# Maximum Entropy Stochastic Tasks Classification

Miguel A. Toledo C. and J. Jesús Medel J.

Instituto Politécnico Nacional, CIC
jjmedelj@yahoo.com.mx, matoledo@uvmnet.edu

**Abstract.** In this work we introduce a task classification model for soft real-time systems. Task processes are analyzed as probabilistic distribution functions which parameters are considered as stationary random variables over the long time. We present a technique for tasks classification based on the maximum entropy level as element of differentiation among tasks. The classification model is described and evaluated based on probabilistic distribution functions properties.

## 1 Introduction

In computer systems, the term "tasks" is referred to those processes which are executed in servers whose perform a specific action. Tasks are software entities that aim to respond to the information generated by Real-time systems; peripheral devices or internal processes [1]. In applications with explicit timing constrains and logical correctness, tasks completion algorithms that accomplish these requirements has been purposed in many works. Tasks scheduling involve certain knowledge of tasks characteristics. Since its earliest definitions, tasks have been classified as: periodic, semiperiodic, aperiodic or sporadic for various authors according the analysis context [1],[7], [8], [16]. In most of cases, task classification is established priory. Periodicity in generated events mark a classical division among tasks types.

Scheduling in real time systems concerns the determination of a temporal ordering of tasks within constraints of some specified timing, precedence and resource requirements. Depending on the nature of the application, scheduling may be classified into different kinds: static scheduling, dynamic scheduling and mixed scheduling [8].

Static scheduling assumes prior knowledge of the relevant characteristics of all tasks, which may be taken into account in scheduling. When this information is available priority ordering is assigned to tasks. However, any change in tasks requires the complete rescheduling. On the other hand, Dynamic scheduling algorithms are designed to work with unpredictable arrival times of tasks and possible uncertainties in their execution times or deadlines; priority assignation is done in line; that is, tasks priority is assigned at the task arrival time.

Assign priority ordering only have sense when tasks are preemptable. A task is preemptable by other task with higher priority if it can be interrupted and resumed later meanwhile its overall goal can be achieved. In dynamic scheduling algorithms, tasks assigned priority vary over time from request to request, but in a statistical way the priority tasks ordering remains the same.

*© A. Gelbukh, S. Suárez. (Eds.)*
*Advances in Computer Science and Engineering.*
*Research in Computing Science 23, 2006, pp. 53-59*

*Received 08/07/06*
*Accepted 07/10/06*
*Final version 10/10/06*

Tasks priority ordering depends mostly on real time constraints; such as arriving time or deadline. Scheduling based on priority ordering aims to deal with these constraints optimally. In dynamic priority scheduling algorithms, where tasks characteristics are known at the time they arrives, statistical knowledge of their process is required.

In stationary systems which maintain its characteristics over long time, it is possible to determine some tasks performance information based on their statistics. Tasks process outcomes can be described by density distribution function with defined parameters. Probability does not say much about individual events, but describes a faithfulness level of task predictability. Meanwhile tasks processes repeat during an infinite number of times. Some of these relevant variables are: incoming channel, inter-arrival time or period, relative deadline and, computations or communications streams. Here tasks processes are considered stochastic stationary processes with defined probability distribution function; likewise task statistical behavior can be described. In soft-real-time systems, it is statistically required that task processes be completed in their deadlines. This means that a statistical distribution response time service is acceptable.

While the system evolves in time, tasks knowledge increases. A measure of amount of uncertainty can be represented in probability with the maximum entropy. That is, the information is more informative kind, if it consists of mean values of task variables. Therefore, the analysis of tasks processes estimate a measure of the degree to which a system is schedulable. One such measure is the task classification on maximum entropy level what is the focus of this work.

The purpose of this paper is introducing a scheduling theory based on tasks continuous stochastic processes for soft-real-time systems. Our aim is give an entropy based classification for tasks treatment. As well as task classification is generated while the process evolves. Tasks classification helps us in scheduling algorithms based on tasks characteristics.

## 2 Preliminaries

In this section, we define task functions as continuous time stochastic processes. This approach let us to handling any kind of task; where periodicity in task time arrival becomes a density distribution function. Based on M/M/1 queueing theory [4], tasks time constraints; arriving and deadline time are independently assigned according to a probability distribution. Next assumptions are made in order to simplify out model, but they can be treated in future works.

1. An task becomes a active an instant after it arrives.
2. All tasks are preemptive and overhead are depreciable.
3. Deadline is result of arrival time and calculus time.
4. Start time, lateness and precedence are considered nulls.

