

# DETERMINING A PROCESSOR UTILIZATION MODEL BASED ON FUZZY C-MEANS CLUSTERING

AUGUSTO J. S. FIRMO\*, SAULO O. D. LUIZ†, ANTONIO M.N. LIMA†

\**Post-Graduate Program in Electrical Engineering - PPgEE - COPELE*

†*Electrical Engineering Department (DEE)  
Federal University of Campina Grande (UFCG)  
58429-900, Campina Grande - PB - Brazil*

Emails: [augusto.firmo@ee.ufcg.edu.br](mailto:augusto.firmo@ee.ufcg.edu.br), [saulo@dee.ufcg.edu.br](mailto:saulo@dee.ufcg.edu.br), [amnlima@dee.ufcg.edu.br](mailto:amnlima@dee.ufcg.edu.br)

**Abstract**— The present work presents the modeling of the processor utilization by means of Fuzzy inference systems. We have applied data analysis by means of the Fuzzy C Means clustering method. Experiments were also performed for validating the generated model. It was possible to improve the estimation and the generality of the model by using idleness as an input, enabling the estimation of the processor utilization for several workloads.

**Keywords**— Model, Modeling, Fuzzy, Fuzzy C means, FCM, processors, CPU, utilization, estimation, Takagi, Sugeno, Cluster, Clustering.

## 1 Introduction

Processors are a critical part of any electrical or electronic controlled system. From basic appliances to large factories or also computers of general-purpose and smartphones.

Processor utilization models are widely used in the estimation of consumption of systems fed by batteries, as shown by (Zhang et al., 2017), (Kim et al., 2012a) and (Kim et al., 2012b), for smartphones, for example.

In the case of general-purpose computers, due to the unpredictability of their workload over time, the development of an accurate model of energy consumption becomes quite complicated, being necessary the use of advanced estimation techniques for the accomplishment of such task, as presented by (Neves et al., 2013) and (Luiz et al., 2014). Therefore, it's important to consider situations where processor consumption is based on a utilization model. The power managers Ondemand<sup>1</sup> of Linux system or the Processor Power Manager<sup>2</sup> of Windows system are implemented without incorporating information about the power consumption models of the processors, and base frequency and voltage changes for the reduction of energy consumption in processors mainly in the estimation of their utilization. This is mainly due to the large number of processors on the market, being impossible to incorporate information of the models of the energy consumption of all available processors.

In this context, this paper analyzes the hypothesis that Fuzzy Logic can be used to modeling the processors utilization, mainly due to the applicability in complex systems to obtain a better

result.

## 2 Takagi-Sugeno Fuzzy Models

The principles of Fuzzy Logic were first developed by Jan Lukasiewicz (1878-1956), who in 1920 developed and introduced membership functions or membership groups, that combined with the concepts of classical logic, developed by Aristotle, gave sufficient background in order that in the 60's, Lofti Asker Zadeh, professor of computer science at the University of California proposed the Fuzzy Logic (Chenci et al., 2011).

The membership functions or membership groups, can be represented by different types of functions, such as gaussian, trapezoidal and triangular functions. By means of such functions, its possible evaluate how much a determinate data belongs to a particular group, and not only if it belongs to the particular group. The image of such functions is named degree of membership  $\mu$ , where  $\mu \in [0, 1]$ . An example of membership groups A and B for a generic input, with triangular membership functions is shown in Figure 1. Note that in the interval  $[0, 0.5]$  the input value is more belong to group A and less to group B, and in  $(0.5, 1]$ , it belongs more to group B and less to group A.

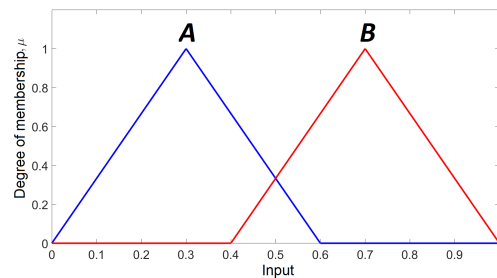


Figure 1: Membership functions for the example.

<sup>1</sup><https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt>

<sup>2</sup><https://docs.microsoft.com/en-us/windows-hardware/customize/power-settings/configure-processor-power-management-options>

There are different types of Fuzzy controllers described in the literature, differing mainly by their method used to evaluate the output. According to (Sugeno, 1985) these controllers can be grouped into the following:

1. Controllers based on Fuzzy membership functions and in operators for the definition of their Fuzzy output.
2. Controllers based on Fuzzy membership functions, but without operators for the definition of the output, Where is obtained by a linear or constant function. The Takagi-Sugeno type controllers belong to this group.

