Evaluation of Processor Health within Hierarchical Condition Monitoring System

Lenka Pavelková and Ladislav Jirsa
Department of Adaptive Systems, Institute of Information Theory and Automation, Czech Academy of Sciences, Pod Vodárenskou věží 4, Prague, Czech Republic

Keywords: CPU Usage, Processor Utilisation, Condition Monitoring, Processor Health Evaluation, Bayesian Estimation, Autoregressive Model, Probabilistic Logic, Binomial Opinion.

Abstract: This paper proposes an intelligible method for the evaluation of a condition of the central processing unit (CPU). Proposed method monitors CPU utilisation with respect to user given bounds and provides result in the form of binomial opinion that serves subsequently for the condition monitoring of an industrial system. The system in question is described by a hierarchical structure and the examined CPU belongs to the set of its basic building blocks.

1 INTRODUCTION

Nowadays, fault detection and isolation (FDI) forms an important part of control systems in engineering applications (Isermann, 2011). FDI provides an opinion whether the system is in faulty state and indicates the location and nature of a possible fault. Within an industrial plant, many possible fault sources exist, e.g., sensors, actuators, hardware components. Therefore, monitoring and processing of the system as a whole is a complex task and it results generally in a solution tailored for a particular system.

A novel dynamic hierarchical monitoring system based on probabilistic approach to fault detection is proposed in (Dedecius and Ettler, 2014). There, the industrial system of interest is decomposed into blocks representing individual physical or logical system units. For each particular block, a binomial opinion on its condition (health) has to be assessed. These individual opinions are subsequently combined to obtain the resulting opinion on a total system health.

In this contribution, we focus on an evaluation of the condition of one particular building block within above mentioned hierarchical structure, namely central processing unit (CPU).

CPU belongs to the most important part of a computer. To evaluate an opinion on its health, the CPU utilisation $\mathcal{U}[\%]$ is usually monitored that refers to a computer's usage of processing resources. An overloaded CPU can slow down a computer unbearably or a running applications can freeze. On other side, too low CPU usage could indicate that some application has been unexpectedly terminated.

Within considered industrial system, only long-lasting crossing of given bounds indicate possible troubles. Short overshoots of given boundaries are acceptable. Therefore, immediate evaluation of a current CPU usage provided e.g. by Task Manager in Windows or htop in Linux is not suitable because an alarm signalising a bad CPU condition would be given after each particular bounds crossing.

Many methods and software have been evaluated that monitor the CPU usage. Some of them serve merely for observation and possibly for prediction while others also take an action after CPU overload detection.

For example, a CPU usage prediction method based on the linear regression technique is presented in (Farahnakian et al., 2013). The proposed approach approximates the short-time future CPU utilization based on the history of usage. It is employed to predict over-loading and under-loading of CPU. In (Wang et al., 2012), Kalman filter is adapted to estimate CPU consumption from observed data. The paper (Akhtar and Sidek, 2013) proposes method for maximum CPU utilization using bandwidth allocation.

To name some software samples, CPU Utilization Monitor[1] is a program that provides a means to monitor and log CPU usage for any period of time. The

[1] http://sourceforge.net/projects/cpu-usage-log/

Pavelková L. and Jirsa L..
Evaluation of Processor Health within Hierarchical Condition Monitoring System.
DOI: 10.5220/0005578106670671
In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics* (ANNIIP-2015), pages 667-671
ISBN: 978-989-758-122-9

optionally produced log file is tab delimited and easily importable e. g. into MS Excel for further trending and analysis. A software for music production, program Reaktor[2], monitors CPU usage during its performance. When the CPU load reaches a certain value limit, Reaktor will shut down audio processing to assure that the system is not frozen by a processor overload.

Despite of highly elaborated tools in this field to monitor CPU performance, none is compatible with a *unified interface* of building blocks enabling a generic modularity.