Let the tuple $(\Omega, \mathcal{F}, \Pr)$ be the probability space that describe task process. The $\Omega$ set throws outcomes out during the sample time. Here $\mathcal{F}$ is the set field in $\Omega$. Then $\Pr$ is the probability measure function in this space. Tasks constraints are random variables generated by a random variables family whom are none decreasing in time [9], [10].

**Definition 1.** *(Real-time Task). A real-time task is a stochastic process described by $\pi\{A(t,\omega), I(t,\omega), D(t,\omega)\} \in \mathcal{F}$, characterized by the random variables family A,I and D defined in the probability space $(\Omega, \mathcal{F}, \Pr)$ with $t\epsilon T$ $t > 0$, and $\omega \subset \Omega$.*

**Definition 2.** *(Arrival Time). The arrival time is a stochastic process $\{A(t,\omega)\} \in \mathcal{F}$ defined as the required waiting time to observe an arrival occurrence when this outcomes at rate $\lambda$ time units, is described by an exponential density function.*

$$\begin{aligned} A(t,\omega) &= \frac{1}{\lambda}e^{-\frac{1}{\lambda}t} \\ A(t,\omega) &> 0 \end{aligned} \tag{1}$$

**Definition 3.** *(Task Information). The information contained in tasks that will be executed according to server performance is a stochastic process $\{I(t,)\} \in \mathcal{F}$ described by an normal density function.*

$$\begin{aligned} I(t,\omega) &= \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{t^2-\mu^2}{2\sigma^2}} \\ I(t,\omega) &> 0 \end{aligned} \tag{2}$$

**Definition 4.** *(Deadline). The time when a task must finish its service is a stochastic process $\{D(t,)\} \in \mathcal{F}$ described by the sum of $A(t,\omega)$ and $I(t,\omega)$; $D(t,\omega) = A(t,\omega) * I(t,\omega)$ Thus the normalized Deadline distribution function is defined by*

$$\begin{aligned} D(t,\omega) &= \frac{exp(-\frac{1}{2}\frac{\sigma^2+2t\lambda-2\mu\lambda}{\lambda^2})}{\lambda exp(\frac{\sigma^2}{2\lambda^2}+\frac{\mu}{\lambda})} \\ D(t,\omega) &> 0 \end{aligned} \tag{3}$$

Task information is taken as the time unit to determine the calculation time. Based on the calculation time is a product of the task information and a constant of process capability of the system. So that, in following developments theses will be teated as synonyms. As independent random variables, the deadline process is the sum of the arrival time and the task information or calculation time.

## 3 Maximum Entropy Principle

In 1948 Claude Shannon [14] published his theory of communication; where he derives a measure of uncertainty denoted entropy. Referring to some system with certain physical or conceptual entities, the messages they produce have meaning. Shannon proposed a measure of how much information is "produced" by such a process, or better, at what rate information is produced.

The measure $H(p_1, p_2, ..., p_n)$ of a set of possible events whose probabilities of occurrence are $p_1, p_2, ..., p_n$. The entropy is the measure function of the uncertain of an event outcome. It is reasonable to such a measure to require of it the following properties:

1. $H$ should be continuous in the $p_i$.

2. If all the $p_i$ are equal, $p_i = 1/n$ then $H$ should be a monotonic increasing function of $n$. With equally likely events there is more choice, or uncertainty, when there are more possible events.
3. If a choice be broken down into two successive choices, the original H should be the weighted sum of the individual values of H.

**Definition 5.** *(Entropy). The only function H that satisfies the three above assumptions is of the form:*

$$H(p_1, p_2, ..., p_n) = -K \sum_{i=1}^{n} p_i \log p_i \tag{4}$$

The form $H = \sum p_i \log p_i$ is recognized as entropy for probabilities $p_i$ because of the mathematical similarity with the thermodynamical definition of entropy. Typically, $K$ is taken as unity and the logarithm either in base 2 or Napierian (natural) base. In an analogous manner the entropy of a continuous distribution with the density distribution function $p(x)$ is given by:

$$H(x) = -\int_{-\infty}^{\infty} p(x) \log p(x) dx \tag{5}$$

For task processes with known density distributions; mathematical maximization techniques like Lagrange multipliers are used to determine a measure of certainness for continuous random variables based on the maximum entropy principle.