In both types of controllers, the control law is obtained by means of a set of rules. These are of type IF-THEN and may be divided in precedent, rule and consequent, as shown in the following example:

Precedent: if  $x$  is  $A$  and  $y$  is  $B$

Rule: if  $x$  is  $A$  and  $y$  is  $B$ , then  $z$  is  $C$

consequent:  $z$  is  $C$

This rule can be read as: If the input  $x$  belongs to the membership group  $A$  and the input  $y$  belongs to the membership group  $B$ , then the output  $z$  belongs to the membership group  $C$ . In this example,  $x$  and  $y$  are input variables of the process and  $z$  is the control output. The groups  $A$  and  $B$  correspond to the membership function  $\mu_A$  and  $\mu_B$ , referring to the input variables. For the Takagi-Sugeno type controllers,  $C$  is a first-order function.

In the current work, the rules for the Takagi-Sugeno controllers are described as:

Rule( $R_i$ ): if  $x$  is  $A_i$  and  $y$  is  $B_i$ , then  $z_i = f_i(x, y)$ , where  $i$  is the index of the rule and  $f_i(x, y)$  is a first-order function corresponding to the rule  $R_i$ .

The result of each rule is a numerical value and not a Fuzzy membership group, such as controllers with their output based on membership groups.

For Takagi-sugeno controllers with more than one input, the output  $z$  is obtained by weighing the various outputs obtained by each rule, as shown in (1).

$$z = \frac{\sum_{i=1}^n w_i z_i}{\sum_{i=1}^n w_i} \quad (1)$$

Where  $n$  is the number of controller rules,  $z_i$  is the output for the rule  $i$ ,  $w_i$  is the weight assigned to the rule  $i$ . This weight is determined as the lowest degree of membership between the inputs of the rule  $i$ , as shown in Figure 2. The weight  $w_1$  is determined as the lowest degree of membership of the inputs  $x$  and  $y$  to the membership groups  $A_1$  and  $B_1$

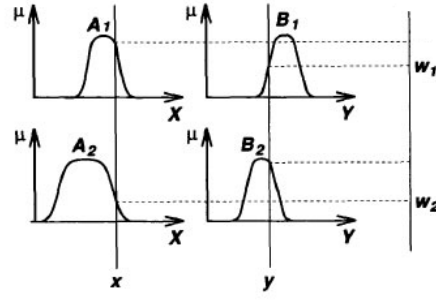


Figure 2: Definition of weights for the Takagi-Sugeno controller.

### 3 Fuzzy C Means algorithm

Fuzzy C Means (FCM) is a clustering method that allows the allocation of a piece of data in one or more groups, named clusters. This method was developed by (Dunn, 1973) and further enhanced by (Bezdek, 1981). It is widely used in pattern recognition and is based on minimizing the objective function shown in (2), where  $n$  is the numbers of data to be grouped,  $c$  is the numbers of adopted clusters,  $m$  is a positive real number, such as  $m \in (1, \infty)$ ,  $x_i$  is the  $i$ th data to be grouped,  $c_j$  is the  $j$ th cluster and  $\mu_{ij}$  is the degree of membership of the data  $x_i$  to the cluster  $c_j$ .

$$J = \sum_{i=1}^n \sum_{j=1}^c (\mu_{ij})^m \|x_i - c_j\|^2 \quad (2)$$

The initial values of degrees of membership  $\mu_{ij}$  are randomly defined and used in updating the clusters  $c_j$  as shown in (3).

$$c_j = \frac{\sum_{i=1}^n (\mu_{ij})^m x_i}{\sum_{i=1}^n (\mu_{ij})^m} \quad (3)$$

When the cluster values are updated, the values of the degrees of membership of each data may also be updated, as shown in (4).

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (4)$$

The objective function (2) is evaluated again and this process is repeated until  $J$  is less than a minimum error adopted or until it reaches the maximum number of iterations specified.

The methods presented in this section will be used for treating the data obtained in the experimental platform for the construction of the Fuzzy model.

## 4 Implementation

### 4.1 Measuring processor utilization

The experimental platform used was a personal computer composed of an Intel i5-6500 processor

with the frequency set {800 MHz, 1GHz, 1.1GHz, 1.3GHz, 1.5GHz, 1.7GHz, 1.8GHz, 2GHz, 2.2GHz, 2.3GHz, 2.5GHz, 2.7GHz, 2.9GHz, 3GHz, 3.2GHz}, 8 GB memory and 1TB hard drive. The adopted Operating System was the Linux Mint 18.3 Sylvia Cinnamon 64-bit.