This paper proposes such a method. The idea is to combine consistently statements on condition of logical blocks within a hierarchically composed system using the binomial opinion (Josang, 2008) as the common interface. This approach is then applicable for any logical structure provided that condition of each logical block is described by the binomial opinion.

The evaluation the CPU's condition is based on an estimation of $\mathcal{U}$ using a probabilistic autoregressive model and on the evaluation of obtained estimates by considering user given bounds on CPU usage.

The paper is organised as follows. Section 2.1 introduces basics of hierarchical monitoring system which comprises CPU as one of its basics parts. Section 2.2 presents the model of a CPU usage. The evaluation of CPU condition is described in Section 2.3. Section 2.4 provides an algorithmic summary. Experiments with simulated data are presented in Section 3. Section 4 concludes the paper.

Throughout, the transposition is marked $'$,
z^* denotes a set of z-values,
z_t is the value of z at discrete-time instant $t \in t^* = \{1,2,\ldots,T\}$, $T < \infty$,
$\hat{X}_t$ denotes the estimate of X in time t.
The symbol f denotes probability (density) function (p(d)f) distinguished by the argument names; no formal distinction is made among a random variable, its realisation and a pdf argument.

2 CPU USAGE MONITORING

The unified interface to describe a condition of a monitored unit as a block in a structure is linked with the modelling of CPU load.

[2]http://www.native-instruments.com/en/products/komplete/synths/reaktor-5/

2.1 Basics of Hierarchical Condition Monitoring

The examined CPU is a part of the above mentioned probabilistic FDI system (Dedecius and Ettler, 2014), where the investigated industrial system is decomposed into a set of interconnected individual components called basic blocks. To each particular block, a binomial opinion (1) on its condition has to be assigned. The basic blocks are interconnected using principles of the subjective logic (SL) (Josang, 2008). In this way, a single opinion on the health of the whole monitored system is obtained by combining opinions on the health of the individual basic blocks.

In SL, the representation of uncertain probabilities is based on a belief model. A subjective binomial opinion expresses a subjective belief of a particular subject about the truth of proposition including a degree of uncertainty. For $x \in \{0,1\}$, a binomial opinion about the truth of proposition $x = 1$ is the ordered quadruplet

$$\omega_x = (b,d,u,a) \tag{1}$$

where
b is the belief of x being true, i. e. $b = f(x = 1)$,
d is the belief of x being false (disbelief), i. e. $d = f(x = 0)$,
u is the uncertainty, i. e. no opinion on x being true or false,
a is the base rate (corresponds to a prior information).
These components are interpreted as corresponding probabilities. They satisfy additivity $b + d + u = 1$ and it holds $b, d, u, a \in [0,1]$.

The expected value of b is

$$E_x = b + au\,. \tag{2}$$

Here, we consider non-informative prior, i.e. $a = 0.5$.

Particularly, belief, disbelief and uncertainty represent opinion on situation that the processor load is within the requested bounds.

2.2 Model of CPU Utilisation

A CPU utilisation $\mathcal{U}_t$ is modelled by the autoregressive model, $t \in t^*$,

$$f(\mathcal{U}_t|\psi_t,\vartheta,r) \equiv \mathcal{N}_{\mathcal{U}_t}(\psi_t'\vartheta,r) \tag{3}$$
$$= \frac{1}{\sqrt{2\pi r}} \exp\left[-\frac{(\mathcal{U}_t - \psi_t'\vartheta)^2}{2r}\right]$$

where

$\mathcal{N}_{\mathcal{U}_t}(\psi_t'\vartheta,r)$ means Gaussian pdf with the expected value $\psi_t'\vartheta$ and the noise variance r,

$\psi_t = [\mathcal{U}_{t-1},\ldots,\mathcal{U}_{t-n}]$ is the regression vector consisting of n previous values of $\mathcal{U}$,

ϑ is a vector of unknown regression coefficients of the length n,

Estimation of the Gaussian model, i.e. estimation of ϑ and r, is performed using sufficient statistics V_t and ν_t which update according to

$$\begin{aligned} V_t &= V_{t-1} + \Psi_t \Psi_t', \\ \nu_t &= \nu_{t-1} + 1, \end{aligned} \tag{4}$$

where $\Psi_t = [y_t, \psi_t']'$ and V_0, ν_0 are given by the prior pdf.

