

Un Modelo Dinámico para el Arribo de Tareas en Tiempo Real

Pedro Guevara López¹, J. J Medel², Flores Rueda Alberto³

Centro de Investigación en Computación

Instituto Politécnico Nacional

Av. Juan de Dios Batiz s/n C. P. 07738 México D. F.,

Tel. 57296000 ext. 56523.² guevara@pollux.cic.ipn.mx, ¹jmedel@pollux.cic.ipn.mx, ³aflores@pollux.cic.ipn.mx

Resumen

Para el estudio de Sistemas en Tiempo Real es necesario conocer el comportamiento de las tareas que lo forman; esto es para aprovechar al máximo los recursos mediante planificadores y conocer el comportamiento del sistema ante diversas situaciones (predictibilidad). Diversos autores han desarrollado modelos de Tareas en Tiempo Real (TTR), sin embargo estos modelos son estáticos y no brindan mucha información del arribo de las mismas. En este documento se propone un modelo dinámico general para tiempos de arribo absolutos basado en un Modelo Recursivo con Promedios Móviles (MRPM) que incluye perturbaciones internas y ajenas al procesador; a través de este modelo es posible representar el comportamiento de las tareas periódicas, esporádicas y aperiódicas.

Palabras clave: **Tiempo Real, tarea, jitter, tiempo de arribo, dinámico.**

1 Introducción

Una aplicación de Tiempo Real está compuesta por un conjunto de tareas concurrentes [2] que cooperan entre sí. Estas tareas son activadas a intervalos regulares o irregulares y deben completar su ejecución antes del término de plazos de respuesta especificados. En cada activación, la tarea lleva a cabo una actividad de cómputo con la que interactúa con su ambiente externo. Las tareas de Tiempo Real poseen requisitos temporales: plazos e intervalos de tiempo en que deben llevarse a cabo actividades concretas. Basándose en estos requisitos temporales, una tarea de Tiempo Real puede ser: a) Hard o Críticas: si el incumplimiento de un requisito temporal supone un fallo intolerable por sus consecuencias en el sistema [7], b) Soft o Acríticas: donde es posible tolerar el incumplimiento ocasional de un requisito temporal [7] o c) Firm: En caso de no respetar las restricciones de tiempo no sirve de nada el servicio que prestan. Las tareas de Tiempo Real se distinguen por contar con las siguientes características de tiempos: a) Tiempo de cálculo: éste se da por la implementación, b) periodo o frecuencia de ejecución: el que dicta la aplicación, y c) plazo de respuesta: que está definido por las características de la aplicación [2],[7].

La funcionalidad requerida en un sistema de Tiempo Real depende de las características de la aplicación. En consecuencia, las tareas de Tiempo Real pueden ser: a) Periódicas: que se ejecutan con una periodicidad fija o variable [6], b) Aperiódicas: las cuales se ejecutan en respuesta a un evento que ocurre en el sistema en instantes irregulares [2], c) Esporádicas: ocurren repetidamente, pero el intervalo de tiempo entre ocurrencias consecutivas varía y puede ser arbitrariamente largo [6],[12] y d) Semiperiódicas: Son tareas periódicas con tiempo de cálculo variables [17].

Las tareas periódicas están en relación a sistemas de adquisición de datos con tiempos de muestreo constantes; las tareas esporádicas están asociadas a eventos que ocurren repetidamente, pero los

intervalos de tiempo entre arribos son variables debido al medio ambiente o mundo físico [6]. Las tareas esporádicas se relacionan con eventos externos como alarmas o notificaciones.

2 Tipos de Tareas de acuerdo a su tiempo de arribo

Las Tareas Periódicas son TTR que se activan regularmente con *periodo* T_i y tiempo de cálculo c_i conocidos y constantes. La restricción principal es el *plazo* de ejecución (d_i) donde $d_i \leq T_i$. Las tareas periódicas son encontradas comúnmente en aplicaciones como aviones y procesos de control donde se requiere un continuo monitoreo y procesamiento de datos [6]. Las Tareas Aperiódicas son activadas irregularmente con un periodo desconocido (y tal vez no existente). La restricción de tiempo es generalmente el *plazo* d_i [2]. De acuerdo a [6], [8], [12], [14], las tareas esporádicas se caracteriza por un *tiempo de ejecución* C_i y un *mínimo tiempo de inter-arribo* T_i entre tiempos de activación. Las tareas esporádicas son asociadas con procesamiento de eventos que responden a entradas de dispositivos no periódicos; esos eventos ocurren repetidamente, pero el intervalo de tiempo entre ocurrencias consecutivas varía y puede ser arbitrariamente largo.

