





**INSTITUTO POLITÉCNICO NACIONAL**

**CENTRO DE INVESTIGACIÓN EN CIENCIA  
APLICADA Y TECNOLOGÍA AVANZADA**

---

**MODELO PARA TIEMPOS DE ARRIBO DE  
TAREAS EN TIEMPO REAL  
CONCURRENTES**

**TESIS QUE PARA OBTENER EL GRADO DE  
DOCTOR EN TECNOLOGÍA AVANZADA**

**PRESENTA:**

**M. EN C. DANIEL CRUZ PÉREZ**

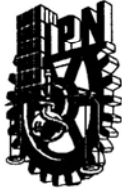
**DIRECTOR DE TESIS: DR. PEDRO GUEVARA LÓPEZ**

**DIRECTOR DE TESIS: DR. JOSÉ DE JESÚS MEDEL JUÁREZ**



**MÉXICO D.F. NOVIEMBRE DE 2007**

# ACTA DE REVISIÓN DE TESIS



SIP-14

## INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

### ACTA DE REVISIÓN DE TESIS

En la Ciudad de México D. F. siendo las 11:00 horas del día 28 del mes de noviembre del 2007 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CICATA-IPN para examinar la tesis de titulada:

#### **Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes**

Presentada por el alumno:

**CRUZ**

Apellido paterno

**PÉREZ**

materno

**DANIEL**

nombre(s)

Con registro: 

A	0	5	0	1	6	8
---	---	---	---	---	---	---

aspirante de:

Doctorado en Tecnología Avanzada

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

Q

#### LA COMISIÓN REVISORA

Director de tesis

Dr. Pedro Guevara López

Director de tesis

Dr. José de Jesús Medel Juárez

Dr. José Antonio Irán Díaz Góngora

Dr. José Antonio Calderón Arenas

Dr. Ernesto Marin Moares

EL PRESIDENTE DEL COMITÉ

Dr. José Antonio Irán Díaz Góngora

CENTRO DE INVESTIGACIÓN EN CIENCIA  
APLICADA Y TECNOLOGÍA AVANZADA  
DEL IPN

## **CARTA DE CESION DE DERECHOS**

En la Ciudad de México Distrito Federal el día 28 de noviembre de 2007, el que suscribe Daniel Cruz Pérez, alumno del Programa de Doctorado en Tecnología Avanzada con registro A050168 adscrito al Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada unidad Legaria, manifiesta que es autor intelectual del presente trabajo de tesis bajo la dirección de los Doctores Pedro Guevara López, y José de Jesús Medel Juárez y cede los derechos del trabajo intitulado "Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes" al Instituto Politécnico Nacional para su difusión con fines académicos y de investigación.

Los usuarios de este documento no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido comunicándose a las oficinas del Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada unidad Legaria del Instituto Politécnico Nacional. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



---

M. en C. Daniel Cruz Pérez

# DEDICATORIAS

A Don **Luis Pérez** (mi abuelito) quien usando su gran retórica y consideración ha compartido conmigo parte de su experiencia y valioso tiempo.

A mi hija **Diana Desiree** por su persistencia en aprender y haber puesto a prueba mi paciencia, cordura y carácter.

A todos aquellos que compartieron conmigo un momento de su tiempo.

# AGRADECIMIENTOS

Agradezco mi hija **Diana Desiree** por invitarme a ser niño nuevamente; a **Irma Olivia Cardoso de Cruz** por haber aceptado compartir su vida conmigo, soportar mi carácter y darme ánimo para continuar superándome; a mi mamá **Jasibe Pérez de Cruz** y mi papá **Alfonso Cruz Córdova<sup>(RIP)</sup>** por apoyarme en todos mis proyectos y mostrarme que las metas son alcanzables: a **Miguel Ángel Cruz Pérez** por ser mi mejor amigo; A mi sobrino Angel Gabriel por ser el mejor amigo de mi hija; a mis abuelitos **Luis Alfonso Pérez Morón** y **Amparo David de Pérez<sup>(RIP)</sup>** por enseñarme lo que es la amistad, sinceridad y ser un pilar importante de mi vida; a mis tíos **Marco Antonio Pérez David** y **Josefina Valenzuela de Pérez** por compartir conmigo su alegría y sabiduría; a mis directores de tesis **Pedro Guevara López** y **José de Jesús Medel Juárez** por todo el apoyo y la amistad que me brindan; al personal del **CICATA Legaria** por su eficiencia, eficacia, amabilidad y buena disposición; Al **Consejo Nacional de Ciencia y Tecnología** por el apoyo económico; Al **Instituto Politécnico Nacional** por ofrecer oportunidades de superación; A los **Estados Unidos Mexicanos** porque me han ofrecido un pluriverso de opciones, paisajes, culturas, ideosincrasias, amistades y conocidos que hacen de la vida un continuo viaje entre el regosijo y la sorpresa.

# ÍNDICE

RESUMEN	III
ABSTRACT	IV
<b>CAPÍTULO 1</b>	<b>SISTEMAS EN TIEMPO REAL</b>
	<b>5</b>
1.1	INTRODUCCIÓN
	5
1.2	SISTEMAS EN TIEMPO REAL (STR)
	5
1.3.1	<i>STR en Sistemas de Cómputo</i>
	9
1.3.2	<i>Implantación de un STR</i>
	12
1.1	CONCLUSIONES
	12
<b>CAPÍTULO 2</b>	<b>EL PROBLEMA DEL MODELADO DE LOS TIEMPOS DE ARRIBO PARA TAREAS EN TIEMPO REAL</b>
	<b>14</b>
2.1	INTRODUCCIÓN
	14
2.1	CLASIFICACIÓN DE LAS TAREAS EN TIEMPO REAL (TTR) DE ACUERDO A SUS TIEMPOS DE INTERARRIBO
	15
2.2	DESCRIPCIÓN DEL PROBLEMA
	16
2.3	OBJETIVOS
	18
2.4.1	<i>Objetivo principal</i>
	18
2.4.2	<i>Objetivos específicos</i>
	18
2.4	HIPÓTESIS
	19
2.5	JUSTIFICACIÓN
	19
2.6	LÍMITES Y ALCANCES
	20
2.7	APORTES DE ESTE TRABAJO
	20
2.8	CONCLUSIONES
	20
<b>CAPÍTULO 3</b>	<b>ESTADO DEL ARTE DEL MODELADO DE TIEMPOS DE ARRIBO EN TAREAS EN TIEMPO REAL</b>
	<b>21</b>
3.1	INTRODUCCIÓN
	21
3.2	EL ESTADO DEL ARTE DEL MODELADO DE TIEMPOS DE ARRIBO EN TAREAS EN TIEMPO REAL
	21
3.3	CONCLUSIONES
	38
<b>CAPÍTULO 4</b>	<b>MODELO PARA TIEMPOS DE ARRIBO DE TAREAS EN TIEMPO REAL CONCURRENTES</b>
	<b>39</b>
4.1	INTRODUCCIÓN
	39
4.2	DEFINICIONES PRELIMINARES PARA LA PRESENTACIÓN DEL MODELO PARA TIEMPOS DE ARRIBO DE TAREAS EN TIEMPO REAL CONCURRENTES
	39
4.3	PROPUESTA DEL MODELO PARA TIEMPOS DE ARRIBO DE TAREAS EN TIEMPO REAL CONCURRENTES
	42
<i>Estabilidad</i>	45
ESTABILIDAD DEL MODELO PARA TIEMPOS DE ARRIBO DE TAREAS EN TIEMPO REAL CONCURRENTES EN SENTIDO DE LYAPUNOV	47

