

Modelo Dinámico para una Tarea en Tiempo Real

Dynamical Model for a Real-Time Task

Pedro Guevara López¹, J. J. Medel², Daniel Cruz Pérez³

¹Centro de Investigación en Tecnologías de Información y Sistemas - UAEH
Carretera Pachuca-Tulancingo km. 4.5 Ciudad Universitaria, México Hidalgo.
pguevara@df1.telmex.net.mx

²Centro de Investigación en Innovación Tecnológica - IPN
Cerrada CECATI s/n. Col. Santa Catarina C.P. 02250 Azcapotzalco, D.F.
jjmedel@pollux.cic.ipn.mx

³Centro de Investigación en Computación – IPN
Av. Juan de Dios Bátiz sn esq. Miguel Othón de Mendizábal C. P. 07738 México D. F.,
dcruzp@yahoo.com

Artículo recibido en enero 21, 2004; aceptado en agosto 09, 2004

Resumen

Las técnicas de planificación estáticas [1] parten de la hipótesis de que el conjunto de tiempos de arribo y de ejecución de las tareas se conocen a priori, mientras que las técnicas dinámicas [1] generalmente se basan en servidores de planificación; sin embargo, en ambos casos se desconoce el comportamiento real de la tarea y los arribos de sus respectivas instancias. Si se desconoce esta dinámica de las tareas no es posible hablar de predecibilidad. En este documento se propone un modelo dinámico general para tiempos de arribo, tiempos de ejecución y determinación de plazos basado en sistemas ARMA que incluyen el jitter y perturbaciones ajenas al procesador [3] y [13]. A través de éste modelo es posible definir el comportamiento de las tareas periódicas, esporádicas, y aperiódicas en un sentido de probabilidad.

Palabras Clave: Arribo, ejecución, modelo, plazo, Tiempo Real.

Abstract

Static scheduling techniques [1] are based on the hypothesis that the arrival times set and tasks execution times are known previously; while dynamical techniques [1] are generally based on scheduling servers. However, in both cases the real-time task behavior and their arrival times are unknown. It is impossible to talk about predictability without knowing those task dynamics. In this paper, we propose a dynamical model for arrival times and execution times, both based on ARMA models that include jitter and external perturbations of processor, in consequence deadlines are determined. With this model it is possible to know periodic, sporadic and aperiodic tasks behavior in a probability sense.

Keywords: Arrival, computation, model, deadline, Real-time

1 Introducción

Las TTR Periódicas son aquellas que se activan regularmente con *periodo* T_i y tiempo de cálculo c_i , *ambos* 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 monitoreo uniforme así como un procesamiento de datos con restricciones de tiempo y con respuestas de excelente calidad (ver: [1] y [2]).

Las Tareas aperiódicas son activadas irregularmente con un periodo desconocido. La restricción de tiempo es generalmente el *plazo* d_i [1].

Las tareas Esporádicas, de acuerdo a [3], [4], [5], se caracterizan por un *tiempo de ejecución* C_i y un *tiempo mínimo 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.

Liu y Layland en [4] dan las bases para el modelado de STR partiendo de conceptos diferentes a los utilizados por Martín en [6]. 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 no-periódica 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".

2 Modelos de Tareas en Tiempo Real

En [4] se presenta un modelo sobre los tiempos de arribo y los plazos de las tareas, tomando como base el trabajo desarrollado en [6]. En este último se presenta un análisis de los sistemas de cómputo para realizar TTR; sin embargo, aunque hace una breve descripción de lo que es un sistema de cómputo en TR, (Se puede definir un sistema de computador en TR 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 se menciona como definir las restricciones de tiempo y no presenta ningún modelo que describa a este tipo de tareas en TR. En el mismo documento se mencionan las causas por las que un sistema de esta clase 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.

En [7] 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. También se propone un modelo "multicadros" donde las tareas en tiempo real tipo quedan definidas con la pareja (I, P) donde I 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. Por desgracia, estos modelos son determinísticos y estáticos y se deben conocer de antemano todos los tiempos de cada instancia y cada tarea.

En [8] se 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. El autor hace un análisis fuera de línea y estático para planificar las tareas; cada tarea esporádica es asociada con dos costos, K_e y K_i . El primero es el costo que implica no aceptar la tarea y el segundo es el costo que implica que la tarea no cumpla con su plazo.

En [9] los autores proponen que las tareas aperiódicas τ_i (del conjunto de tareas aperiódicas $T = \{ \tau_1, \tau_2, \dots, \tau_N \}$) se representan con:

- 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 define un conjunto $S = \{S_1, S_2, \dots, S_m\}$ como el conjunto de tareas esporádicas que tienen que ser planificadas dentro de T . Este modelo tampoco es dinámico y es usado para planificación.