3 Modelos de Tareas en Tiempo Real

Parte importante de este trabajo es documentar las diferentes formas en que se hace mención a las Tareas en Tiempo Real, por tal motivo se presentan algunas consideraciones y modelos mas representativos. El documento [8] fue revolucionario en cuanto al cómputo en Tiempo Real dado que fueron los primeros en modelar los tiempos de arribo y los plazos de las tareas. Antes de éste se citó siempre a James Martín [9]. Por esta razón, aunque no es reciente su libro, es básico para saber cuales fueron sus avances y aportaciones. James Martín hace un análisis de los sistemas de cómputo para realizar tareas en tiempo real, sin embargo, aunque hace una breve descripción de lo que es un sistema de cómputo en Tiempo-Real, “Se puede definir un sistema de computador en Tiempo-Real como aquel que controla el medio a través de la recepción y proceso de datos y que actúa o devuelve los resultados con la suficiente rapidez para afectar el funcionamiento del medio en ese momento” no menciona como definir las restricciones de tiempo, no presenta ningún modelo de tareas en tiempo real. Menciona las causas por las que un sistema puede ser lento, tomando en cuenta desde los discos duros hasta los sistemas de comunicación con un modelo de petición de servicio usando teoría de colas y confiabilidad de sistemas.

3.1 Condiciones de Liu y Layland

Liu y Layland en [8] dan las bases para el modelado de STR partiendo de conceptos diferentes a los utilizados por Martín en [9]. En su artículo se hace una mención a las TTR en donde se asumen ciertas características en el ambiente. Las tareas en tiempo real se restringen a un conjunto de condiciones: 1) se considera que las tareas críticas son las periódicas con tiempos de arribo constantes entre solicitudes, 2) el plazo consiste solamente en la restricción de capacidad de ejecución, esto es, cada tarea debe ser completada antes de que ocurra la próxima solicitud. 3) las tareas son independientes en cuanto a que los tiempos de arribo para ciertas tareas no dependen de la iniciación o terminación de solicitudes de otras tareas, 4) el tiempo de ejecución de cada tarea es constante para esa tarea y no varía con el tiempo. Este tiempo de ejecución se refiere al tiempo que se tome el procesador para ejecutar la tarea sin interrupción, 5) cualquier tarea aperiódica en el

sistema es especial; estas son rutinas de inicialización o recuperación de fallas; éstas tareas desplazan tareas periódicas mientras ellas mismas son ejecutadas y no tienen restricciones de tiempo críticas o tipo “hard”.

Con la anterior se puede caracterizar una tarea por dos números: su tiempo de arribo y su tiempo de ejecución. Con $\tau_1, \tau_2, \dots, \tau_m$, denota m tareas periódicas, con sus periodos de tiempo T_1, T_2, \dots, T_m y sus tiempos de ejecución C_1, C_2, \dots, C_m respectivamente

3.2 El Modelo general de tareas de Mok y Chen

En este modelo [13] se define una tarea en tiempo real como la pareja (Φ, P) , donde Φ es un arreglo de tiempos de ejecución (ϕ_1, ϕ_2, \dots) , y P es el tiempo mínimo de separación, el plazo de cada periodo es P después de su tiempo de arribo. La desventaja de este modelo es que es necesario conocer los tiempos de ejecución para cada una de las tareas, pero se pueden tomar en cuenta cualquier tipo de tareas ya que se especifica explícitamente el tiempo de ejecución de cada una de ellas en cada instante, pero tiene el inconveniente de tener que llevar explícitos los tiempos de ejecución y por lo tanto su implementación es complicada. Este modelo difiere de el de Multiframe en que el tiempo de ejecución de las tareas varía arbitrariamente.

3.3 El modelo “Multiframe” de Mok y Chen

Mok y Chen en [13] proponen que las tareas en tiempo real tipo “multiframe” quedan definidas con la pareja (Γ, P) donde Γ es un arreglo de N tiempos de ejecución $(C^1, C^2, \dots, C^{N-1})$, para $1 \leq N$ y P es el tiempo mínimo de separación. El tiempo de ejecución del i -ésimo elemento es $C^{((i-1) \bmod N)}$, donde $1 \leq i$. El plazo para cada instancia es P después de su tiempo de arribo.