ESTABILIDAD DEL MODELO PARA TIEMPOS DE ARRIBO DE TAREAS EN TIEMPO REAL CONCURRENTES EN UN SENTIDO AMPLIO _____	49
ESTABILIDAD PARA LOS TIEMPOS DE ARRIBO ABSOLUTOS _____	50
CONCLUSIONES _____	52
<b>CAPÍTULO 5 EJEMPLOS Y RESULTADOS EXPERIMENTALES</b> _____	<b>53</b>
INTRODUCCIÓN _____	53
EJEMPLOS _____	53
PODEMOS HACER LAS INTERPRETACIONES SIGUIENTES _____	54
<i>Ejemplo 1</i> _____	54
<i>Ejemplo 2</i> _____	66
<i>Ejemplo 3</i> _____	67
CONCLUSIONES _____	70
CONCLUSIONES GENERALES _____	71
TRABAJOS FUTUROS _____	72
GLOSARIO _____	73
BIBLIOGRAFÍA _____	77
ANEXOS _____	84
PRODUCTOS OBTENIDOS _____	84
<i>Libros</i> _____	84
<i>Revistas con arbitraje internacional</i> _____	84
<i>Congresos nacionales e internacionales con memorias publicadas</i> _____	84



## Resumen

Un Sistema en Tiempo Real (STR) implantado en una computadora digital, interactúa con el mundo físico a través de acondicionamientos de variables (sensores, actuadores convertidores Analógico/Digital  $A/D$  y Digital/Analógico  $D/A$ ) y procesa sus peticiones mediante tareas concurrentes con restricciones temporales. En general cada una de las variables del proceso dinámico (entradas, salidas y estados) es relacionada con una tarea específica; si se trata de un sistema multivariable, entonces se tendrá un conjunto de tareas por cada arreglo de variables donde cada una tendrá que dar respuestas correctas acotadas temporalmente.

En este sentido, la representación matemática de los sistemas en Tiempo Real es de gran importancia; con estos modelos, las pruebas de funcionamiento se pueden llevar a la simulación, haciendo el proceso de diseño, dimensionamiento e implantación más accesible y menos caro.

En el capítulo 1 se describen los antecedentes y conceptos básicos del entorno de las Tareas en Tiempo Real, la problemática, objetivos, hipótesis, alcances y justificación del proyecto. En el capítulo 2 se realiza en estudio del estado del arte sobre el modelado y representación de las Tareas en Tiempo Real. El capítulo 3 contiene la propuesta del modelo de Tareas en Tiempo Real basado en eventos discretos, se definen los conjuntos, las funciones de transición entre estados y se exponen algunos criterios de estabilidad. En el capítulo 4 se presentan diferentes ejemplos de aplicación del modelo propuesto y se dan algunas gráficas y resultados. Finalmente, se presentan conclusiones, referencias bibliográficas y anexos.

### **Palabras clave:**

*Sistemas en Tiempo Real, Tareas en Tiempo Real, Sistema de Eventos Discretos, función de transición, interarribos.*

## **Abstract**

A Real-time system implemented in a computer, interacts with the real physical world through variables conditioners (sensors, actuators, D/A and A/D converters) and processes each processing need as concurrent tasks with temporal deadlines. In general each variable of the dynamic process (inputs, outputs and states) is related with a specific task; if it is about a multivariable system then there is going to be a set of tasks for each variable where each will have to be answered correctly with temporal bounds.

In this mater the mathematical representation of Real-time Systems is of great importance because with these models, functioning analysis could be taken into simulation, making the design, dimensioning and implementation processes less expensive and accessible.

In chapter 1 previous work and basic concepts of Real-time Task are descried. Additionally the problem we must face, objectives, hypothesis, scope and project justification are presented. Chapter 2 shows the state of the art of Real-time tasks modeling and representation. Chapter 3 contains the proposed model for Real-Time Tasks using discrete event systems. Chapter 4 presents various examples for applying the model and shows several graphics and results. Finally, at the en of this work conclusions, bibliographic references and annexes are added.

### **Keywords:**

*Real-time systems, Real-time tasks, discrete event systems, transition function, inter-arrivals.*

# CAPÍTULO 1 **Sistemas en Tiempo Real**

## **1.1 Introducción**

Actualmente el uso de Sistemas en Tiempo Real es tan común, que se puede llegar a imaginar que todas las aplicaciones tecnológicas trabajan en Tiempo Real. Estas aplicaciones deben estar en contacto y en sincronía con el medio que se desee y deben entregar respuestas correctas cumpliendo los plazos temporales que impone el medio.

Debido a las características de sincronía con el medio y las restricciones de tiempo, no existe un Sistema en Tiempo Real universal que resuelva de forma satisfactoria todos los problemas. Cada sistema debe tratarse por separado y se le debe dar solución en específico, de lo contrario se podría caer en la pérdida de sincronía o sencillamente usar un sistema de control muy por debajo de su capacidad, lo que puede elevar injustificadamente el costo de un controlador.