The file `/proc/stat`<sup>3</sup> was used to measure the processor utilization. Such file has a variety of different system data since it was restarted for the last time. An example of its content regarding the processor and its cores is presented below:

```
cpu 79242 0 74306 842486413 756859 6140 67701
cpu0 49663 0 40234 104757317 542691 4420 39572
cpu1 2724 0 2118 105420424 767 1719 6084
cpu2 18578 0 18430 105191522 204592 0 714
cpu3 513 0 979 105428698 739 0 2907
cpu4 1623 0 2105 105426291 444 0 3373
cpu5 3491 0 5326 105414798 7134 0 3087
cpu6 1636 0 3081 105420689 201 0 8229
cpu7 1011 0 2029 105426670 288 0 3731
```

The sampling intervals of the processor cores are time units USER\_HZ or Jiffies (usually thousands of seconds, and may change according to the architecture or distribution used). The numerical values file `/proc/stat` are values of counters that are incremented at each sampling interval at which the processor or certain core performs a certain activity.

The first line “cpu” of the file `/proc/stat` contains data for all processor cores. The next lines “cpu*i*”, where  $i = 0, 1, 2, 3, \dots, 7$ , contain data for each core in the processor. Each column of these lines refers to some type of core activity:

- 1st column (User): processes running in user mode;
- 2nd column (Nice): high-priority processes running in user mode;
- 3rd column (System): processes running in kernel mode;
- 4th column (Idle): waiting time;
- 5th column (IOwait): waiting for input and output processes;
- 6th column (Irq): service interruptions;
- 7th column (Softirq): process maintenance time.

Based on these data, it is possible to evaluate the utilization of each processor core  $i$ . The sum of all these columns is named  $t_{total}$  and the sum of the Idle and IOwait columns is named  $t_{idle}$ . Measurements are performed at each sampling interval. The elapsed interval  $\Delta t_{total}$  and the idle interval  $\Delta t_{idle}$  are shown in (5) and (6) respectively.

$$\Delta t_{total}(t) = t_{total}(t) - t_{total}(t-1) \quad (5)$$

$$\Delta t_{idle}(t) = t_{idle}(t) - t_{idle}(t-1) \quad (6)$$

The utilization  $y$  and idleness  $\iota$  are evaluated as shown in (7) e (8) respectively.

$$y(t) = \frac{\Delta t_{total}(t) - \Delta t_{idle}(t)}{\Delta t_{total}(t)} \quad (7)$$

$$\iota(t) = \frac{\Delta t_{idle}(t)}{\Delta t_{total}(t)} = 1 - y(t) \quad (8)$$

#### 4.2 Selecting the processor frequency

(Luiz, 2012) described some usual workloads on general-purpose computers, showing the relation between frequency and utilization. Examples of the average utilization as a function of frequency, for a light and a heavy workload are shown in Figures 3 and 4 respectively.

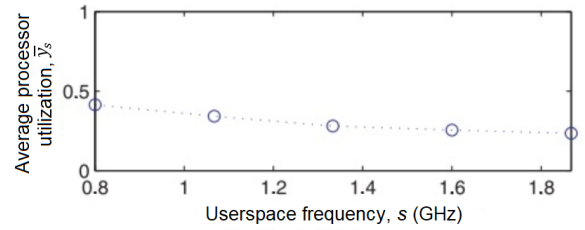


Figure 3: Utilization as a function of frequency for a light workload.

Font: Adapted from (Luiz, 2012).

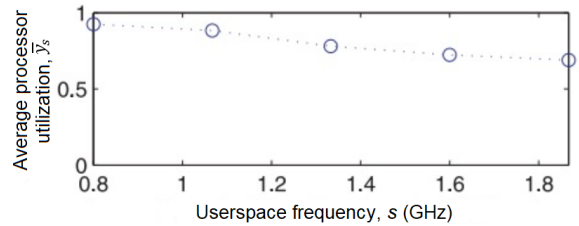


Figure 4: Utilization as a function of frequency for a heavy workload.

Font: Adapted from (Luiz, 2012).

In this work, the data collected from the processor were the chosen frequencies, the idleness and the utilization. Workloads were selected to generate demands that cause multiple utilization ranges on the processor, these were defined according to the range of coverage. Extreme<sup>4</sup>, covering utilization between 55% and 95%. Heavy<sup>2</sup>, covering utilization between 35% and 85%. Medium<sup>2</sup>, covering utilization between

<sup>3</sup><http://www.linuxhowtos.org/manpages/5/proc.htm>

<sup>4</sup><http://bbb3d.renderfarming.net/download.html>

Table 1: Description of the used workloads.