3.4 Caracterización de tareas esporádicas de Baruah, Mok y Rosier

En [3] se explica que un sistema de tareas esporádicas τ es una colección de tareas esporádicas $\{T_1, T_2, \dots\}$. Una tarea esporádica T_i está caracterizada por tres elementos: tiempo de ejecución e_i , un plazo crítico d_i , y un periodo mínimo de separación p_i , con $e_i \leq d_i$ y $e_i \leq p_i$, de tal forma que $T_i = (e_i, d_i, p_i)$, $i \leq n$. En este caso, las tareas se representan con tres valores. No presenta ninguna relación temporal con las otras tareas.

3.5 El modelo de tareas periódicas y esporádicas de Ramanathan y Kang

Ramanathan y Kang en [15] Define un modelo de tareas esporádicas, en donde los tiempos de ejecución tienen una variación estocástica y la función de distribución probabilística de la variación es conocida para el sistema. Además cada tarea esporádica es asociada con dos costos, K_e y K_l . K_e representa el costo del sistema en caso que la tarea sea rechazada tan pronto como llegue. K_l es el costo del sistema en caso que la tarea no cumpla con su plazo crítico después de ser aceptada por el sistema.

En este caso el costo para el sistema depende de cuando es rechazada la tarea esporádica. Las tareas son definidas por su tiempo plazo como (t, D_s) , donde “ t ” es el tiempo actual y D_s es el tiempo crítico. El tiempo crítico característico de la tarea $\tau \in P \cup S$, donde P es el conjunto de tareas periódicas del sistema y S es el conjunto de tareas esporádicas del sistema D_s . El tiempo crítico “deadline” se expresa como:

$$\Delta_{\tau}(t, D_s) = \begin{cases} \left(\left\lceil \frac{t}{P_{\tau}} \right\rceil + 1 \right) P_{\tau} & \text{Si } \tau \in P \\ D_{\tau} & \text{Si } \tau \in S \end{cases}$$

Donde el subíndice τ denota el tiempo crítico de la tarea. Un proceso esporádico es aceptado por el planificador según una función de selección. Esta función de selección analiza el tiempo disponible actual y verifica si es posible o no aceptar una tarea esporádica.

3.6 El modelo de tareas aperiódicas y esporádicas de Choi y Agrawala

Choi y Agrawala en [4] proponen que las tareas aperiódicas τ_i (del conjunto de tareas aperiódicas $T = \{ \tau_1, \tau_2, \dots, \tau_N \}$) se representan con lo siguiente

- Tiempo de arribo R_i
- Plazo absoluto D_i
- Peor tiempo de ejecución C_i
- Variable de ejecución e_i denotando el tiempo de procesamiento ya hecho para τ_i en cualquier instante de tiempo
- Variable de ejecución w_i denotando el último de comienzo o de τ_i , que es una función del tiempo actual “ t ” y el valor de e_i
- Tiempo de arribo menor $est(i)$
- Tiempo de arribo mayor $lst(i)$

Para las tareas esporádicas de tipo crítico se asume que el tiempo de inicio es igual al tiempo de arribo. Se considera que la planificación inicial de las instancias se da en una ventana de planificación $[0, L]$, denotada por Γ . $T = \{ \tau_1, \tau_2, \dots, \tau_N \}$ es un conjunto de instancias de tarea en Γ , donde τ_i aparece antes que τ_{i+1} en Γ . Se define un conjunto $S = \{ S_1, S_2, \dots, S_m \}$ como el conjunto de tareas esporádicas que tienen que ser planificadas con T . Para cada tarea esporádica S_i se asume que se conocen el tiempo mínimo de interarribo δ_i , el tiempo máximo de ejecución c_i^S y el deadline relativo d_i^S ($\leq \delta_i$). También se asume que las S_i s están ordenadas ascendentemente por su plazo relativo, d_i^S , ejem, $d_i^S \leq d_{i+1}^S$.

3.7 El modelo de tareas periódicas y esporádicas de Jeffay, Stanat y Martell

En [6] se expone que una tarea periódica se invoca con periodos regulares mientras que una tarea esporádica se invoca con periodos arbitrarios de tiempo pero este periodo tiene un valor mínimo conocido.

