressive processor with all SPEC2000 CPU benchmarks. The experimental setting for our simulation are as follows: the processor is a high performance aggressive out-of-order processor with pipeline width of 8 instructions and an instruction window of 128 instructions. Level 1 caches (both instruction cache and data cache) are 32KB 4-way set associative. All the caches in the hierarchy are using LRU replacement policy and a block size of 64 bytes. For each benchmark \( k \), we simulated a representative 250M instructions (compiled with the default parameters provided with the suite) and sampled the chip power dissipation values using uniform time intervals. We superimposed a certain amount of randomness (in line with [9]) to the power sample which essentially represents the collective effects of supply voltage fluctuation, environmental randomness and fabrication variability. Then we extract \( \mu^p_k \) and \( Q_k \) based on the sample mean and sample covariance. Essentially we are using the above simulation to capture the statistical characteristics (\( \mu^p_k, Q_k \)) of the system’s power dissipation when it is executing benchmark \( k \). Such statistical information generalized from all benchmarks are used as our underlying hypotheses of the system’s power states. Our goal.
is to autonomously detect the change of states as well as to predict the new state so that we can update the Kalman filter adaptively to track the dynamic system thermal profile.

To test the effectiveness of our adaptive approaches, we randomly choose and run a different benchmark from the SPEC2000 suite every 50 seconds and apply 1) the normal non-adaptive Kalman filter, 2) our adaptive scheme based on residual whitening (autocorrelation function is computed based on 30 data samples), 3) our adaptive scheme based on hypothesis testing (using 10 consecutive data samples) and 4) our adaptive scheme based on hypothesis testing (using a single data sample). The actual temperature is computed using the differential equation based model presented earlier. The actual temperature and the estimated temperature (for a randomly chosen chip location that does not coincide with the sensor locations) obtained for each of the above schemes are shown in figures 2, 3, 4 and 5 respectively. The average RMS error across the entire chip is compared in table 1. Although it may seem that hypothesis testing is better than residual whitening, the latter does not require us to be constantly estimating the power state and is instantiated only if we detect a power state change. Hypothesis testing approach must always make predictions to keep up its estimates and hence has more overhead.

### Table 1: RMS error (°C) comparison for all 4 settings (KF: Kalman filter)

<table>
<thead>
<tr>
<th>Settings</th>
<th>normal KF</th>
<th>adaptive KF</th>
<th>% gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residual Whitening</td>
<td>4.41</td>
<td>1.83</td>
<td>58.5%</td>
</tr>
<tr>
<td>Hypo (mult-sample)</td>
<td>4.41</td>
<td>1.21</td>
<td>72.6%</td>
</tr>
<tr>
<td>Hypo (singl-sample)</td>
<td>4.41</td>
<td>1.32</td>
<td>70.1%</td>
</tr>
</tbody>
</table>

As can be seen from the figures, our adaptive scheme successfully detected the underlying system’s power statistical characteristics change and adapted the Kalman filter to keep constant thermal tracking with high accuracy without any external interference. Table 1 shows that our scheme can give about 67% accuracy improvement on average over the normal Kalman filter method. This clearly demonstrate the effectiveness of our adaptive and autonomous thermal tracking methodologies.

![Figure 2](image2.png)

**Figure 2:** Actual temperature vs thermal estimation using normal Kalman filter

![Figure 3](image3.png)

**Figure 3:** Actual temperature vs our adaptive thermal estimates using residual whitening

8. CONCLUSION

In this paper we present an adaptive and autonomous approach for thermal estimation using sensors. We developed several adaptive algorithms and reported accuracy improvements over the traditional non-adaptive Kalman filter based methods.

9. REFERENCES