En [3] se dice 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 arribo de la k -ésima instancia de 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 dependen solamente de la instancia inmediata anterior.

En [10] se explica que un sistema de tareas esporádicas τ es una colección de tareas $\{T_1, T_2, \dots\}$. 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.

Para el modelo propuesto en [11] se aclara que los tiempos de arribo y plazos son arbitrarios. El tiempo de arribo de una tarea T_i se denota con A_i , su tiempo de ejecución (posiblemente desconocido) se denota con C_i y su tiempo de respuesta máximo deseado se denota con D_i . Una tarea cumple con su plazo si termina antes de $A_i + D_i$. En el documento $A_i + D_i$ representa el plazo absoluto de la tarea y D_i su plazo relativo. El modelo no es general ya que solo toma en cuenta las tareas aperiódicas y además no propone un patrón para el arribo de tareas.

El modelo propuesto en [12] 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 activadas simultáneamente al tiempo $t=0$; por ejemplo, la primera 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.

El modelo propuesto en [13] (un primer intento por parte de los autores de este artículo) se desarrolló para el análisis dinámico de predecibilidad. Consiste en el desarrollo de un algoritmo que brinde mayor información sobre el arribo y comportamiento de las TTR a través de un FDTR (Filtro Digital en Tiempo Real (ver por ejemplo: [2] y [14])) que permita reconstruir, seguir y predecir el desempeño de cada tarea, todo esto con el fin de hacer estudios de confiabilidad de los STR y obtener mejores algoritmos de planificación en línea [1].

En [15] se introduce un modelo de planificación para Tareas en Tiempo Real llamado IRIS (Increasing Reward with Increasing Service). Este modelo asocia un valor de "recompensa" a cada instancia según el tiempo que se le conceda de ejecución. Entre más tiempo se le conceda, más preciso es el cálculo. El valor de la "recompensa" está asociado a una función no decreciente y cóncava.

3 Modelo Dinámico para una TTR

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 tipo ARMA (modelo auto-regresivo con promedios móviles) 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 monovariable, 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.

3.1 Definiciones Preliminares

Para la presentación del modelo de tiempos de arribo es necesario dar un conjunto de definiciones básicas que servirán de punto de partida:

Definición 1. (Tarea) Una tarea en Tiempo Real es una entidad ejecutable J_i que al menos es caracterizada por un tiempo de arribo y una restricción temporal. Está formada por un conjunto de instancias j_k , tal que $J_i = \{j_k\}$ con $i, k \in \mathbf{Z}^+$.

Definición 2. (Instancia) Una instancia j_k es una unidad de trabajo de una tarea J_i , queda definida por $j_{i,k} = (l_k, c_k, d_{k_max})$ para $i, k \in \mathbf{Z}^+$ donde l_k es el tiempo de arribo de cada instancia, " c_k " es el tiempo de ejecución de cada instancia y d_{k_max} es el plazo máximo absoluto de cada instancia

Definición 3. (Tiempo de arribo absoluto de una TTR). El tiempo de arribo absoluto l_k de una instancia con índice k de una TTR J_i está definido como el tiempo en que la instancia pide atención al procesador en relación al origen temporal de referencia (ver Fig. 1).

Definición 4. (Tiempo de inter-arribo de una TTR). El tiempo de inter-arribo o tiempo de arribo relativo π_k de una instancia con índice k de una TTR J_i está definido como el tiempo en que la instancia pide atención al procesador con relación al arribo absoluto de la instancia con índice $k-1$. De acuerdo la definición 1 y a [2], [16] se tiene que: $\pi_k = \mu[l_k, l_{k-1}]$. El tiempo de inter-arribo puede estar relacionado con la dinámica del sistema, por ejemplo, en el caso que se desee muestrear las variables de un sistema, los tiempos de inter-arribo se relacionan con el criterio de Nyquist en [17] (ver Fig. 1).

Definición 5. (Fase de una TTR). La fase ϕ_i de una TTR está definida como el tiempo de arribo absoluto o el tiempo de interarribo de su primera instancia, esto es: $\phi_i = l_1 = \pi_1$ (ver Fig. 1).