## **1.2 Sistemas en Tiempo Real (STR)**

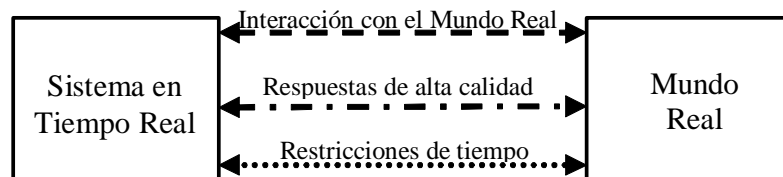
Diversos autores que utilizan a los sistemas digitales y en específico a los sistemas computacionales como una subclase de estos últimos, para observar y/o controlar a un proceso dinámico definen a un STR, de la siguiente manera:

Un STR es aquel sistema digital que obedece tres condiciones: a) Interacción con el mundo físico, b) Emisión de una respuesta correcta de acuerdo a algún criterio

preestablecido, c) Cumplimiento de las restricciones de tiempo del proceso dinámico, conforme a varios criterios.

A pesar de estas características, se tiende a confundir muy a menudo a los sistemas rápidos con los STR o a los lentos como sistemas excluidos de esta clasificación. Pero un STR, es que se ajusta a las restricciones de tiempo impuestas por el sistema o proceso dinámico y que se sincroniza con las dinámicas del sistema dinámico.

A menudo tiende a confundirse el concepto de STR con Sistema en Línea. Un “Sistema en Línea es aquel sistema que siempre debe estar encendido, disponible y generalmente conectados a una red de computadoras y depende de la capacidad del hardware para atender peticiones de servicio” y en ningún momento está en sincronía con el mundo real ni tiene restricciones temporales. En adición a esto, un Sistema Fuera de Línea es aquel que no siempre está disponible para recibir y enviar información y que depende de una base de datos previamente establecida para ejecutar su cometido. Como ejemplos de sistema en línea se tienen las aplicaciones de Internet como los navegadores de páginas Web o la adquisición de datos a través de una tarjeta especializada en un ambiente de tiempo compartido como Windows.



**Figura 1.1 Características de un Sistema en Tiempo Real.**

Una computadora digital con soporte para Tiempo Real interactúa con el mundo físico a través de acondicionamientos (sensores, actuadores convertidores Analógico/Digital (A/D) y Digital/Analógico (D/A) procesa sus peticiones mediante tareas, en este caso tareas de Tiempo Real. En general cada una de las variables del

proceso dinámico (entradas, salidas y estados) es relacionada con una tarea específica. Si se trata de un sistema multivariable, entonces se tendrá un conjunto de tareas por cada arreglo de variables y tendrán que dar respuesta con restricciones de tiempo. Los STR son sistemas compactos con recursos mínimos de hardware (por ejemplo: un solo procesador), y sus tareas se realizan en forma concurrente.

Un STR se puede clasificar conforme al cumplimiento de las restricciones de tiempo generadas por el proceso dinámico, en:

- a. *Críticos o Hard*. En estos, es absolutamente imperativo que las respuestas ocurran estrictamente dentro de los períodos de tiempo especificados, a riesgo de causar un desastre. Como ejemplo se tienen los sistemas de control de vuelo donde se requiere mucha precisión en el manejo de las diferentes superficies de control de vuelo en la aeronave; donde el mínimo retardo en el tiempo de entrega de la información por los sistemas de control a las superficies de control es causa de accidentes e incluso la pérdida de vidas.
- b. *Sistemas no críticos o Soft*. Sus requerimientos temporales son importantes, pero funcionan correctamente si ocasionalmente alguna restricción se pierde: el desempeño es degradado pero no se destruye el sistema por el hecho de fracasar en el tiempo de entrega de la información. Como ejemplo está el control en Tiempo Real sobre un motor eléctrico del tipo derivado, en donde el algoritmo de control implantado para regular su velocidad de arranque tiene pequeños retardos de tiempo respecto a la velocidad nominal de su flecha; y se ha observado que estas variaciones en tiempo de la entrega de la información no afectan el funcionamiento del motor en su desempeño global.
- c. *Sistemas inflexibles o Firm*. Son sistemas no críticos, pero su diferencia consiste que en caso de no respetar las restricciones de tiempo no sirve de nada el servicio que prestan. Como ejemplo se tiene un sistema de adquisición de datos que esté tomando lecturas de sonido con cierto tiempo de muestreo; si algunas lecturas

importantes no son obtenidas, el sistema sigue funcionando, no se corre ningún riesgo pero el sistema prácticamente no sirve debido a que se perdió información.

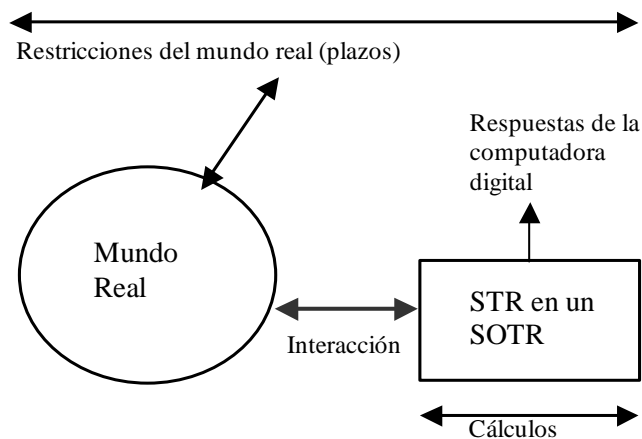
Los STR deben tener ciertas cualidades por encima de otro tipo de sistemas; a continuación se resumen éstas:

- a. *Puntualidad*. Deben acatarse las restricciones de tiempo que dicta el mundo físico, de no ser así, pueden presentarse alteraciones no deseables.
- b. *Soporte para carga pico*. No deben colapsarse bajo condiciones de carga pico. De otra manera, sería peligroso (¡en un avión por ejemplo!).
- c. *Predecibilidad*. El sistema debe ser predecible a consecuencias ante cualquier contingencia del mundo físico.
- d. *Tolerancia a fallos*. Las fallas en el software o hardware no deben ocasionar una caída del sistema, pérdida de su relación con el mundo físico así como pérdida de los procesos de monitoreo y control implantados en el sistema.
- e. *Accesibles para mantenimiento*. La arquitectura de un STR debe tener una estructura modular para posibles modificaciones.
- f. *Mapeo de números reales a enteros y viceversa*. Los procesos reales utilizan señales de entrada y salida continuas y los STR manejan información en tiempo discreto, a la entrada y salida del STR se requiere un mapeo de los reales a los enteros y de los enteros a los reales.
- g. *Soporte de concurrencia*. El sistema físico puede generar múltiples entradas de información (MI- Multi Inputs) para el STR con un solo sistema de monitoreo y / o control, por lo cual el STR debe atender en forma concurrente (multitarea) a ellas. Para obtener ésta característica, muchas veces se hace uso de computadoras.
- h. *Manejo de prioridades*. No todas las actividades que atiende un STR tienen la misma importancia para el mundo real, algunas tienen mayor prioridad para ser atendidas (por que son críticas) mientras que otras pueden esperar hasta que los recursos estén libres. Debido a esto, los STR deben ser capaces de discriminar la importancia de las actividades a través del uso de prioridades.

- i. *Interacción con interfaces de hardware.* La naturaleza de los STR requiere que haya interacción con el ambiente. Por ello son necesarios dispositivos (sensores y actuadores) para tal fin, algunas veces es necesario construir los manejadores o administradores de dispositivos para este fin.

### 1.3.1 STR en Sistemas de Cómputo

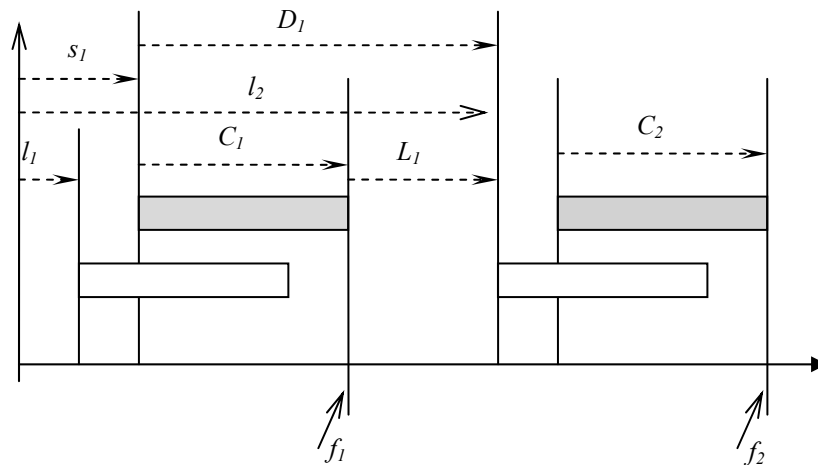
Un STR puede integrarse por una computadora personal (PC) y un Sistema Operativo de Tiempo Real (SOTR) que cumpla con las restricciones que imponga el proceso dinámico. Para éste fin pueden emplearse sistemas operativos como QNX<sup>®</sup>, Lynx<sup>®</sup>, RT-Linux<sup>®</sup>, etc. capaces de dar el soporte adecuado debido a su capacidad de multitareas, sincronización, paso de mensajes entre otras características de los Sistemas de Tiempo Real.



**Figura 1.2 Un STR en una computadora Digital. Cortesía de Medel, Guevara y Cruz, Temas Selectos de Sistemas en Tiempo Real.**

Las principales características de los STR utilizadas en este trabajo son:

- a. *Tarea en Tiempo Real (TTR)*. Una tarea en Tiempo Real es una entidad ejecutable de trabajo  $J_i$  que al menos es caracterizada por un tiempo de arribo y una restricción temporal. Está formada por un conjunto de instancias  $j_{k,i}$ , tal que  $J_i = \{j(k)_i\}, i, k \in \mathbf{Z}^+$ . Una instancia  $j(k)_i$  es una unidad de trabajo de una tarea  $J_i$ , queda definida con  $j(k)_i = f(l(k)_i, c(k)_i, d(k)_i), i, k \in \mathbf{Z}^+$  donde  $l(k)_i$  es el tiempo de arribo de cada instancia,  $c(k)_i$  es el tiempo de ejecución de cada instancia y  $d(k)_i$  es el plazo de cada instancia.
- b. *Tareas concurrentes*. Las tareas concurrentes ( $\{J_i\}, i \in \mathbf{Z}^+$ ) son un conjunto de tareas que se ejecutan en un plazo máximo y en forma no secuencial tomando recursos en instantes diferentes. No pueden existir tareas concurrentes sin un planificador predefinido.



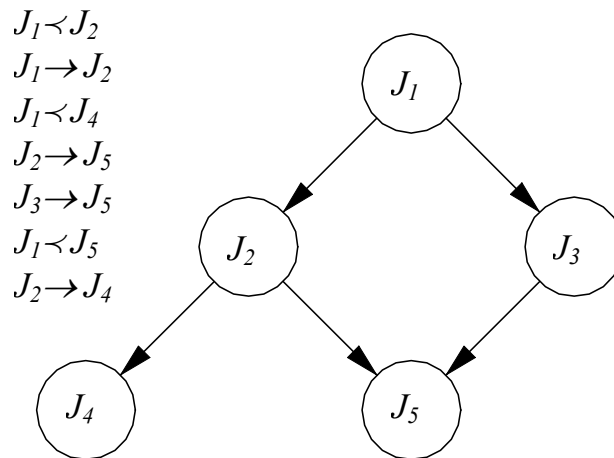
**Fig 1.** Gráfica representativa de las variables presentes en una tarea en Tiempo Real. ( $l_1$ ) es el tiempo de arribo de la instancia 1, tiene un tiempo de inicio  $s_1$ , un tiempo de ejecución  $c_1$  y un tiempo de finalización  $f_1$ . Esta instancia 1 de una tarea es atendida en un tiempo  $s_1$  y sus resultados son entregados en un instante  $f_1$ , con una holgura de tiempo de  $L_1$ . En la segunda instancia sucede algo similar.



c. *Restricción de precedencia.* En ciertas aplicaciones, las actividades computacionales no pueden ser ejecutadas en un orden arbitrario y tiene que respetar ciertas relaciones de precedencia de acuerdo al marco de diseño, es decir, deben tener un orden de ejecución. Estas relaciones de precedencia se describen usualmente por grafos dirigidos ( $G$ ) donde las tareas son representadas por nodos y las relaciones de precedencia por flechas tal que:

- a. La notación  $J_a \prec J_b$  especifica que la tarea  $J_a$  es predecesora de la tarea  $J_b$ , por lo tanto, el grafo  $G$  tiene una trayectoria de los nodos  $J_a$  a  $J_b$ .
- b. La notación  $J_a \rightarrow J_b$  especifica que la tarea  $J_a$  es predecesora inmediata de  $J_b$ , entonces existe una arista dirigida en  $G$  tal que va de  $J_a$  a  $J_b$ .