Una tarea T es una pareja (c,p) donde: c es el tiempo de ejecución máximo de terminación de la tarea, p es el período o intervalo mínimo entre invocaciones de T . Si T es periódico p especifica un intervalo constante entre invocaciones. Si T es esporádico p especifica el intervalo mínimo entre invocaciones.

Para las tareas periódicas si t_k es el momento de la k -ésima invocación de la tarea T entonces:

- La $(k+1)$ -ésima invocación ocurrirá en $t_{k+1}=t_k+p$.
- La k -ésima ejecución de la tarea T debe comenzar después de t_k y debe ser completada antes de t_k+p .

Para las tareas esporádicas, si t_k es el tiempo de la k -ésima de la tarea T entonces:

- La $(k+1)$ -ésima invocación ocurrirá no antes de t_k+p , por lo tanto $t_{k+1} \geq t_k+p$.
- La k -ésima ejecución de la tarea T debe comenzar después de t_k y debe ser completada antes de t_k+p .

Se asume que las invocaciones de las tareas esporádicas son independientes y dependen solamente de el momento de la última invocación.

3.8 Modelo para tareas aperiódicas de Spuri y Buttazzo

El modelo propuesto en [16] da una relación entre arribos contiguos de dos instancias en una TTR sin llegar a ser un modelo dinámico. Las consideraciones son:

- Todas las tareas τ_i : $i=1, \dots, n$ tienen plazos críticos;
- Todas las tareas *aperiódicas* J_i : $i=1, \dots, m$ no tienen plazos;
- Cada tarea periódica τ_i tiene un periodo constante T_i y un tiempo máximo de ejecución C_i , que se considera conocido y puede ser derivado por un análisis estático del código fuente;
- Todas las tareas periódicas son activada simultáneamente al tiempo $t=0$; por ejemplo, la primer instancia de cada tarea periódica tiene un tiempo de arribo $r_i(0)=0$;
- El tiempo de arribo de la k -ésima instancia periódica está dado por $r_i(k)=r_i(k-1)+T_i$;
- El plazo de la k -ésima instancia periódica está dado por $d_i(k)=r_i(k)+T_i$;
- El tiempo de arribo de cada tarea aperiódica es desconocido;
- El peor tiempo de ejecución de cada tarea aperiódica se considera conocido en su tiempo de arribo.

3.9 Modelo de tareas de Patricia Balbastre

Balbastre en [1] propone un modelo de tareas periódicas utilizado para el manejo de sistemas de control en tiempo real; divide al sistema en dos fases: la parte del control, definida por el algoritmo de control y su calidad de respuesta y la fase de la Tarea en Tiempo Real. El modelo de tareas que se plantea en esta tesis parte de la definición de actividad de control. La actividad de control son todas aquellas tareas o procesos relacionados con un mismo bucle de control.

El modelo de tareas propuesto consta de n actividades de control $A = (A_1, \dots, A_n)$. Una actividad de control está caracterizada como $A_i = (C_i, D_i, P_i, O_i)$. Cada actividad de control A_i da lugar a las siguientes tareas:

- T_{ii} Correspondiente a la adquisición de datos de la actividad A_i .
- T_{im} Correspondiente al cálculo de la acción de control de la actividad A_i .
- T_{if} Correspondiente al envío de la acción de control dentro de la actividad A_i .

Por tanto, el modelo de tareas completo consta de $3n$ tareas $T = (T_{1i}, T_{1m}, T_{1f}, \dots, T_{ni}, T_{nm}, T_{nf})$ caracterizadas como:

- $T_{ii} = (C_{ii}, D_{ii}, P_{ii}, O_{ii})$
- $T_{im} = (C_{im}, D_{im}, P_{im}, O_{im})$
- $T_{if} = (C_{if}, D_{if}, P_{if}, O_{if})$

Donde C es el tiempo de ejecución, D es el plazo relativo, P es el periodo y O es el desplazamiento o tiempo de inicio.

Este modelo es solo para tareas periódicas determinísticas, no considera el jitter y no es dinámico. Su principal utilidad está basada en la disminución de retardos y planificabilidad para sistemas de control en tiempo real.