Matrix V_t can be decomposed as a product $V = L'DL$, where L is a lower triangular matrix with unit diagonal and D is a diagonal matrix.

The moments of the parameters' posterior pdfs are given by

$$\mathcal{E}[\vartheta|L,D,\nu] = L_\psi^{-1} L_{d\psi} \equiv \hat{\vartheta}, \tag{5}$$

$$\mathcal{E}[r|L,D,\nu)] = \frac{D_d}{\nu - 2} \equiv \hat{r}. \tag{6}$$

where L_ψ, $L_{d\psi}$, D_d are obtained by the following decomposition of L and D

$$L = \begin{bmatrix} 1 & 0 \\ L_{d\psi} & L_\psi \end{bmatrix}, \qquad D = \begin{bmatrix} D_d & 0 \\ 0 & D_\psi \end{bmatrix}. \tag{7}$$

The point output prediction is

$$\hat{\mathcal{U}}_t = \hat{\vartheta}' \psi_t. \tag{8}$$

For computational details, see (Kárný et al., 2006).

2.3 Evaluation of CPU Condition

After estimating of CPU usage, it remains to assign the required opinion on the CPU health. To declare that CPU is health, CPU utilisation $\mathcal{U}$ has to be within interval

$$(\underline{\mathcal{U}}; \overline{\mathcal{U}}). \tag{9}$$

For the purpose of hierarchical monitoring system described in Section 2.1, the information about CPU health is required in the form of binomial opinion (1). Direct evaluation based on comparing estimates $\hat{\vartheta}$ (5) and respective point predictions $\hat{\mathcal{U}}_t$ with bounds $\underline{\mathcal{U}}$ and $\overline{\mathcal{U}}$ does not take into account the admissible occasional crossings of given bounds.

We propose the following procedure that treats these events. It uses σ confidence intervals given by boundaries

$$L_{3\sigma} = \hat{\vartheta}' \psi_t - 3\sigma, \qquad U_{3\sigma} = \hat{\vartheta}' \psi_t + 3\sigma \tag{10}$$

where $\sigma = \sqrt{\hat{r}}$ is the standard deviation, $\hat{r}$ is given by (6).

The values of b, d, u in binomial opinion (1) are obtained by evaluating of boundaries (10) with respect to user given bounds $\underline{\mathcal{U}}$ and $\overline{\mathcal{U}}$ (9) according to the Table 1. When constructing this Table, we tried to imitate a decision-making of a real technical operator who examines the estimation results personally. Note that the value of u expresses user uncertainty concerning his/her opinion.

2.4 Algorithmic Summary

Initialisation

- enter bounds $\underline{\mathcal{U}}$, $\overline{\mathcal{U}}$ (9)
- set $t = 0$ and end time T

On-line Phase

1. $t = t + 1$
2. measure data $\mathcal{U}_t$
3. update statistics ν_t, V_t (4)
4. estimate ϑ, r according to (5), (6)
5. compute boundaries $L_{3\sigma}$, $U_{3\sigma}$ (10)
6. assign b, d, u based on comparing $L_{3\sigma}$, $U_{3\sigma}$ with $\underline{\mathcal{U}}$, $\overline{\mathcal{U}}$ according to the Table 1
7. IF $t < T$, GO TO 1

3 EXPERIMENTS

Experiments with simulated data are presented to illustrate the proposed method of assigning an opinion on CPU condition. We consider bounds (9) as follows, $\underline{\mathcal{U}} = 10$, $\overline{\mathcal{U}} = 90$. CPU usage $\mathcal{U}$ is modelled by (3) with $n = 2$.