**Theorem 1.** *(Maximum task arrive entropy). Lets $A(t, \omega)$, $t > 0$, be the arrival time of a task, subject to the following constraints:*

1. $\int_0^\infty A(t, \omega) dt = 1$
2. $\int_0^\infty t A(t, \omega) dt = \lambda$ *for* $\lambda > 0$
3. $H(A(t, \omega)) = -\int_0^\infty A(t, \omega) \log A(t, \omega) dt$

*The maximum entropy for $A(t, \omega)$ variable with pre-specified first moment in $[0, \infty)$ is $H_{i,M}$ given by*

$$H(A(t, \omega)) = \log \lambda e \tag{6}$$

*Proof.* In stochastic sense, the arrival time function $A(t, \omega)$ is defined as an exponential type function, [4] and [6]. Using the method of Lagrangian multipliers, the density distribution that satisfies the constraints 1) and 2), is the one described in DEF 2. The associated maximal entropy is then obtained by $H(A(t, \omega)) = -\int_0^\infty \frac{1}{\lambda} e^{-\frac{1}{\lambda}t} \log \frac{1}{\lambda} e^{-\frac{1}{\lambda}t} dt = ln\lambda \int_0^\infty \frac{1}{\lambda} e^{-\frac{1}{\lambda}t} dt + \frac{1}{\lambda} \int_0^\infty \frac{t}{\lambda} e^{-\frac{1}{\lambda}t} dt \; H(A(t, \omega)) = ln\lambda + 1$

**Theorem 2.** *(Maximum task deadline entropy). Lets $D(t, \omega)$, $t > 0$, be the deadline of a task, subject to the following constraints:*

1. $\int_0^\infty D(t, \omega) dt = 1$
2. $\int_0^\infty t D(t, \omega) dt = \lambda$ *for* $\lambda > 0$
3. $H(D(t, \omega)) = -\int_0^\infty D(t, \omega) \log D(t, \omega) dt$

*the maximum entropy for $D(t, \omega)$ variable with pre-specified first moment in $[0, \infty)$ is $H_{i,M}$ given by*

$$H(D(t, \omega)) = \log \lambda e \tag{7}$$

*Proof.* Deadline defined by the convolution between task information and arrival time distribution functions where $D(x, \omega) = \int_{-\infty}^{\infty} A(t, \omega) I(x - t, \omega) dt$ normalized is considered. Thus, the density distribution that satisfies the constraints 1) and 2), is the one described in DEF 4. The associated maximal entropy is then obtained by $H(D(x, \omega)) = -\int_0^{\infty} D(x, \omega) \log D(x, \omega) dt$, this yield $H(D(x, \omega)) = ln\lambda + 1$

## 4 Maximum Entropy Tasks Classification

In previous section we establish a probabilistic criteria for task classification in dynamic algorithms, according with tasks temporal constraints. As in firsts works in the area, periodicity and deadline constraints have been considered bases on task planning algorithms.

A main task property is the period. In Liu and Lyland pioneer analysis, [7], the length of successive tasks is a constant called period $T$. A periodic task is said to have a regular release time, or it is regular time triggered. If a task does not accomplish with this criteria is called a nonperiodic or sporadic task. Task periodicity plays a relevant role in scheduling algorithms because of the assumptions considered in its analysis; in order to have the expected results [5]. Periodicity is considered as a measure of task order timely speaking. A higher hierarchy task corresponds a higher periodicity task. What is defined by the task arrival distribution function.

Deadline property is considered in dynamic scheduling as the EDF algorithm [15]. By means of the processor utilization factor, task priority is selected by the absolute deadline. Tasks with earlier deadlines will have higher priorities.

**Definition 6.** *(Hierarchy in Probabilistic Sense). Let $J_M = J(\pi(t, \omega))$ be the task parameter assigned that describes the relative importance among tasks in the system*

$$J(\pi(t, \omega)) = min \begin{cases} H(A(t, \omega)) \\ H(D(t, \omega)) \end{cases} \tag{8}$$

**Theorem 3.** *(Tasks Hierarchy based on Maximum Entropy). Lets $\pi_i(t, \omega)$ be a task process with arrival time $A_i(t, \omega)$, $I_i(t, \omega)$ and $D_i(t, \omega)$. Task hierarchy of $J_{i,M}$ respect to $J_{j,M}$ according with the maximum entropy principle is established by*

$$J_{i,M} > J_{j,M} \qquad \textit{if and only if} \qquad H_{i,M} < H_{j,M} \tag{9}$$

*Proof.* Results of THEO 1 and THEO 2 relate maximum entropy measure with the distribution density mean for the task parameters under consideration. Accomplish with inequality $H_{i,M} < H_{j,M}$ give the hierarchy inequality

## 5 Simulation Results

Assigning hierarchy to task processes is a first step in task planning due to the relevance of classify tasks in dynamic scheduling algorithms. That is, while tasks arrive and we have enough information about their probabilistic parameters, we are able to distinguish those tasks that must be served first from the others.