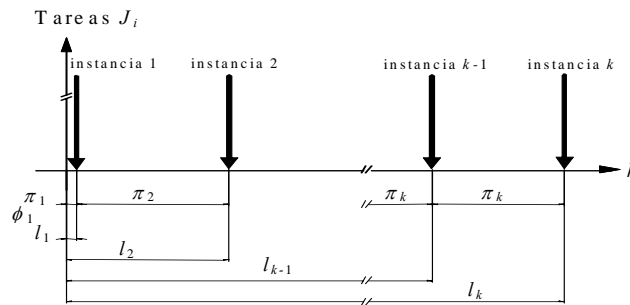


Fig. 1. Tiempos de arribo relativos π_k y absolutos l_k para varias instancias de una tarea J_i

Definición 6. (Tiempo de inicio de una TTR). El tiempo de inicio s_k de una instancia con índice k de una TTR J_i está definido como el momento en que la instancia es atendida por el procesador con relación al origen temporal de referencia. Esto es:

$$s_k = l_k + v_k \quad (1)$$

donde v_k es una variable aleatoria positiva denominada “jitter” y que depende del software y hardware de la computadora.

Definición 7. (Tiempo de ejecución de una TTR). El tiempo de ejecución c_k es el tiempo en que la instancia con índice k de una TTR J_i concluye sus operaciones sin considerar los desalojos de ésta en el procesador.

Definición 8. (Tiempo de finalizado). El tiempo de finalizado f_k es el tiempo en que la instancia con índice k de una TTR J_i termina su ejecución.

Definición 9. (Plazo absoluto mínimo de una TTR). El plazo absoluto mínimo d_{k_min} es el límite temporal inferior en relación al origen temporal de referencia después del cual la instancia con índice k de una TTR J_i debe finalizar.

Definición 10. (Plazo absoluto máximo de una TTR). El plazo absoluto máximo d_{k_max} es el límite de tiempo superior en relación al origen temporal de referencia, antes o cuando la instancia con índice k de una TTR J_i debe finalizar.

Definición 11. (Plazo relativo mínimo de una TTR). El plazo relativo mínimo D_{k_min} es el límite temporal inferior en relación al arribo absoluto l_k después del cual la instancia con índice k de una TTR J_i debe finalizar.

Definición 12. (Plazo relativo máximo de una TTR). El plazo relativo máximo D_{k_max} es el límite temporal superior en relación al arribo absoluto l_k antes o cuando la instancia con índice k de una TTR J_i debe finalizar.

3.2 Modelo para tiempos de arribo en TTR

En la mayoría de los artículos revisados, la referencia a los tiempos de arribo se asumen como correspondiente a TTR periódicas, esporádicas y aperiódicas y solamente en algunos de ellos se menciona que las TTR esporádicas tienen un arribo correspondiente a una función. A excepción de [18] en donde se usa un sistema de simulación de carga para servidores web (Aun que el artículo habla de QoS y no de Tiempo Real) no se tiene referencia al sistema que se planifica o cual es su comportamiento en cuanto a Tiempos de Arribo. En [19] comentan que las tareas esporádicas pueden representar sistemas de multimedia como videoconferencia y sistemas que necesitan controlar dispositivos asíncronos pero no ahondan en el tema, Adas menciona en [20] que los modelos regresivos (AR) se usan para modelar tasas de bits de salida de codificadores VBR (Variable Bit Rate), tráfico de video VBR y escenas de video MPEG. Comenta que los modelos ARMA se han usado para modelar tráfico VBR.

El modelo está representado por una ecuación de diferencias del tipo estocástica, con las siguientes propiedades: de primer orden y primero grado, variante en el tiempo y no estacionario; considerando que las perturbaciones externas al procesador no están correlacionados y que obedecen a una función de distribución normal. El modelo propuesto es monovariable: solo se caracteriza el tiempo de arribo de una tarea de acuerdo a [13].

Este modelo brindará mayor información sobre el arribo y tiempo de ejecución de las TTR a través de un Filtro Digital en Tiempo Real (ver: [2], [14]) que permita reconstruir, seguir y predecir el desempeño de cada tarea; todo esto con el fin de hacer estudios de predecibilidad y confiabilidad de los STR y obtener mejores algoritmos de planificación en línea de acuerdo a [1].

Proposición 1. (Tiempo de arribo absoluto de una TTR). El tiempo de arribo absoluto l_k de una instancia con índice k de una TTR J_i está descrito en forma recursiva por:

$$l_k = l_{k-1} + \pi_k \quad (2)$$

Prueba. De acuerdo a la Definición 5 (fase ϕ) $l_1 = \pi_1$ y $l_2 = \pi_1 + \pi_2$; entonces:

$$l_k = \sum_{j=1}^k \pi_j \quad (3)$$

$$l_{k-1} = \sum_{j=1}^{k-1} \pi_j \quad (4)$$

$$l_k = \sum_{j=1}^{k-1} \pi_j + \pi_k \quad (5)$$

Sustituyendo (4) en (5) se obtiene (2); por lo tanto la proposición se cumple.