Workload	Description
Extreme	Video with resolution 4K (3840x2160) 60 fps (frames per second)
Heavy	Video with resolution 4K (3840x2160) 30 fps
Medium	Video with resolution 1080p (1920x1080) 60 fps
Light	Video with resolution 1080p (1920x1080) 30 fps
Minimum	Music

15% and 55%. Light<sup>5</sup> covering utilization between 10% and 35%. And Minimum<sup>6</sup>, covering utilization between 3% and 14%. These workloads are shown in Table 1.

A software was implemented in C to run in the Linux userspace and apply the same frequencies to all the processor cores by means of the set of utilities indicator-cpufreq<sup>7</sup> and measure the utilization  $y$  and idleness  $\iota$ , using the methodology presented in Section 4.1.

The tests performed to train the model consisted in running the workloads described in Table 1 in different frequencies, and measuring the utilization  $y$  and idleness  $\iota$  of the processor during the test (first line of the file `/proc/stat`). The average utilization and the idleness along time are shown in Figures 5 and 6 respectively. The frequency was increased every 140 seconds, from 800 MHz up to 3.2 GHz. The variation in utilization and idleness values when the frequency is constant occurs because of the scheduling of userspace workloads and Kernel workloads.

#### 4.3 The model estimation

In this work, a data set with the maximum frequency divided by the current frequency, lagged idleness and utilization, is defined in (9).

$$Z_l^N = \{(u(t), \iota(t-1), y(t))\}, t = 1, 2, \dots, N \quad (9)$$

$$l \in \{ext, hea, med, lig, min\}$$

where  $N$  stands for the number of data points, and  $l$  denotes the type of workload.

The following data sets were acquired:  $Z_{ext}^{2200}$  for extreme load,  $Z_{hea}^{2200}$  for heavy load,  $Z_{med}^{2200}$  for medium load,  $Z_{lig}^{2200}$  for light load e  $Z_{min}^{2200}$  for minimum load. Such data sets are presented in Figures 5 and 6.

The sets  $u_{ext}(t)$ ,  $u_{hea}(t)$ ,  $u_{med}(t)$ ,  $u_{lig}(t)$  and  $u_{min}(t)$  represent the maximum frequency divided by the actual frequency values shown in Figure 6.

<sup>5</sup><https://durian.blender.org/download/>

<sup>6</sup><http://freemusicarchive.org/>

<sup>7</sup><https://launchpad.net/indicator-cpufreq>

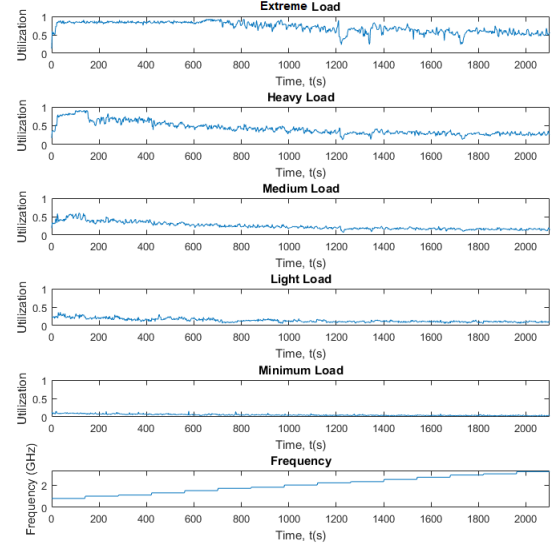


Figure 5: Utilization for different frequencies and workloads.

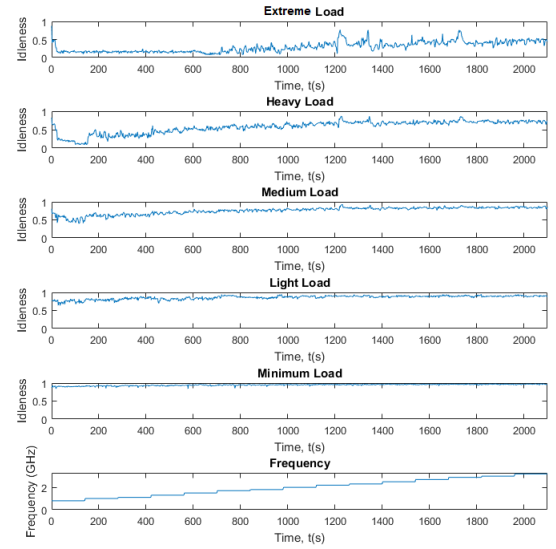


Figure 6: Idleness for different frequencies and workloads.