3.1 Data with Outliers

We simulate the set of $\mathcal{U}$s that meets given bounds ($\underline{\mathcal{U}}$, $\overline{\mathcal{U}}$) including two outliers. The simulated data with highlighted estimates of σ-intervals are in the upper part of Figure 1. The assigned ω (1) is in its bottom part.

3.2 Constantly Changing Data

We simulate monotonically changing $\mathcal{U}$ with a flat maximum that is greater than the upper bound $\overline{\mathcal{U}}$. The simulated data with highlighted estimates of σ-intervals are depicted in the upper part of Figure 2. The assigned ω (1) is in its bottom part. Figure 3 shows a zoomed part of Figure 2.

Table 1: Assignment of b, d, u (1) depending on $L_{3\sigma}$ and $U_{3\sigma}$ (10).

	$L_{3\sigma} \in \langle 0, \underline{\mathcal{U}})$			$L_{3\sigma} \in \langle \underline{\mathcal{U}}; \overline{\mathcal{U}} \rangle$		$L_{3\sigma} \in (\overline{\mathcal{U}}; 100\rangle$
	$U_{3\sigma} \in \langle 0, \underline{\mathcal{U}})$	$U_{3\sigma} \in \langle \underline{\mathcal{U}}; \overline{\mathcal{U}} \rangle$	$U_{3\sigma} \in (\overline{\mathcal{U}}; 100\rangle$	$U_{3\sigma} \in \langle \underline{\mathcal{U}}; \overline{\mathcal{U}} \rangle$	$U_{3\sigma} \in (\overline{\mathcal{U}}; 100\rangle$	$U_{3\sigma} \in (\overline{\mathcal{U}}; 100\rangle$
b	0	0.5	0	1	0.5	0
d	1	0	0	0	0	1
u	0	0.5	1	0	0.5	0

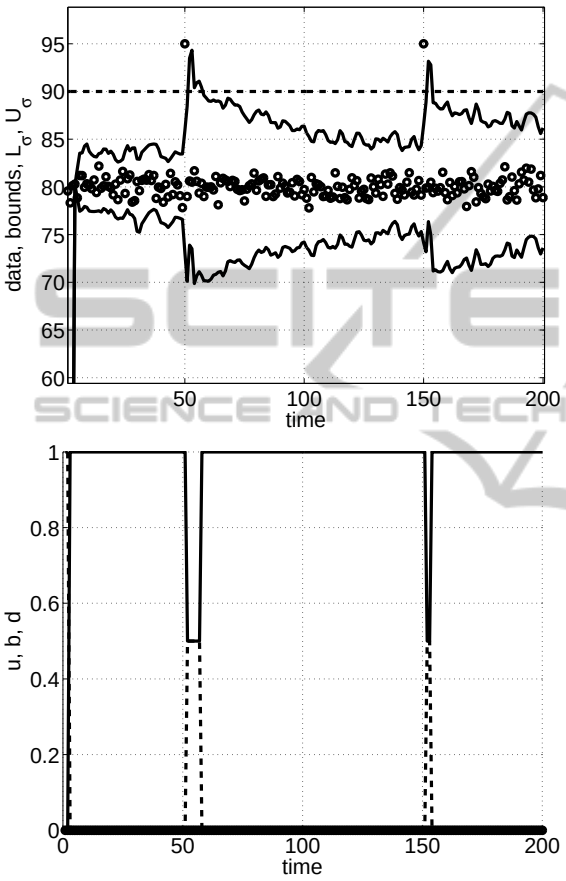

Figure 1: **Simulated data with outliers:** On top: CPU usage (circles) with highlighted boundary $\overline{\mathcal{U}}$ (9) (dashed line) and $L_{3\sigma}$, $U_{3\sigma}$ (10) (full line). Below: The course of b (thin line), d (thick line), u (dashed line) in ω (1) according to Table 1.