3.10 Modelo dinámico de tareas de Guevara y Medel

Este modelo [6] se desarrollo para análisis dinámico de predictibilidad, consiste en el desarrollo de un algoritmo que brinde mayor información sobre el arribo y comportamiento de las TTR a través de un Filtro Digital en Tiempo Real (ver [10], [11]) que permita reconstruir, seguir y predecir el desempeño de cada tarea, todo esto con el fin de hacer estudios de y confiabilidad de los STR y obtener mejores algoritmos de planificación en línea [2]. El modelo propuesto es:

$$\begin{aligned}x_{k+1} &= a_k x_k + \omega_k^1 \\ y_k &= x_k + \omega_k^2\end{aligned}$$

donde los ruidos ω_k^1 y ω_k^2 corresponden al jitter y a perturbaciones externas al procesador, no están correlacionados entre si, pero si con la señal generadora de tareas y_k . Las dinámicas internas descritas por x_k permiten a través del parámetro a_k y del ruido interno ω_k^1 , modelar las TTR.

4 Modelo General para Tiempos de Arribo en Tareas de Tiempo Real

Como respuesta a los problemas expuestos en las secciones anteriores, en este trabajo se presenta un Modelo General para Tiempos de Arribo en Tareas de Tiempo Real basado en un Modelo Recursivo con Promedios Móviles (MRPM) que caracteriza a las TTR: a) periódicas, b) aperiódicas y c) esporádicas. Para tal fin se propone que el un sistema lineal, de primer orden, variante en el tiempo y no estacionario considerando que el jitter y las perturbaciones externas al procesador no están correlacionados y obedecen a una función de distribución normal. El modelo propuesto es

monovariante, esto quiere decir que solo se caracteriza el tiempo de arribo de una tarea. Se considera que los tiempos de ejecución máximos; los plazos mínimo y máximos son conocidos; las tareas son críticas; el jitter está acotado dentro de un intervalo conocido.

Este modelo servirá para desarrollar un algoritmo que brinde mayor información sobre el arribo y comportamiento de las TTR a través de un Filtro Digital en Tiempo Real (ver [10], [11]) que permita reconstruir, seguir y predecir el desempeño de cada tarea, todo esto con el fin de hacer estudios de predictibilidad y confiabilidad de los STR y obtener mejores algoritmos de planificación en línea [2].

Definición 1. (Tiempo de arribo de una Tarea de Tiempo Real). El tiempo de arribo absoluto de una Tarea de Tiempo Real (TTR) está dado por:

$$l_k := l_{k-1} + \pi_k \quad (1)$$

$$\pi_k := a_k \pi_{k-1} + \Sigma_k \quad (2)$$

$$\Sigma_k = u_k + v_k + w_k - a_k w_{k-1} \quad (3)$$

Donde:

- l_k es el tiempo de arribo absoluto,
- π_k es el tiempo de interarribo,
- a_k es el parámetro del sistema
- v_k es el jitter, es una variable aleatoria y se considera una perturbación interna (ver [6]).
- w_k son las perturbaciones externas al procesador, es una variable aleatoria,
- u_k es el tiempo de interarribo de referencia.
- Σ_k son las perturbaciones y entrada del sistema.

l_k es creciente a l_n y π_k está acotado por sus momentos de probabilidad.

Caso 1. (Tarea Periódica). El tiempo de arribo de una TTR periódica toma el modelo de la Propuesta 1 con las siguientes consideraciones:

- $a_k < 1$ e invariante para todo k ,
- u_k es el tiempo de interarribo ponderado, es constante para todo k ,
- $|E(v_k)| \leq |E(w_k)|$

Caso 2. (Tarea esporádica). El tiempo de arribo de una TTR esporádica toma el modelo de la Propuesta 1 con las siguientes consideraciones:

- $a_k = f(k)$, $f(k)$ es una función continua y periódica
- u_k es el periodo (tiempo de arribo ponderado), es constante para todo k ,
- $|E(v_k)| < |E(w_k)|$

Caso 3 . (Tarea aperiódica). El tiempo de arribo de una TTR esporádica toma el modelo de la Propuesta 1 con las siguientes consideraciones:

- a_k es una variable aleatoria,

u_k es una variable aleatoria,
 $|E(v_k)| \ll |E(w_k)|$

5 Resultado de las simulaciones del modelo general

Las simulaciones del modelo para cada tipo de tarea fueron realizadas en línea utilizando Matlab 5.0. Se graficaron los resultados de las ecuaciones (1), (2) y (3). La entrada al sistema (u_k) fue el periodo, el tiempo mínimo de interarribo y una variable aleatoria para las tareas periódica, esporádica y aperiódica;

Para la tarea periódica se consideró $u_k=3$, $v_k= \pm 0.03$ s y $w_k= \pm 0.02$ s y $a_k= 0.3$, ver Fig. 1. Para la tarea esporádica se consideró $u_k= \sin(k)$, $a_k= 0.3$, $v_k= \pm 0.03$ s y $w_k= \pm 0.02$ s, ver Fig. 2. Para la tarea aperiódica se consideró u_k como una variable aleatoria con valor ± 5.0 , $a_k= \pm 0.6$, $v_k= \pm 0.1$ s y $w_k= \pm 0.15$ s, ver Fig. 3.