El manejo de las restricciones de precedencia puede darse por mensajes, señales, tuberías o prioridades en un SOTR del tipo POSIX.



**Figura 1.3** Ejemplo de restricciones de precedencia en tareas concurrentes. Cortesía de Medel, Guevara y Cruz, Temas Selectos de Sistemas en Tiempo Real.

- d. *Plazos*. Los plazos ( $d(k)$ ) de las instancias de las tareas son las restricciones temporales que dicta el proceso con que se interactúa, deben ser cumplidos por la computadora y en función de esto se califica la calidad del STR.

### 1.3.2 Implantación de un STR

La implantación de un Sistema en Tiempo Real depende principalmente del tipo de aplicación a desarrollar. En se proponen algunos casos de estudio donde, en forma general, el proceso de implantación es el siguiente:

- a. Descomposición del sistema en varios procedimientos o tareas. Aquí se dividen los controladores, actuadores, convertidores y se decide si el sistema será lineal, concurrente o paralelo,
- b. Identificación de fases. Una fase representa el comportamiento de un proceso en un intervalo de tiempo,
- c. Formulación de los requerimientos de interacción. Aquí se describe la comunicación entre tareas y su esquema de interacción,
- d. Especificación de la funcionalidad del proceso. En este punto se sincronizan las fases y se empieza a probar el proceso,
- e. Implementación en forma iterativa. Los algoritmos de los procedimientos deben implantarse en forma iterativa para hacer el proceso eficiente.

## 1.1 Conclusiones

Un STR es aquel sistema digital que obedece tres condiciones: a) Interacción con el mundo físico, b) Emisión de una respuesta correcta de acuerdo a algún criterio preestablecido, c) Cumplimiento de las restricciones de tiempo del proceso dinámico.

Un “Sistema en Línea es aquel sistema que siempre debe estar encendido, disponible y generalmente conectados a una red de computadoras y depende de la capacidad del hardware para atender peticiones de servicio” y en ningún momento está en sincronía con el mundo real ni tiene restricciones temporales.

# CAPÍTULO 2 El Problema del Modelado de los Tiempos de Arribo para Tareas en Tiempo Real

## 2.1 Introducción

Las técnicas de planificación estáticas 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 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. Fueron Liu y Layland [1] los que dieron las bases para el modelado de STR partiendo de conceptos diferentes a los utilizados por Martín.

La clasificación común de las TTR se realiza de acuerdo al comportamiento de la ocurrencia de sus arribos, tema que se tratará en este punto, sin embargo es interesante comentar que en función de la forma de ejecución dentro del sistema digital, las TTR pueden clasificarse en expulsivas (preemptive) o no expulsivas (non preemptive). Una tarea expulsiva es aquella cuya ejecución puede ser interrumpida por otras tareas y reanudada de acuerdo a la prioridad que tenga respecto a un nivel de prioridades con que cuenta el sistema digital para permitir el uso de sus recursos. Una tarea no expulsiva es aquella que debe ejecutarse hasta que se complete sus operaciones y realizaciones sin perder su tiempo de cómputo y sin ninguna interrupción. Lo referente a tareas expulsivas y no expulsivas ya no se abordará en

este libro y se dará mayor énfasis a la clasificación de acuerdo a la periodicidad de los tiempos de arribo.

## 2.1 Clasificación de las Tareas en Tiempo Real (TTR) de acuerdo a sus tiempos de interarribo

A continuación se presenta la clasificación de las Tareas en Tiempo Real (TTR) en función de su tiempo de arribo:

- 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.
- Las Tareas aperiódicas son activadas irregularmente con un periodo desconocido. La restricción de tiempo es generalmente el *plazo*  $d_i$ .
- Las tareas Esporádicas 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.
- Las Tareas Semiperiódicas son TTR que tienen una liberación igual a las TTR periódicas pero su tiempo de ejecución  $C_i$ , es variable, pudiendo representarse éste con una función determinística o probabilística acotada por una función de distribución.

## **2.2 Descripción del problema**

La representación matemática siempre ha sido de gran importancia porque da noción del comportamiento de los elementos que la componen, así como de las características de los sistemas que se pueden representar con el. En nuestro caso para representar tareas en Tiempo Real es necesario saber qué tipo de proceso es el que se quiere tratar, ya que los algoritmos matemáticos usados para cálculos varían de una aplicación a otra dependiendo de: la precisión que se busque, tiempo de convergencia y tipo de cálculo. Este comportamiento representa el porcentaje de utilización que se presentará cuando el procesador atienda dicha tarea.

De forma similar, una tarea solicita ser atendida cada determinada frecuencia, en otras palabras una tarea solicitará uso de procesador. Para que dicho sistema conserve la sincronía con el medio con el cual interactúa, los cálculos solicitados para su ejecución en el procesador deben cumplir con cierto plazo temporal, que en la mayoría de los casos está limitado a terminar antes de la siguiente petición de procesamiento por la misma tarea.

Cuando se habla de un sistema complejo o hasta de un sistema con aplicación en el mundo real, se tienen que atender múltiples tareas que solicitan atención de manera simultánea y su representación es necesaria para poder realizar experimentos y simulaciones fuera de línea.

Los modelos encontrados en la revisión bibliográfica representan el comportamiento de las tareas en Tiempo Real para una sola tarea, pero carecen de representación para múltiples tareas con una base matemática clara. Dando una representación matemática a las tareas concurrentes se puede proponer un criterio de estabilidad para el sistema e identificar sus condiciones de estabilidad.