La Fig. 2. Muestra la representación a bloques de l_k .

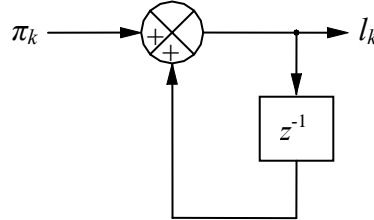


Fig. 2. Diagrama de bloques del modelo de tiempos de arribo absoluto en función del tiempo de arribo relativo.

Proposición 2. (Tiempo de inter-arribo en una TTR). El tiempo de inter-arribo π_k de una instancia con índice k de una TTR J_i está definido por:

$$\pi_k = a_k (\pi_{k-1} - w_{k-1}) + u_k + w_k. \quad (6)$$

Donde:

a_k es el parámetro del sistema,

π_{k-1} Tiempo de inter-arribo de la instancia con índice $k-1$,

w_k son las perturbaciones externas al procesador, representadas a través de una variable aleatoria con distribución gaussiana,

u_k es el tiempo de inter-arribo de referencia.

Prueba. Se parte de la hipótesis de que el tiempo de inter-arribo obedece a un sistema lineal, invariante en el tiempo, estacionario y con perturbaciones gaussianas. Entonces se tienen dos ecuaciones, la de estado (7) y la señal observable (8):

$$\pi_k = x_k + w_k, \quad (7)$$

$$x_k = a_k x_{k-1} + u_k. \quad (8)$$

La ecuación (8) no incluye perturbaciones internas debido a que el arribo de las tareas depende del entorno físico y no de la computadora.

Sustituyendo (8) en (7):

$$\pi_k = a_k x_{k-1} + u_k + w_k, \quad (9)$$

de (7) al retrasar una unidad de tiempo:

$$x_k = \pi_k - w_k, \quad (10)$$

sustituyendo en (10) en (9):

$$\pi_k = a_k (\pi_{k-1} - w_{k-1}) + u_k + w_k. \quad (6)$$

La proposición está probada.

En la Fig. 3 se muestra en diagrama a bloques del modelo del tiempo de inter-arribo π_k y la ecuación (6) se describe en la Fig. 4.

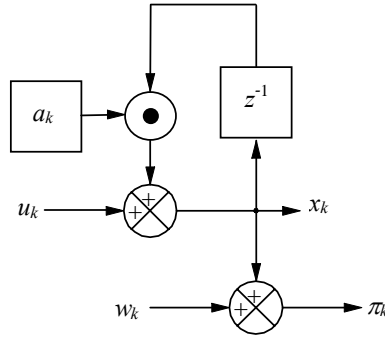


Fig. 3. Diagrama de bloques del modelo de tiempos de arribo a través de sus ecuaciones de estado y señal observable

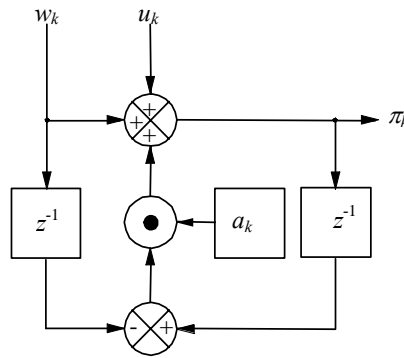


Fig. 4. Diagrama de bloques del modelol de tiempos de arribo expresado solo por sus entradas y su señal observable

Proposición 3. (Tiempo de inicio de una TTR). El tiempo de inicio s_k de una instancia realizable con índice k de una TTR J_i está descrito por:

$$s_k = s_{k-1} + \pi_k + \delta_k, \quad (11)$$

Prueba. De las ecuaciones (1) y (2) se tiene:

$$s_k = l_{k-1} + \pi_k + v_k. \quad (12)$$

Donde v_k es el jitter de la instancia con índice k . De (1), despejando y retrasando una unidad de tiempo a l_k :

$$l_{k-1} = s_{k-1} - v_{k-1}. \quad (13)$$

Sustituyendo (13) en (12):

$$s_k = s_{k-1} + \pi_k + v_k - v_{k-1}. \quad (14)$$

Si $\delta_k := v_k - v_{k-1}$ es sustituido en (14), la proposición está demostrada.