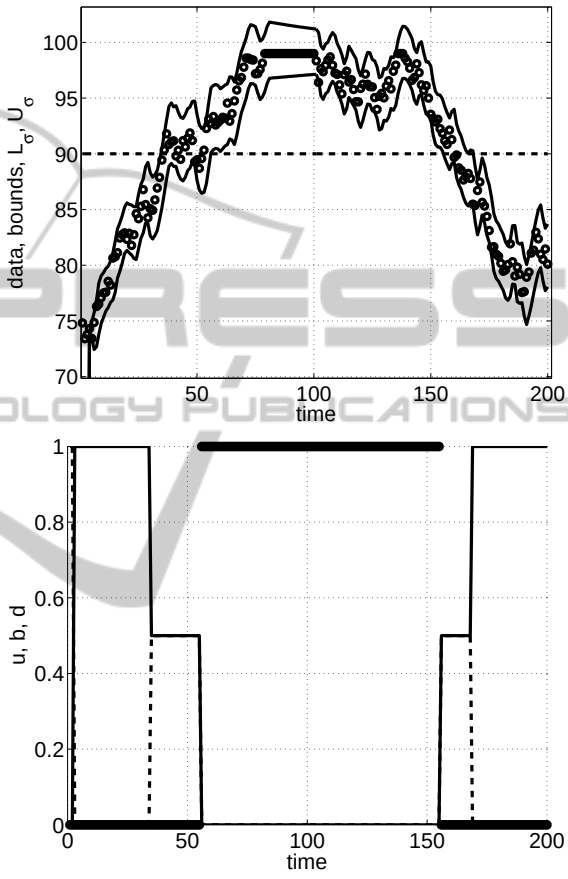

Figure 2: **Simulated data:** On top: CPU usage (circles) with highlighted boundary $\overline{\mathcal{U}}$ (9) (dashed line) and $L_{3\sigma}$, $U_{3\sigma}$ (10) (full line). Below: The course of b (thin line), d (thick line), u (dashed line) in ω (1) according to Table 1.

3.3 Discussion

A binomial opinion $\omega = (b, d, u, a)$ on the CPU condition, see (1), was evaluated. The components of the binomial opinion are interpreted as probabilities that the monitored block operates correctly (b), incorrectly (d) or there is an uncertainty about it (u). We used two types of simulated data.

The experiments show that single overshoots increase temporarily an uncertainty u in ω but they do not influence the disbelief d.

Concerning the permanent crossing of $\overline{\mathcal{U}}$, the disbelief d in ω is set only after both $L_{3\sigma}$ and $U_{3\sigma}$ (10) are outside the required area, which indicates the CPU overload. Otherwise, uncertainty u is increased only.

These results are in the accordance with conclusions that might be reached by the operator, i.e. when one of the σ-bounds is outside the required area $(\underline{\mathcal{U}}, \overline{\mathcal{U}})$ and the other one is inside, then operator could be uncertain about the CPU condition.

Use of the Gaussian noise may cause that the boundaries of the 3σ confidence interval exceed the physically meaningful set $[0\,\%, 100\,\%]$. This property