## Capítulo 2. El Problema del Modelado de los Tiempos de Arribo para Tareas en Tiempo Real

Un *Sistema en Tiempo Real* obedece tres condiciones: a) Interacción con el mundo físico, b) Emisión de una respuesta correcta de acuerdo a algún criterio preestablecido y c) Cumplimiento de las restricciones de tiempo del proceso dinámico. En una computadora digital con soporte para Tiempo Real en una plataforma QNX<sup>®</sup>, Lynx<sup>®</sup>, RT-Linux<sup>®</sup>, etc., se interactúa con el mundo físico a través de acondicionamientos y se procesan las peticiones mediante *Tareas en Tiempo Real*; si se trata de un sistema multivariable o concurrente, entonces se tendrá un conjunto de tareas por cada arreglo de variables y deberá darse respuesta con restricciones de tiempo llamadas *plazos* y *restricciones de precedencia*.

Una *Tarea en Tiempo Real (TTR)* es una entidad ejecutable de trabajo que al menos es caracterizada por un tiempo de arribo y una restricción temporal. Está formada por un conjunto de instancias; una instancia es una unidad de trabajo de una tarea.

*Las tareas concurrentes* son un conjunto de tareas que se ejecutan en un plazo máximo y en forma no secuencial tomando recursos en instantes diferentes. No pueden existir tareas concurrentes sin un planificador predefinido.

Las *restricción de precedencia* dan un orden de ejecución a las instancias de las Tareas en Tiempo Real, esto es debido a que en ciertas aplicaciones, las actividades computacionales no pueden ser ejecutadas en un orden arbitrario y tiene que respetar ciertas relaciones de precedencia de acuerdo al marco de diseño.

*Los plazos* de las instancias de las tareas son las restricciones temporales que dicta el proceso con que se interactúa, deben ser cumplidos por la computadora y en función de esto se califica la calidad del STR.

Por lo anterior, una representación del universo involucrado en un sistema en Tiempo Real es de gran ayuda, porque puede acotar límites, da noción para la aplicación de

criterios de estabilidad, se pueden mostrar características de distribución de probabilidad y otras mas.

## **2.3 Objetivos**

### **2.4.1 Objetivo principal**

Desarrollar un modelo discreto para los tiempos de arribo en Tareas en Tiempo Real concurrentes, que generalice la representación de Tareas Periódicas, Esporádicas y Aperiódicas a través de una función de transición conocida.

### **2.4.2 Objetivos específicos**

- Realizar un estudio del estado del arte sobre el modelado y representación de las Tareas en Tiempo Real,
- Proponer un modelo basado en sistemas discretos para Tiempos de Arribo de Tareas en Tiempo Real concurrentes,
- Proponer un conjunto de valores posibles en Tiempos de Arribo,
- Proponer una función de transición entre Tiempos de Arribo que permita la representación de Tareas Periódicas, Esporádicas y Aperiódicas,
- Proponer un análisis de estabilidad del modelo,
- Difundir los resultados obtenidos en libros, artículos de revista y congresos nacionales e internacionales.



## **2.4 Hipótesis**

- Es posible desarrollar un modelo que puede representar el comportamiento de los Tiempos de Arribo de Tareas en Tiempo Real concurrentes.
- Los Tiempos de Arribo están acotados por tiempos máximos y mínimos conocidos, su distribución de probabilidad es conocida y es la misma en todos los instantes de tiempo, en otras palabras son sistemas estacionarios.

## **2.5 Justificación**

Para el diseño de los sistemas en Tiempo Real es necesario conocer la dinámica del medio con el que se desea interactuar y el tipo de procesamiento que se le desea dar a los datos obtenidos del mismo así como en que momento. El hacer un dimensionamiento adecuado de un sistema o conocer en que momento puede producirse una falla por sobre carga de trabajo es de gran ayuda una vez que se conoce el comportamiento de las tareas que intervienen en el proceso en cuanto a su tiempo de arribo, tiempo de ejecución y plazo.

Los modelos que representan a las tareas en Tiempo Real no siempre se pueden llevar a la simulación y existen muy pocos modelos que involucren el comportamiento concurrente de solicitudes de atención ni presentan algún criterio de estabilidad.

La representación del comportamiento de las TTR es importante dado que con ella se pueden hacer pruebas sin tener que probar un sistema físico real, o sea que haría posible la simulación del comportamiento de la concurrencia de solicitudes de atención y analizar su planificación y en varios tipos de sistemas y planificadores. Adicionalmente se propondrá un criterio de estabilidad para los tiempos de arribo, dando así las bases para una representación matemática formal.

## **2.6 Límites y alcances**

- No se analizarán técnicas de planificación,
- La simulación del modelo no se implantará en un Sistema Operativo en Tiempo Real,
- No se presentará un análisis de complejidad computacional o espacial,
- Se propondrá un estudio básico de estabilidad.

## **2.7 Aportes de este trabajo**

- Un modelo matemático para representar los tiempos de interarribo de Tareas en Tiempo Real Concurrentes, basado en Eventos Discretos,
- Una función de transición entre estados (Tiempos de Arribo) recursiva y estocástica, capaz de representar Tareas Periódicas, Esporádicas y Aperiódicas,
- Análisis de estabilidad basado en valores posibles y transiciones válidas entre estados.

## **2.8 Conclusiones**

En el presente capítulo se trataron de los conceptos básicos de Tiempo Real y se mencionó la importancia y las características generales que se deben cumplir para trabajar en Tiempo Real. Se presentaron los objetivos, justificación, hipótesis, límites y alcances del presente trabajo de tesis, definiendo así el ámbito general del presente documento.

# CAPÍTULO 3 Estado del Arte del Modelado de Tiempos de Arribo en Tareas en Tiempo Real

## 3.1 Introducción

Para poder desarrollar un modelo de Tareas en Tiempo Real (TTR) es necesario realizar una revisión de artículos relacionados con el tema. En el presente capítulo se realiza una revisión extensa con respecto a los distintos modelos de TTR que se han utilizado hasta este momento. Es importante mencionar que existen muchos artículos relacionados con TTR y que los aquí seleccionados son representativos ya que en ellos se presentan los modelos usados en otros trabajos.

## 3.2 El Estado del Arte del Modelado de Tiempos de Arribo en Tareas en Tiempo Real

Parte importante de este trabajo es presentar las diferentes formas en que se describe a las Tareas en Tiempo Real. Es necesario señalar que los comentarios que se dan a continuación buscan solamente tomar en cuenta la información relacionada con los modelos de representación de las tareas en Tiempo Real.