Comentario 1. Si se desea modelar el tiempo de inter-inicio π'_k bastaría agregar el jitter v_k como perturbación interna en la ecuación de estado en (8) tal que:

$$\pi'_k = x'_k + w_k \quad (14)$$

$$x'_k = a_k x'_{k-1} + u_k + v_k \quad (15)$$

Y se obtiene la ecuación (16) siguiendo el mismo procedimiento de la prueba de la Proposición 2:

$$\pi'_k = a_k (\pi'_{k-1} - w_{k-1}) + u_k + w_k + v_k \quad (16)$$

3.3 Caracterización de los diferentes tipos de las TTR según su tiempo de arribo

El modelado de los diferentes tipos de TTR de acuerdo a las ecuaciones (2) y (6) se basa en la manipulación del parámetro a y la entrada al sistema u_k . Para los ejercicios de simulación se utilizó Matlab 5.0 y se presentan las gráficas correspondientes.

Definición 13. (Tarea en Tiempo Real Periódica). Una TTRP es aquella tarea J_i donde todas sus instancias tienen tiempos de inter-arribo π_k próximos a un periodo T y los tiempos de arribo absolutos l_k crecen en forma monotónica respecto a la aparición de las instancias.

Parámetro a_k	$0 < a_k < 1$	a_k constante $\forall k$
Entrada u_k	$u_k = T - a_k T$	u_k constante $\forall k$
Otra característica	$0 < T$	Periodo $T \forall k$

Definición 14. (Tarea Esporádica en Tiempo Real). Una TTRE es aquella tarea J_i donde todas sus instancias tienen tiempos de inter-arribo π_k activados en forma irregular relacionados a una función de repetición con un tiempo ínfimo de inter-arribo π_{inf} y los tiempos de arribo absolutos l_k crecen en forma oscilante y acotada respecto a la aparición de las instancias.

Parámetro a_k	$0 < a_k < 1$	a_k es una función de repetición
Entrada u_k	$u_k = \pi_{inf}$	u_k constante $\forall k$,
Otra característica	$0 < \pi_{inf} < 0$	Tiempo ínfimo de inter-arribo

Definición 15. (Tarea en Tiempo Real Aperiódica). Una TTRA es aquella tarea J_i donde todas sus instancias tienen tiempos de inter-arribo π_k activados en forma irregular, tiene un tiempo mínimo de inter-arribo π_{min} y los tiempos de arribo absolutos l_k crecen en forma aleatoria respecto a la aparición de las instancias.

Parámetro a_k	$0 < a_k$	a_k es una variable aleatoria
Entrada u_k	$u_k = \pi_{min}$	u_k constante $\forall k$,
Otra característica	$0 < \pi_{min}$	Tiempo mínimo de inter-arribo

3.4 Modelo para tiempos de ejecución en una TTR

Por razones de Hardware existen ciertas variaciones en los tiempos de atención este dato es considerado en los STR con le *jitter*. Los tiempos de ejecución tienen también variaciones de forma tal que la mayoría de los autores consideran los peores casos de tiempo de ejecución (WCET). En [21] Lehoczky menciona que los sistemas están considerados para no necesitar una solución estocástica, de tal forma que consideran los WCET. Otros autores que toman en cuenta variaciones de los tiempos de ejecución como distribuidos probabilísticamente son T.S. Tia, Z. Deng, M. Storch, J. Sun, L.C. Wu, J.W.S. Liu. en [22]. Con las referencias anteriores, el desarrollo de un descriptor de tiempos de ejecución es de mucha importancia para futuros desarrollos como planificadores, seguidores de señales, simuladores de sistemas, etc.

El modelo está representado por una ecuación de diferencias del tipo estocástica, con las siguientes propiedades: de primer orden y primero grado, variante en el tiempo y estacionario; considerando que las perturbaciones externas al procesador no están correlacionados y que obedecen a una función de distribución normal. El modelo propuesto es monovariante: solo se caracteriza el tiempo de ejecución de una tarea para todas sus instancias. Para el desarrollo del modelo de los tiempos de ejecución se parte de la proposición 2, la ecuación 6 y se considera un tiempo de ejecución normalizado, a manera de una tarea periódica (definición 14). De ésta manera se obtiene la proposición 4.