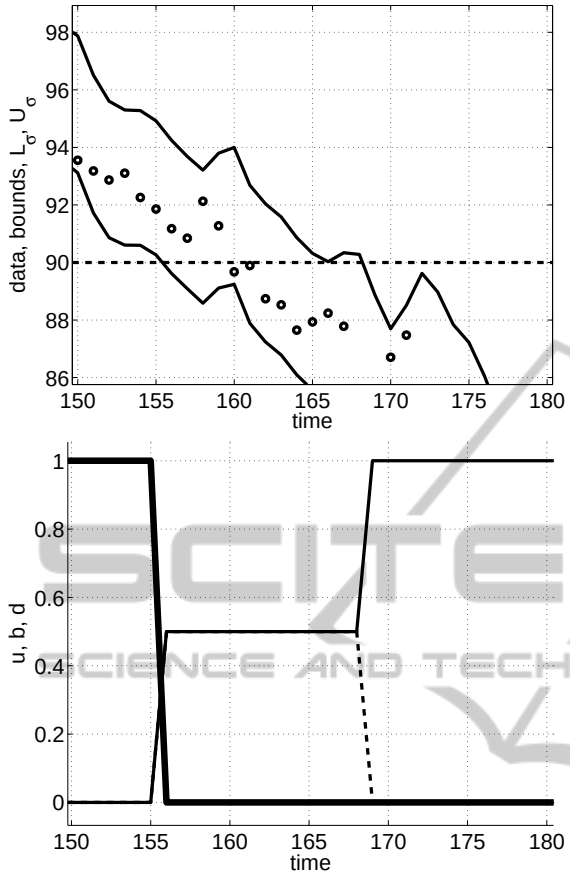

Figure 3: **Simulated data - zoomed part of Figure 2:** On top: CPU usage (circles) with highlighted boundary $\overline{\mathcal{U}}$ (9) (dashed line) and $L_{3\sigma}$, $U_{3\sigma}$ (10) (full line). Below: The course of b (thin line), d (thick line), u (dashed line) in ω (1) according to Table 1.

is practically not relevant because the boundaries of interest are $\underline{\mathcal{U}}$ and $\overline{\mathcal{U}}$.

4 CONCLUDING REMARKS

This paper proposes a straightforward method for evaluation of the CPU condition that is expressed in the form of binomial opinion, including uncertainty. The binomial opinion plays a role of a unified interface of building blocks that can be organized in a logical structure. The proposed method uses an autoregressive probabilistic model for modelling CPU usage $\mathcal{U}$. The parameters of this model are continuously estimated and 3σ intervals are constructed around the point output predictions. Boundaries of these intervals are compared with required CPU usage bounds $\underline{\mathcal{U}}$ and $\overline{\mathcal{U}}$. In this way, a binomial opinion on CPU condition is easily inferred using the assignment table. This table is constructed in accordance with the way of operator's decision-making. This opinion can be directly used within hierarchical condition monitoring (Dedecius and Ettler, 2014) where CPU belongs to set of basic building blocks.

Future works can focus on proposing of a function that assigns b, d, u continuously, depending on the distance between boundaries of 3σ intervals and bounds $\underline{\mathcal{U}}$, $\overline{\mathcal{U}}$. The proposed assignment table might be in fact too rough in some cases because the assignment of b, d, u switches between several discrete values. Models with bounded noise can be taken into consideration as well.

ACKNOWLEDGEMENTS

The research project is supported by the grant MŠMT 7D12004 (E!7262 ProDisMon).

REFERENCES

Akhtar, M. and Sidek, O. (2013). An intelligent adaptive arbiter for maximum CPU utilization, fair bandwidth allocation and low latency. *IETE Journal of Research*, 59(1):48–54.

Dedecius, K. and Ettler, P. (2014). Hierarchical modelling of industrial system reliability with probabilistic logic. In *Proceedings of the 11th international conference on informatics in control, automation and robotics (ICINCO)*, Vienna, Austria.

Farahnakian, F., Liljeberg, P., and Plosila, J. (2013). Lircup: Linear regression based cpu usage prediction algorithm for live migration of virtual machines in data centers. In *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*, pages 357–364.

Isermann, R. (2011). *Fault Diagnosis Applications: Model Based Condition Monitoring, Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems*. Springer Verlag.

Josang, A. (2008). Conditional reasoning with subjective logic. *Journal of Multiple-Valued Logic and Soft Computing*, 15(1):5–38.

Kárný, M., Böhm, J., Guy, T. V., Jirsa, L., Nagy, I., Nedoma, P., and Tesař, L. (2006). *Optimized Bayesian Dynamic Advising: Theory and Algorithms*. Springer.

Wang, W., Huang, X., Qin, X., Zhang, W., Wei, J., and Zhong, H. (2012). Application-level CPU consumption estimation: Towards performance isolation of multi-tenancy web applications. In *2012 IEEE 5th International Conference on Cloud Computing*, pages 439–446.