En el artículo de Liu y Layland (Scheduling algorithms for multiprogramming in Hard Real-Time Environments) [LL73], se clasifica a las Tareas en Tiempo Real de acuerdo a las restricciones temporales, considerando los tiempo de arribo, los plazos, los tiempos de ejecución y clasifica a las tareas en periódicas o aperiódicas de acuerdo a las condiciones temporales de las mismas respecto a su plazo. En

### Capítulo 3. Estado del Arte del Modelado de Tiempos de Arribo en Tareas en Tiempo Real

consecuencia este trabajo caracteriza a una tarea por dos estados que describen: su tiempo de arribo y su tiempo de ejecución, respectivamente. 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. Menciona también que la utilización del procesador está dada por una relación entre los periodos y los tiempos de ejecución de la siguiente manera:

$$U = \sum_{i=1}^m (C_i / T_i) \quad (3.1)$$

Algunos autores como por ejemplo: A. Cervin y J. Eker, en [CE]; M.K. Gardner y J.W.S. Liu en [GL99]; H. Aydin, R. Melhem, D. Mossé y P. Mejia-Alvarez, en [AMMM99] han usado el modelo de Liu y Layland, para representar las tareas y poder aplicar técnicas de planificación.

A. Mok y D. Chen. (A Multiframe Model for Real-Time Tasks) en [MC97-2] escriben que Las Tareas en Tiempo Real tipo “multiframe” quedan definidas por la pareja  $(\Gamma, P)$  donde  $\Gamma$  es un arreglo de  $N$  tiempos de ejecución  $(C^1, C^2, \dots, C^{N-1})$ , con  $1 \leq N$  y  $P$  es el tiempo mínimo de separación con  $0 \leq P$ . El tiempo de ejecución del  $i$ -ésimo elemento es  $C^{((i-1) \bmod(N))}$ , donde  $1 \leq i$ . Y el plazo para cada instancia es  $P$  después de su tiempo de arribo. En este artículo se tomó en consideración los diferentes tiempos de ejecución de las tareas periódicas donde cada tarea queda especificada por los tiempos de ejecución acotados por el período o tiempo mínimo de separación. En este caso es necesario conocer todas y cada una de las instancias de cada tarea para poder conocer los tiempos de ejecución, pero no menciona en que momento se aplica cada uno de los tiempos mencionados.

A. Mok y D Chen (A General Model for Real-Time Tasks) en [MC97-1] define una tarea en Tiempo Real a través de la pareja  $(\Phi, P)$ , donde  $\Phi$  es un arreglo de tiempos

de ejecución ( $\phi_1, \phi_2, \dots$ ), 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 trabajo es que es necesario conocer los tiempos de ejecución para cada una de las tareas en cada momento, pero se pueden tomar en cuenta cualquier tipo de tareas ya que se especifica explícitamente el tiempo de ejecución de cada una de ellas para cada instante. Tiene el inconveniente de llevar explícitos los tiempos de ejecución, por lo tanto su implementación es “complicada” ya que se necesita tener una relación muy grande donde tener almacenados estos datos. Este trabajo difiere con el de Multiframe en que el tiempo de ejecución de las tareas varía arbitrariamente.

Michael E. Thomadakis (The Mixed Task Scheduling Problem in Open Computing System Algorithm and Techniques) en [Tho99] Menciona sólo de forma breve y sin mayor explicación la forma en que varios tipos de tareas se pueden representar, para poderse planificar. Un sistema de tareas mezcladas es definido de la siguiente forma:  $F := (\tau, J_s, J_a, A_f, A_w)$ , por medio de lo cual se pueden clasificar a las tareas en a) Periódicas “hard”  $\tau := \{ \tau_i(T_i, C_i, D_i, r_i), i=1, \dots, n \}$ , b) esporádicas  $J_s := \{ J_{si}(T'_{si}, C_{si}, D_{si}), i=1, \dots \}$ , c) aperiódicas “soft”  $J_a := \{ J_{ai}(C_{ai}, D_{ai}, r_{ai}), i=1, \dots \}$ , d) interactivas “smooth”  $A_f := \{ J_{fj}(C_{fj}, F_{fj}, Fairness), j=1, \dots \}$ .

Edward A. Lee (Modeling Concurrent Real-Time Processes Using Discrete Events) en [Lee98] comenta que dado un conjunto de valores  $V$  y etiquetas  $T = \mathcal{R}$ , se puede definir un evento  $e$  como un miembro del producto cruz; i.e.:  $E = T \times V$ , donde de acuerdo a este trabajo, un evento es un vector que tiene una etiqueta y un valor: las etiquetas las usa para modelar el tiempo y los valores ( $V$ ) representan los operandos y los resultados del cálculo. Entonces el evento  $e$  queda descrito por la pareja  $(t, v)$ , es decir:  $e=(t, v)$ . En este trabajo la forma en que se ejecutan las tareas es usando una función de disparo, en donde la instrucción a ejecutarse es la de menor en tiempo (*time stamp*) el cual se relaciona con las etiquetas de los eventos y el tiempo en que ocurren. Finalmente relaciona las TTR y los eventos discretos, sin embargo tiene el

inconveniente que a la hora de hacer la programación se toma en cuenta el tiempo mínimo entre procesos. Relaciona el sistema en TR con el análisis de entrada / salida, no presenta las restricciones de tiempo de forma clara, ni define el tipo de tareas que está tratando y no están muy claras las restricciones de tiempo en las TTR.

P. Ramanathan y D. Kang (A Generalized Guarantee Model for Servicing Sporadic Tasks with Firm Deadlines) en [RD94] definen un modelo de tareas esporádicas, en donde los tiempos de ejecución tienen una variación estocástica y la función de distribución probabilística es de la variación conocida para el sistema. Además cada tarea esporádica es asociada con dos costos,  $K_e$  y  $K_l$ . Donde el costo del sistema en caso que la tarea sea rechazada tan pronto como llegue es expresado por  $K_e$ . El costo del sistema en caso que la tarea no cumpla con su plazo crítico después de ser aceptada por el sistema es expresado por  $K_l$ . 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 plazo. El plazo característico de la tarea  $D$  donde  $\tau \in P \cup S$ ,  $P$  es el conjunto de tareas periódicas del sistema y  $S$  es el conjunto de tareas esporádicas del sistema  $D_s$ . El plazo se expresa como:

$$\Delta_{\tau}(t, D_s) = \begin{cases} \left( \left\lfloor \frac{t}{P_{\tau}} \right\rfloor + 1 \right) P_{\tau} & \text{Si } \tau \in P \\ D_{\tau} & \text{Si } \tau \in S \end{cases} \quad (3.2)$$

El subíndice “ $\tau$ ” denota la instancia 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 de estas clases de tareas. Pero en este trabajo no se define claramente el modelo de la tarea ya que sólo se caracteriza por su plazo y su capacidad para integrarse al planificador, pero pueden representar tareas esporádicas y periódicas. La forma de tratar los dos tipos de tareas es distinta, y su representación es la misma.