The frequency was as shown in equation (10), where  $s_0$  is the maximum processor frequency and  $s(t)$  is the actual processor frequency. The division occurs because the utilization is a monotonically increasing function of  $\frac{s_0}{s}$ .

$$u(t) = \frac{s_0}{s(t)} \quad (10)$$

The model presented by (Luiz, 2012) uses as input the sets  $u_{ext}(t)$ ,  $u_{hea}(t)$ ,  $u_{med}(t)$ ,  $u_{lig}(t)$  and  $u_{min}(t)$ , and is presented in (11). As can be seen in Figure 7, different values of  $y^{on}$  and  $y^{off}$  are generated for each type of workload. The estimation of these parameters at run time requires the use of an estimation technique such as least squares recursive, this may result in an overhead in the power manager, which can be avoided through the use of

$\iota(t-1)$ , which gives an indirect indication of the type of workload being executed, and thus, when added to the model makes it more generic.

$$y(t) = u(t)y^{\text{on}} + y^{\text{off}} \quad (11)$$

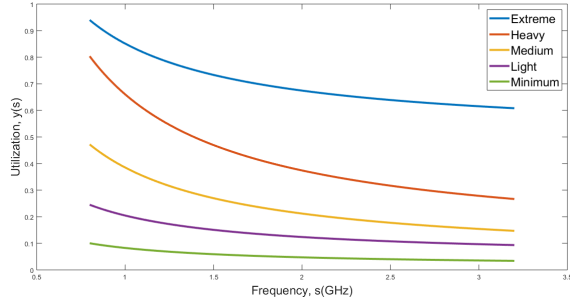


Figure 7: Curves for each workload.

The inputs of the Fuzzy model were the data sets of idleness  $\iota_{ext}(t-1)$ ,  $\iota_{hea}(t-1)$ ,  $\iota_{med}(t-1)$ ,  $\iota_{lig}(t-1)$  and  $\iota_{min}(t-1)$  and maximum frequency divided by the current frequency  $u_{ext}(t)$ ,  $u_{hea}(t)$ ,  $u_{med}(t)$ ,  $u_{lig}(t)$  and  $u_{min}(t)$ . As output of the model was the set of utilization  $y_{ext}(t)$ ,  $y_{hea}(t)$ ,  $y_{med}(t)$ ,  $y_{lig}(t)$  and  $y_{min}(t)$ . First, from each individual workload data set (composed of utilization and idleness), clusters were generated by means of the FCM algorithm presented in Section 3. The stopping criterion was chosen as the maximum error of 5% or maximum of 100 iterations. Then the clusters for each workload data set were grouped in two sets of clusters, one for the idleness input and one for the utilization output. It was seen that maximum errors of less than 5 % did not produce much better results on the quality of clusters.

The clusters generated for each dataset are used to create the membership functions to the Fuzzy model, for each workload.

Gaussian membership functions were adopted, since these have only two parameters, unlike the triangular and trapezoidal functions that have 3 and 4 parameters respectively. The centers are defined as the clusters and the width  $\sigma$  is defined as shown in (12).

$$\sigma = \frac{\sum_{i=1}^n \sqrt{\frac{-(x_i - c_j)^2}{2 \log(\mu_{ij})}}}{n} \quad (12)$$

Where  $x_i$  is the data of an input,  $c_j$  is the cluster used as the center of the gaussian membership function and  $\mu_{ij}$  are the degrees of membership of the data  $x_i$  to the cluster  $c_j$  as shown in (4).

The membership functions obtained for the frequency input are shown in Figure 8. The input  $u(t)$  varies between 1 and 4, where 1 corresponds to  $s(t) = s_0$  and 4 corresponds to  $s(t) = 800$  MHz.

The membership functions for the input idleness  $\iota(t-1)$  for the extreme, heavy, medium, light and minimum workloads are shown in Figures 9,

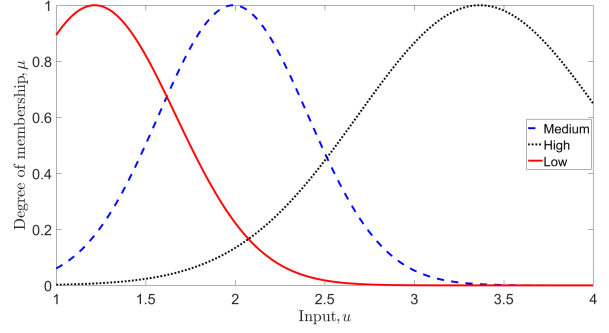


Figure 8: Membership functions for the input  $u$ .