A simulation algorithm for task classification using the maximum entropy principle is presented. Here we take tasks arriving time samples from whom we observed their statistical parameters during the simulation time. With the first probability moment values we applied the classification based on maximum entropy principle in order to give them a relative hierarchy giving an order of tasks to be processed by the server.

In Figure 1 a simulation outcome is showed. Here, the simulation algorithm is tested for three tasks with different first probability moments which arriving task density distribution are stationary. Moreover, the tasks classification is presented for these task samples giving them a priority value in function of maximum entropy certainness measure.

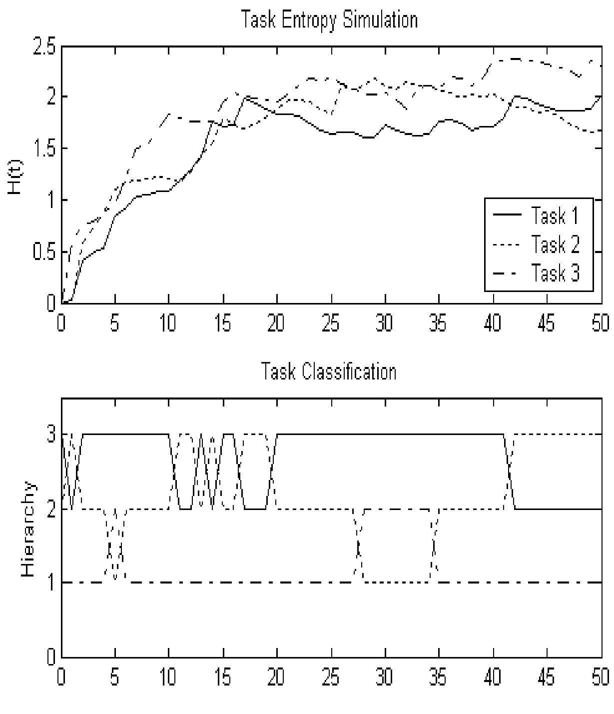


**Fig. 1.** A tasks classification simulation outcome

We observe from the tasks hierarchy values for figure 1), that while the simulation time evolve; these values remains unchanged because of the information gathered from tasks.

## 6 Summary and Conclusions

In this paper the priority rule problem for task planning is treated. Assigning priority to tasks based on the maximum entropy principle let us deal with tasks with differences in their statistical parameters. The technique presented here is applied to not only periodic tasks but aperiodic tasks also. A remarkable result of this work is the classification method, in relation with tasks periodicity and deadline constraints is a similar manner.

Based on the maximum entropy level as element of differentiation among tasks, this theory gives a continuous analysis in task planning processes. We have modeled soft real-time systems as tasks processes with stationary distribution functions. This analysis establish the basis for scheduling algorithms future works.

## References


1. G. Buttazzo. *Hard Real-time Computing Systems*. Kluwer Academic Publishers, Norwell, MC, USA, 1997.
2. K. Jeffay and S. Goddard. A theory of rate-based execution. *in Proceedings of the 20th IEEE Real-Time Systems Symposium*, 1999.
3. K. Jeffay and D. Stanat. On non-preemptive scheduling of periodic and sporadic tasks. *in Proceedings of the 12th IEEE Real-Time Systems Symposium*, 1991.
4. J. P. Lehocky. Real-time queuing network theory. *in Proceedings of the 18th Real-Time System Symposium*, pages 58–67, December 1997.
5. S. L. Lehoczky, J. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. *in Proceedings of the IEEE Real-Time Systems Symposium*, December 1989.
6. T.-H. Lin and W. Tarng. Scheduling periodic and aperiodic tasks in hard real-time computing systems. *Performance Evaluation Review*, 19(1):31–38, May 1991.
7. C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of ACM*, 20(1), 1973.
8. N. Nissanke. *Real-time Systems*. Prentice Hall, Great Britain, 1997.
9. A. Papoulis. *Probability, Random Variables, and Stochastic processes*. McGraw-Hill, 1965.
10. E. Parzen. *Stochastic Processes*. Holden-day, 1962.
11. F. Reza. *An Introduction to Information Theory*. Dover Publications, NY, USA, 1994.
12. K. Schwan. Dynamic scheduling of hard real-time threads. *IEEE Transactions on Computers*, 18(8), 1992.
13. L. Sha. Real time scheduling theory: A historical perspective. *Journal of Real-Time Systems*, (28), 2004.
14. C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27, July 1948.
15. B. G. Spuri, M. Scheduling aperiodic tasks in dynamimc priority systems. *Journal of Real-Time Systems*, 2(10), 1996.
16. J. Stankovic. Real-time and embedded systems. *Journal of ACM Computing Surveys*, 29(1), March 1996.