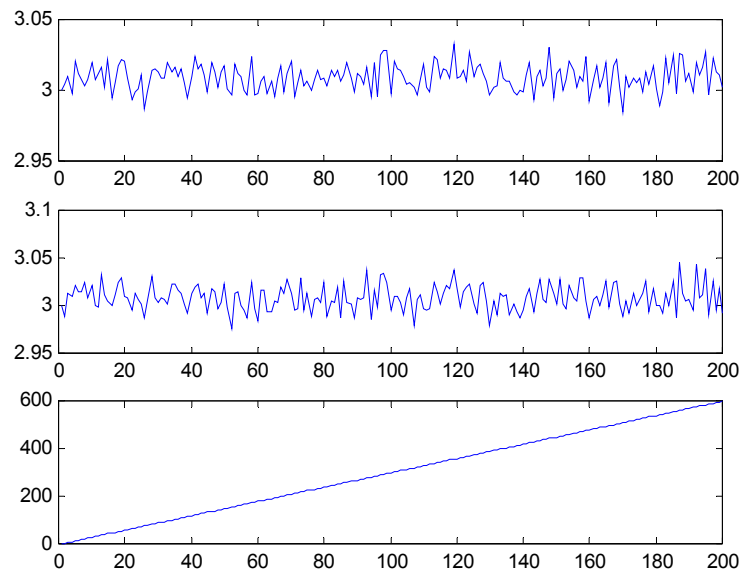


Fig. 1. Comportamiento de una tarea periódica con periodo 3.

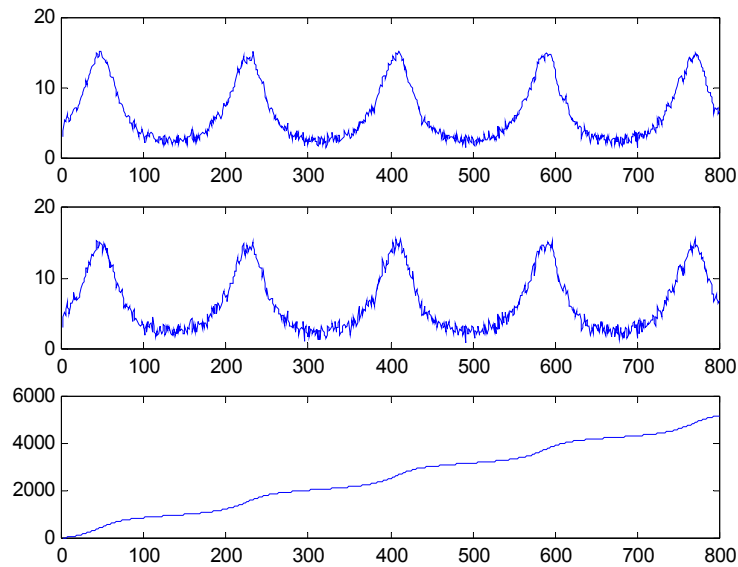


Fig. 2. Comportamiento de una tarea esporádica con tiempo mínimo de interarribo 3.