Table 2: Parameters for the output functions for the model rules.

Outputs	$a_1$	$a_2$	$a_3$
$z_{ext}$	$-2.855 \times 10^{-9}$	-1	1
$z_{hea}$	$-1.535 \times 10^{-8}$	-1	1
$z_{med}$	$4.747 \times 10^{-9}$	-1	1
$z_{lig}$	$-3.846 \times 10^{-8}$	-1	1
$z_{min}$	$2.144 \times 10^{-9}$	-1	1

10, 11, 12 and 13 respectively. The idleness values corresponding to the centers of the membership functions increase from the extreme load to the minimum load, because the extreme load is less idle than the minimum load. The membership functions for idleness input  $\iota(t-1)$  of the model is composed of all the membership functions presented for all workloads (Figures 9, 10, 11, 12 and 13).

The output functions for the model rules are obtained by least squares method for the datasets of each workload. This way, we have the output for the Fuzzy system as shown in (13).

$$z_i = a_1 u(t) + a_2 \iota(t-1) + a_3 \quad (13)$$

Solving  $a_1$ ,  $a_2$  and  $a_3$  as shown by (Gruen and Akca, 2005), for each workload, it's obtained five output function as shown in Table 2. Where  $z_{ext}$  is the function for the extreme workload,  $z_{hea}$  is the function for the heavy workload,  $z_{med}$  is the function for the medium workload,  $z_{lig}$  is the function for the light workload and  $z_{min}$  is the function for the minimum workload.

The rules for the Fuzzy model are shown in Table 3.

The block corresponding to the developed model is shown in Figure 14. The input  $\iota(t-1)$  is used to detect the system workload and the input  $u(t)$  to detect the current processor frequency.

## 5 Model validation

In this section, the results of a validation experiment will be presented. By means of the inputs in the validation data and the Fuzzy model presented in Section 4.3, outputs were estimated and

Table 3: Rules for Fuzzy model.

Rules	$u(t)$	Inputs		Output $y(t)$
		$t$ -norm	$\iota(t-1)$	
$R_1$	Medium	AND	$Extreme_1$	$z_{ext}$
$R_2$	High	AND	$Extreme_2$	$z_{ext}$
$R_3$	Low	AND	$Extreme_3$	$z_{ext}$
$R_4$	Medium	AND	$Heavy_1$	$z_{hea}$
$R_5$	High	AND	$Heavy_2$	$z_{hea}$
$R_6$	Low	AND	$Heavy_3$	$z_{hea}$
$R_7$	Medium	AND	$Medium_1$	$z_{med}$
$R_8$	High	AND	$Medium_2$	$z_{med}$
$R_9$	Low	AND	$Medium_3$	$z_{med}$
$R_{10}$	Medium	AND	$Light_1$	$z_{lig}$
$R_{11}$	High	AND	$Light_2$	$z_{lig}$
$R_{12}$	Low	AND	$Light_3$	$z_{lig}$
$R_{13}$	Medium	AND	$Minimum_1$	$z_{min}$
$R_{14}$	High	AND	$Minimum_2$	$z_{min}$
$R_{15}$	Low	AND	$Minimum_3$	$z_{min}$

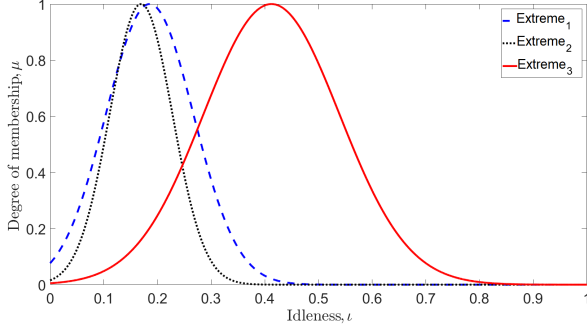


Figure 9: Membership functions for idleness input  $\iota$  and extreme load.

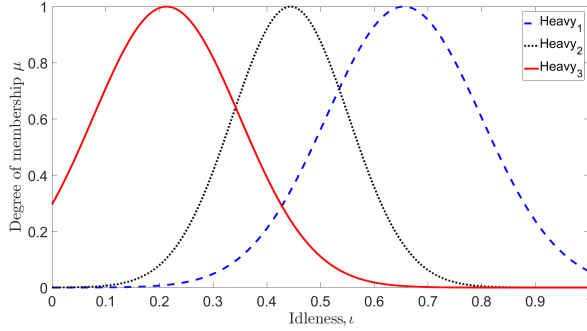


Figure 10: Membership functions for idleness input  $\iota$  and heavy load.