Proposición 4. (Tiempo de ejecución de una TTR). El tiempo de ejecución c_k de una instancia con índice k de una TTR J_i está definido por:

$$c_k = b_k c_{k-1} + h_k + z_k \quad (17)$$

Donde:

b_k es el parámetro del sistema,

c_{k-1} Tiempo de ejecución de la instancia con índice $k-1$,

z_k son las perturbaciones internas del procesador, representadas a través de una variable aleatoria con distribución gaussiana,
 h_k es el tiempo de ejecución de referencia

Prueba. Se considera que no hay perturbaciones externas, esto es porque la computadora es la encargada de ejecutar todos sus cálculos. Partiendo de un sistema de dos ecuaciones de primer orden y lineales:

$$c_k = x_k \quad (18)$$

$$x_k = b_k x_{k-1} + h_k + z_k \quad (19)$$

Sustituyendo (19) en (18)

$$c_k = b_k c_{k-1} + h_k + z_k. \quad (17)$$

La proposición está demostrada.

3.5 Caracterización de plazos relativos y absolutos para una TTR

En las Definiciones 9 a la 12 se explica el concepto de plazo; en la práctica los plazos de las tareas se pueden determinar de varias maneras: mediante las variaciones del proceso, restricciones de los sistemas de adquisición, restricciones de los actuadores, anchos de banda de la red de comunicaciones, etc. Para el presente trabajo se consideran solamente los concernientes a la dinámica del sistema con el que se interactúa.

Proposición 5. (Plazo absoluto máximo de una TTR). El plazo absoluto d_{k_max} de una instancia k de la tarea J_i está acotado por el tiempo de arribo absoluto de la instancia con índice $k+1$, esto es:

$$l_k \leq d_{k_max} < l_{k+1}. \quad (20)$$

Prueba. Por inducción

Para $m=1$

$$l_1 \leq d_{1_max} < l_{1+1}. \quad (21)$$

Para $m=2$

$$l_2 \leq d_{2_max} < l_{2+1}. \quad (22)$$

Para $m=k-1$

$$l_{k-1} \leq d_{k-1_max} < l_k. \quad (23)$$

Para $m=k$ y del Proposición 1.

$$l_k = l_{k-1} + \pi_k, \quad (24)$$

$$l_{k+1} = l_k + \pi_{k+1}, \quad (25)$$

$$l_{k-1} + \pi_k \leq d_{k_max} < l_k + \pi_{k+1}. \quad (26)$$

De la proposición 1

$$l_k \leq d_{k_max} < l_k + \pi_{k+1}, \quad (27)$$

De acuerdo al Proposición 1, de (27) se llega a

$$l_k \leq d_{k_max} < l_{k+1}. \quad (20)$$

La proposición está demostrado.

Proposición 6. (Plazo relativo de una TTR). El plazo relativo D_{k_max} de una instancia k de la tarea J_i está acotado por el tiempo de arribo absoluto de la instancia con índice $k+1$, esto es:

$$0 \leq D_{k_max} < \pi_{k+1}. \quad (28)$$

Prueba. Restando l_k a cada miembro de la ecuación (27) tenemos

$$l_k - l_k \leq d_{k_max} - l_k < l_k + \pi_{k+1} - l_k, \quad (29)$$

$$0 \leq D_{k_max} < \pi_{k+1}. \quad (28)$$

La proposición está demostrada.

3.6 Generalización del Modelo para una TTR

De acuerdo a las Definiciones 1 y 2, una TTR está formada por un conjunto de instancias, esto es:

$$J_i = \{j_{i,k}\} \text{ y } j_{i,k} = (l_{i,k}, c_{i,k}, d_{i,k}), \quad (30)$$

$$J_{i,k} = \{(l_{i,k}, c_{i,k}, d_{i,k})\}, \quad (31)$$

Con

$$l_k = l_{k-1} + \pi_k, \quad (2)$$

$$c_k = b_k c_{k-1} + h_k + z_k, \quad (17)$$

$$l_k \leq d_{k_max} < l_{k+1}, \quad \forall i \in Z^+. \quad (20)$$

4. Resultados Obtenidos

En base a las ecuaciones (2), (6) y (17). Las simulaciones fueron realizadas en MatLab ver. 5.0 (los valores mostrados son UT “unidades de tiempo”), y sus resultados ilustrativos con: tarea periódica ($T=5$ UT, $w_k = \pm 0.1$ UT, $a_k = 0.01$), esporádica ($\pi_{min} = 4$ UT, $w_k = \pm 0.8$ UT, $a_k = \sin(k)$) y aperiódica ($\pi_{min} = 3$ UT, $w_k = \pm 0.01$ UT, $a_k = \pm 1.0$) (ver Figs. 5 y 6). Para un tiempo de ejecución se consideró: $b_k = 0.01$, $h_k = 10$ UT y $z_k = \pm 0.01$ UT. Esto se ilustra en la Fig. 7 y el histograma de frecuencias de c_k en la Fig. 8.