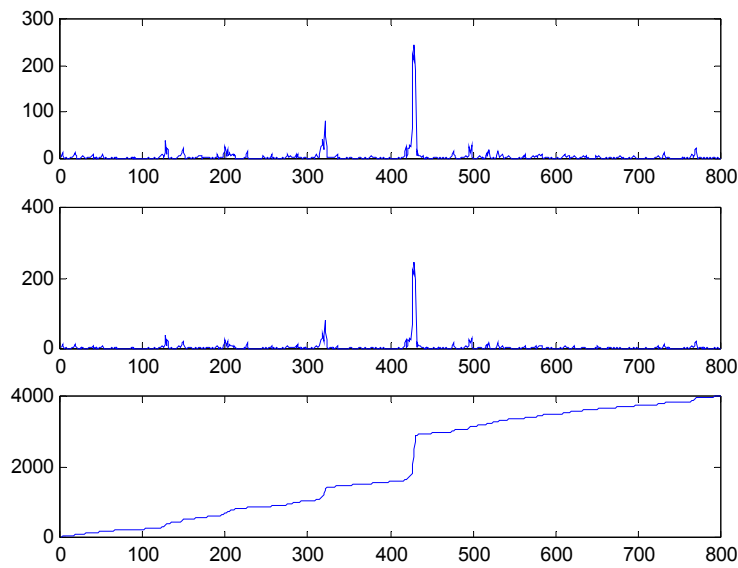


Fig. 3. Comportamiento de una tarea aperiódica

6 Conclusiones

En este trabajo se presentó un modelo general para tiempos de arribo en tareas de tiempo real; a través de éste es posible simular el comportamiento de tareas periódicas, esporádicas y aperiódicas modificando el valor del parámetro del sistema y el valor de su entrada. A diferencia de otros modelos, el modelo propuesto es dinámico describe el comportamiento en línea de una tarea; de

ésta manera la salida obtenida puede aprovecharse para planificación dinámica y análisis de predictibilidad.

7 Bibliografía

- [1] Balbastre P. (2002). *Modelo de tareas para la integración del control y la planificación en sistemas de tiempo real*. Tesis Doctoral, Universidad Politécnica de Valencia, Departamento de Informática de Sistemas y Computadores, España.
- [2] Buttazzo G. (1997) *Hard real-time computing systems*. Scuola Superiore S. Anna, Kluwer Academic Publishers.
- [3] Baruah, S., Aloysius Mok, and Louis Rosier. (1990). *Preemptively scheduling hard-real-time sporadic tasks on one processor*. Proceedings of the Real-Time Systems Symposium, pp 182-190, Orlando, Florida. December 1990. IEEE Computer Society Press.
- [4] Choi s., Agrawala A (1997). *Scheduling aperiodic and sporadic tasks in hard real-time systems*. Technical report University of Maryland
- [5] Guevara P, Medel J. J. (2002). *Modelo ARMA para caracterización de tareas en tiempo real*. Taller Internacional de Instrumentación Virtual CIC-INDI 2002, Pachuca agosto de 2002.
- [6] Jeffay K., Stanat D., Martel C. (1991). *On non-preemptive scheduling of periodic and sporadic tasks*. Proceedings of the Twelfth IEEE Real-Time Systems Symposium, San Antonio Texas.
- [7] Liu C., Layland J. (1973). *Scheduling algorithms for multiprogramming in hard-real-time environment*. Journal of the ACM, Vol. 20, No. 4, (1982), pp273-250.
- [8] Liu J. "Real-time Systems" Ed. Prentice Hall USA 2000.
- [9] Martin J. (1980). *Diseño de sistemas de computadores en Tiempo Real*. Ed. Diana 1980.
- [10] Medel J.J., Guevara P. (2002). *Caracterización de Filtros Digitales en Tiempo Real para Computadoras Digitales*. Sometido a revisión en la Revista Computación y sistemas, mayo de 2002, México D.F.
- [11] Medel J.J., Guevara P. (2002). *Análisis Restrictivo para Filtros Digitales en Tiempo Real Presentando un Ejemplo para un SLIT tipo SISO*. IBERAMIA 2002, Sevilla España.
- [12] Migge J. (1999). *Theoretical background*. Sophia Antipolis. <http://www.sop.inria.fr/mistral/personnel/Jorn.Migge/manual/node1.html>
- [13] Mok A., Chen D. (1997). *A general model for real-time tasks*. Technical report, University of Texas at Austin.
- [14] Rajkumar R. (1991). *Synchronization in real-time systems, a priority inheritance approach*. Kluwer Academic Publishers.
- [15] Ramanathan P., Kang D. (1994). *A generalized guarantee model for servicing sporadic tasks with firm deadlines*. Real-time Systems Journal May 1994.
- [16] Spuri M. & Buttazzo G. (1996). *Scheduling aperiodic tasks in dynamic priority systems*. Journal of real-time systems.
- [17] Tia T., Deng Z., Shankar M., Storch M., Sun J., Wu L., Liu J. (1995). *Probabilistic performance guarantee for real-time tasks with varying computation times*. IEEE Real-time technology applications symposium.