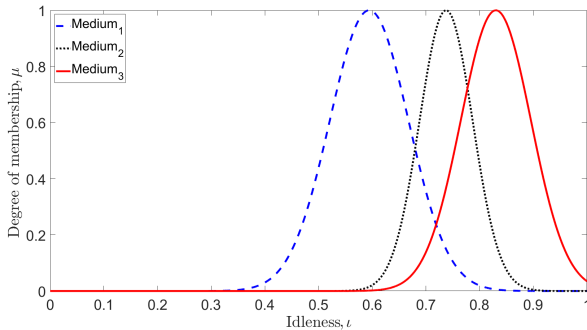


Figure 11: Membership functions for idleness input  $\iota$  and medium load.

compared to the outputs in the validation data. The validation data is composed of several differ-

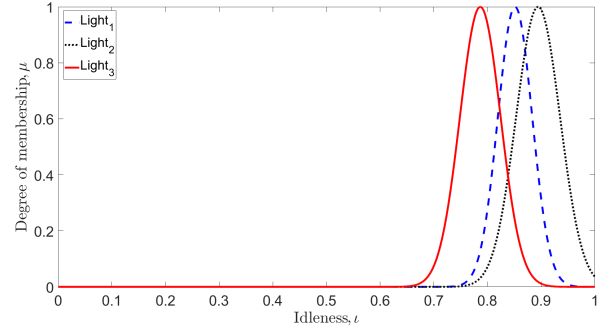


Figure 12: Membership functions for idleness input  $\iota$  and light load.

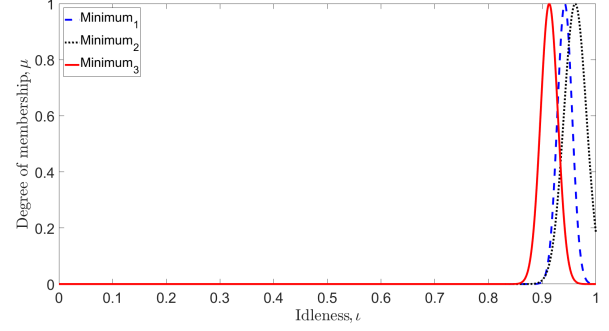


Figure 13: Membership functions for idleness input  $\iota$  and minimum load.

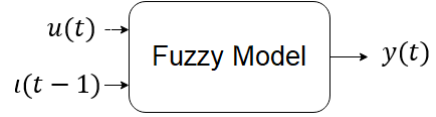


Figure 14: Block corresponding to the Fuzzy model.

ent workloads at different frequencies. The loads were the open license videos and songs. During the experiment, the computer network interfaces were disconnected, and the update processes and other non-essential processes of the operating system were canceled.

The experiment was performed in the following sequence:

1. For  $t \in [0, 140]$ s: Video reproduction with resolution 2K (2048x1080), 30 fps, and processor with 3 GHz frequency.
2. For  $t \in (140, 280]$ s: Video reproduction with resolution 2K (2048x1080), 60 fps, and processor with 2.5 GHz frequency.
3. For  $t \in (280, 420]$ s: Video reproduction with resolution 4K (3840x2160), 60 fps, and processor with 2.7 GHz frequency.
4. For  $t \in (420, 560]$ s: Video reproduction with resolution 720p (1280x720), 60 fps, and processor with 1.5 GHz frequency.



5. For  $t \in (560, 700]s$ : Video reproduction with resolution 360p (640x360), 30 fps, and processor with 1 GHz frequency.
6. For  $t \in (700, 840]s$ : Music reproduction, and processor with 800 MHz frequency.

The validation data and the estimated data are shown in Figure 15.

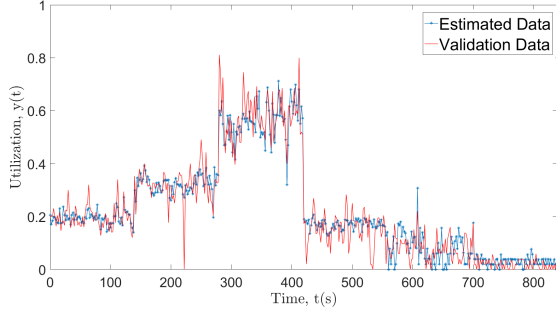


Figure 15: Result of the validation experiment, with  $\epsilon_N = 0.0027$ .