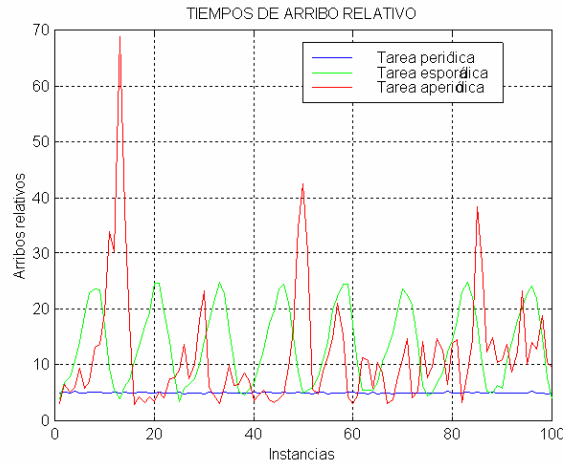


Fig. 5. Tiempos de arribo relativos (inter-arribo) de tres tareas (gráficas sobrepuestas)

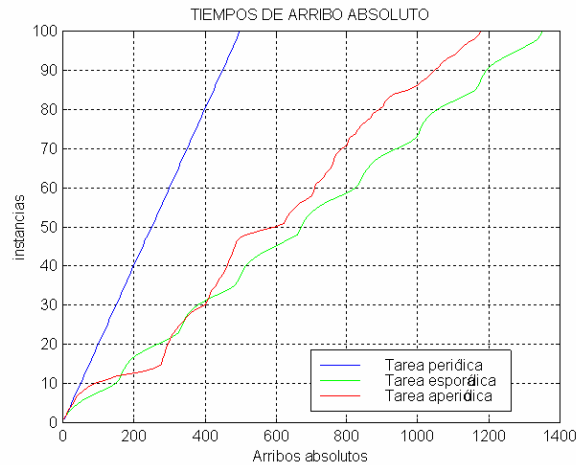


Fig. 6. Tiempos de arribo absolutos de tres tareas (gráficas sobrepuestas)

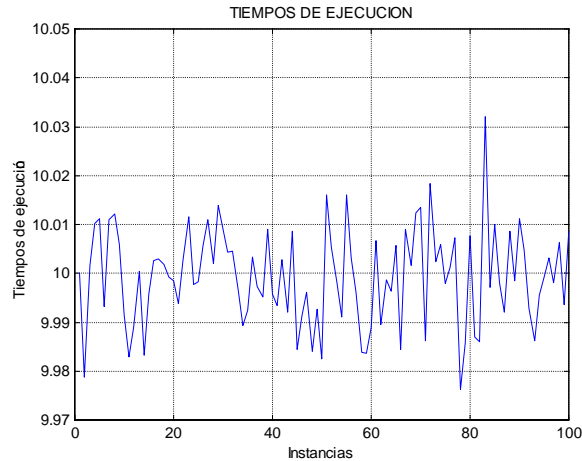


Fig. 7. Tiempos de ejecución para una TTR.

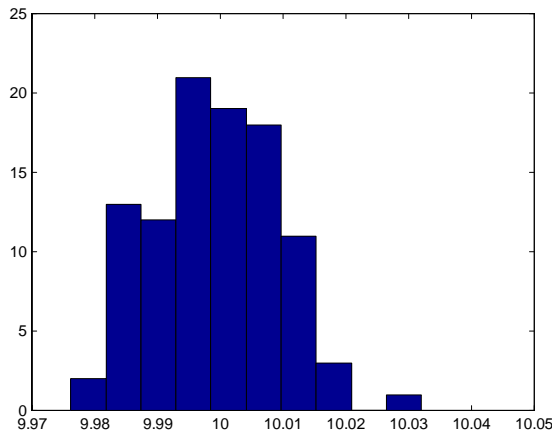


Fig. 8. Histograma de frecuencias del tiempo de ejecución en una TTR.

5 Conclusiones

La mayoría de los modelos de Tareas de Tiempo Real son estáticos y no consideran la dinámicas internas que pueden intervenir su comportamiento; Este trabajo presenta el resultado de una investigación que involucra diferentes factores tanto internos como externos (jitter y errores de medición entre otros) que afectan directamente al comportamiento de una tarea, ya sea periódica, esporádica o aperiódica. El modelo desarrollado es el expuesto por las ecuaciones (2), (17), (20), (30) y (31) se basa en un sistema ARMA [3] y [13], el cual consume pocos recursos para su simulación y presenta un comportamiento en tiempos de arribo tanto relativos como absolutos. De esta manera, si se conoce la dinámica de las tareas es posible hablar de predecibilidad y mejores técnicas de planificación en trabajos futuros.

6 Referencias

[1] **Buttazzo G.** (1997) *Hard real-time computing systems*. Scuola Superiore S. Anna, Kluwer Academic Publishers.
 [2] **Medel J. J., Guevara P.** “Constraints for real-time digital filters” Automatic Control and Computer Sciences AVT No. 5 pages 63-69 ISSN 0132-4160, September 2003, Latvia.
 [3] **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.

- [4] **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.
- [5] **Rajkumar R.** (1991). *Synchronization in real-time systems, a priority inheritance approach*. Kluwer Academic Publishers.
- [6] **Martin J.** (1980). *Diseño de sistemas de computadores en Tiempo Real*. Ed. Diana 1980.
- [7] **Mok A., Chen D.** (1997). *A general model for real-time tasks*. Technical report, University of Texas at Austin.
- [8] **Ramanathan P., Kang D.** (1994). *A generalized guarantee model for servicing sporadic tasks with firm deadlines*. Real-time Systems Journal May 1994.
- [9] **Choi s., Agrawala A** (1997). *Scheduling aperiodic and sporadic tasks in hard real-time systems*. Technical report University of Maryland.
- [10] **Burns A., Wellings A.** “*Real-time systems and programming languages*”. University of York, Addison Wesley, 1997.
- [11] **Abeni L., Buttazzo G.** (1999). *QoS Guarantee using probabilistic deadlines*. Proceedings of the IEEE Euromicro Conference on Real-Time Systems, York, UK, June 1999.
- [12] **Spuri M. & Buttazzo G.** (1996). *Scheduling aperiodic tasks in dynamic priority systems*. Journal of real-time systems.
- [13] **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.
- [14] **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.
- [15] **J. K. Dey, James Kurose, Don Towsley** (1993). *Efficient On-Line Processor for a Class of IRIS (Increasing Reward with Increasing Service) Real-Time Tasks*. SIGMETRICS 1993: 217-228.
- [16] **Ash R.** “*Real Analysis and Probability*”, Ed. Academic Press, USA, 1970.
- [17] **Nyquist H.** “*Certain Topics in Telegraph Transmission Theory*”. AIEE Transactions, USA 1928.
- [18] **T. Abdelzaher, John Stankovic, Chenyang Lu, Ronghua Zhang, Ying Lu.** (2003) “*Feedback Performance Control in Software Services*”
- [19] **K. Jeffay, D. Becker D. Bennett** “*The Design, Implementation and Use of a Sporadic Task Model*”
- [20] **A. Adas,** (1997) “*Traffic Models in Broadband Networks*”. IEEE Communications, 35(7):82--89, July 1997.
- [21] **J. P. Lehoczky.** (1996), “*Real Time Queueing Theory*”. 17th IEEE Real-Time Systems Symposium (RTSS '96) December 04 - 06, 1996 Washington D.C.
- [22] **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.



Pedro Guevara López. *Obtuvo el grado de Maestro en Ciencias de la Computación (1999) en el Centro de Investigación en Computación; el título de Ingeniero Electricista (1997) en la Escuela Superior de Ingeniería Mecánica y Eléctrica, ambos del Instituto Politécnico Nacional. Cuenta con un libro, cerca de 25 publicaciones nacionales e internacionales en revistas y congresos. Sus áreas de investigación son: Filtrado Digital en Tiempo Real y Sistemas en Tiempo Real. Actualmente estudia el Doctorado en Ciencias Computacionales.*



J. J. Medel. *Obtuvo los grados de Doctor en Ciencias (1998) y Maestro en Ciencias (1996) en Ingeniería Eléctrica con especialidad en Control Automático en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional. Actualmente es Profesor-Investigador del Centro de Investigación en Innovación Tecnológica del Instituto Politécnico Nacional y miembro del Sistema Nacional de Investigadores (nivel I), tiene publicado un libro y más de 35 artículos en revistas y congresos nacionales e internacionales. Sus áreas de investigación son: Filtrado Digital, Teoría de Control, Sistemas en Tiempo Real y Procesos Estocásticos.*



Daniel Cruz Pérez. *Es candidato a Maestro en Ciencias de la Computación (2004) del Centro de Investigación en Computación, obtuvo el Título de Ingeniero en Comunicaciones y Electrónica (1999) en la Escuela Superior de Ingeniería Mecánica y Eléctrica, ambos en el Instituto Politécnico Nacional. Ha laborado en el sector privado y ha sido catedrático de diferentes universidades. Sus áreas de interés e investigación son: Redes de Datos, Seguridad Informática y Sistemas en Tiempo Real.*