The error between the validation and the estimation data was the root mean squared error, as shown in (14). An error of 0.0027 was obtained, with the weight  $w_t$  equal to 1 for all estimations.

$$\epsilon_N = \sqrt{\frac{\sum_{t=1}^N w_t (y(t) - \hat{y}(t))^2}{N}} \quad (14)$$

The model proposed in this work was compared to the model developed by (Luiz, 2012) presented in (11), where  $u(t)$  is presented in (10). The parameters  $y^{\text{on}}$  and  $y^{\text{off}}$  were estimated for each workload presented in this section. Such parameters were estimated by means of curve fitting, and the result are shown in Table 4.

Table 4: Parameters  $y^{\text{on}}$  and  $y^{\text{off}}$ .

Workloads	$y^{\text{on}}$	$y^{\text{off}}$
Extreme	0.1106	0.4975
Heavy	0.1791	0.0876
Medium	0.1083	0.0386
Light	0.0506	0.0428
Minimum	0.0221	0.0120

The validation and estimation data are shown in Figure 16. The estimated output follows the average value of the utilization  $y$ , this occurs because this model don't uses idleness as a input, causing an error of 0.0042, greater than the error of 0.0027 obtained by the model presented in this work.

## 6 Conclusion

In this work, a Fuzzy model for the processor utilization was developed using data clustering techniques. The proposed model was validated as de-

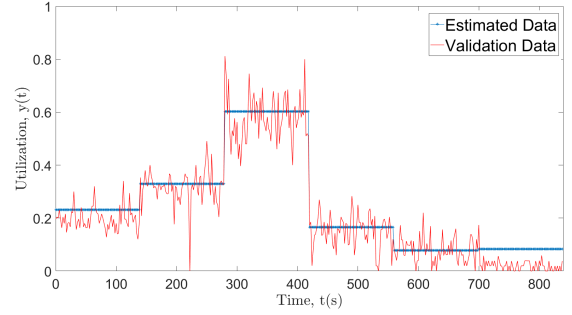


Figure 16: Result of the validation experiment with the model developed by (Luiz, 2012), with  $\epsilon_N = 0.0042$ .

scribed in Section 5. It was also possible to improve the estimation of processor utilization by means by including the idleness as input of the model. The generality of the model has also been improved, because the utilization due to several types of workloads, may be estimated by means of a single model, not requiring the use of different parameters for each type of load and another type of estimation for the current processor workload as presented by (Luiz, 2012).

## Acknowledgment

The authors would like to thank the PPgEE-COPELE and CAPES for the necessary support during this work.

## References

- Bezdek, J. C. (1981). Objective function clustering, *Pattern recognition with fuzzy objective function algorithms*, Springer, pp. 43–93.
- Chenci, G. P., Rignel, D. G. and Lucas, C. A. (2011). Uma introdução a lógica fuzzy, *Revista Eletrônica de Sistemas de Informação e de Gestão Tecnológica* **1**(1).
- Dunn, J. C. (1973). A fuzzy relative of the iso-data process and its use in detecting compact well-separated clusters, *Journal of Cybernetics* **3**(3): 32–57.
- Gruen, A. and Akca, D. (2005). Least squares 3d surface and curve matching, *ISPRS Journal of Photogrammetry and Remote Sensing* **59**(3): 151–174.
- Kim, M., Kong, J. and Chung, S. W. (2012a). Enhancing online power estimation accuracy for smartphones, *IEEE Transactions on Consumer Electronics* **58**(2).
- Kim, M., Kong, J. and Chung, S. W. (2012b). An online power estimation technique for

- multi-core smartphones with advanced display components, *Conf. Rec. ICCE*, IEEE, pp. 666–667.
- Luiz, S. O. D. (2012). Gerenciamento dinâmico de energia em processadores com cargas de trabalho variantes no tempo, *Tese de Doutorado, UAEE/UFCG, Campina Grande, PB*.
- Luiz, S. O. D., Silva, A. V. d. N., Santos, D. R. and Melo, T. (2014). Increasing power models accuracy by means of synchronized measurements, *Conf. Rec. ICCE*, IEEE, pp. 318–322.
- Neves, B. H., Cruz, B. M., Mendes, R. R., Bublitz, F. M., Silva, J., Luiz, S. O. D., Perkusich, A. and Almeida, H. (2013). Computer systems power model estimation, *Conf. Rec. ICCE*, IEEE, pp. 173–176.
- Sugeno, M. (1985). An introductory survey of fuzzy control, *Information Sciences* **36**(1): 59 – 83.
- Zhang, Y., Liu, Y., Liu, X. and Li, Q. (2017). Enabling accurate and efficient modeling-based cpu power estimation for smartphones, *Conf. Rec. IWQoS*, IEEE, pp. 1–10.