S. Choi y A. Agrawala (Scheduling Aperiodic and Sporadic Tasks in Hard Real-Time Systems) en [CA97], describen las tareas aperiódicas  $\tau_i$  (del conjunto de tareas aperiódicas  $T = \{ \tau_1, \tau_2, \dots, \tau_N \}$ ) considerando el 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  $\tau_i$  en cualquier instante de tiempo, Variable de ejecución  $w_i$  denotando el último de comienzo o de  $\tau_i$ , 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 del tipo críticas asumieron que el “tiempo de listo” (*ready time*) es igual al tiempo de arribo. Asumieron además que la planificación inicial de las instancias se da en una ventana de planificación  $[0, L]$ , denotada por  $\Gamma$ , donde  $T = \{ \tau_1, \tau_2, \dots, \tau_N \}$  es un conjunto de instancias de tarea en  $\Gamma$ , y  $\tau_i < \tau_{i+1}$ . Cada tarea consta de los siguientes parámetros en el planificador: a) Tiempo de Liberación Virtual (*virtual release time*)  $R_i$ , b) Plazo Virtual (*Virtual Deadline*)  $D_i (\leq L)$ , c)  $\Gamma$  es el espacio de tiempo del planificador  $[0, L]$ . Definen 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  $S_i$  y se asume que se conocen el tiempo mínimo de inter-arribo  $\delta_i$ , el tiempo máximo de ejecución  $c_i^S$  y el plazo relativo  $d_i^S (\leq \delta_i)$ . También se asume que las  $S_i$  están ordenadas ascendentemente por su plazo relativo,  $d_i^S$ , ejemplo  $d_i^S \leq d_{i+1}^S$ .

K. Jeffay, D. Becker D. Bennett (The Design, Implementation and Use of a Sporadic Task Model) en [JBB94] definen una tarea  $T$  como una secuencia lógica infinita de instancias  $J_j$ . Proponen una tarea aleatoria  $T$  como un conjunto  $(J, c, p)$ , donde  $J$  es una secuencia de instancias (*jobs*)  $\langle J_1, J_2, \dots \rangle$  de la tarea  $T$ . Una instancia  $J_i$  se define con una pareja  $(r_i, d_i)$ , donde  $r_i$  es el tiempo de arribo y  $d_i$  es su plazo, “ $c$ ” es el tiempo de cálculo y “ $p$ ” es la separación mínima entre instancias. En este trabajo se asumió que los tiempos de inter-arribo son los de razón máxima, o sea el peor caso

donde la relación para el pase de mensajes que relaciona con los tiempos de arribo, está descrita:

$$f(r) = \left(\frac{1}{x}\right)r \quad (3.3)$$

donde  $r$  es la razón de llegada de los mensajes y  $x$  es positivo, diferente de cero y para cada función de razón de transmisión, su valor dependerá de la lógica del proceso. Para poder hacer el modelado de estas tareas es necesario conocer todos los tiempos de arribo mínimos “ $p$ ” para realizar la relación de variables de las tareas. La metodología desarrollada no es general ya que sólo hace referencia a un tipo de tareas.

K. Jeffay, D.F. Stanat y C. U, Martel (On Non-Preemptive Scheduling of Periodic and Sporadic Tasks) en [JSM91] describen a una tarea periódica comentando que estas invocan periodos regulares mientras que una tarea esporádica invoca periodos arbitrarios de tiempo pero este periodo tiene un valor mínimo conocido. De acuerdo a su trabajo, consideran que una tarea “ $T$ ” esta formada por una pareja  $(c,p)$  donde: a) “ $c$ ” es el tiempo de cálculo máximo de terminación de la tarea, b) “ $p$ ” 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 le momento de la  $k$ -ésima invocación de la tarea “ $T$ ” entonces: 1) La  $(k+1)$ -ésima invocación ocurrirá en  $t_{k+1}=t_k+p$ , 2) 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: 1) La  $(k+1)$ -ésima invocación ocurrirá no antes de  $t_k+p$ , por lo tanto  $t_{k+1} \geq t_k+p$ , b) La  $k$ -ésima ejecución de la tarea  $T$  debe comenzar después de  $t_k$  y debe ser completada antes de  $t_k + p$ . *Asumieron que* las invocaciones de las tareas esporádicas son independientes y dependen solamente del momento de la última invocación. Este trabajo presentó la relación de tiempos de



arribo y relaciona las tareas periódicas y esporádicas aplicando ciertas restricciones pero no toma en cuenta las tareas aperiódicas.

Kevin Jeffay (Scheduling Sporadic Tasks with Shared Resources in Hard-Real-Time Systems) en [Jeff92] define una tarea por un arreglo de 3 elementos:

$$T_i = (s_i, \{(c_{ij}, C_{ij}, r_{ij}) | 1 \leq j \leq n_i\}, p_i), \quad (3.4)$$

donde:  $s_i$  es el tiempo de arribo de la tarea  $T_i$ , el tiempo de la primer invocación de la tarea  $T_i$ ,  $\{(c_{ij}, C_{ij}, r_{ij})\}$  es un conjunto de fases donde para cada fase:  $c_{ij}$  es el costo mínimo de cómputo: el lapso mínimo de tiempo de procesador requerido para ejecutar la fase  $j$ -ésima de la tarea  $T_i$  hasta completarlo en un procesador dedicado,  $C_{ij}$  es el costo máximo de cómputo: el lapso máximo de tiempo de procesador requerido para ejecutar la fase  $j$ -ésima de la tarea  $T_i$  hasta completarlo en un procesador dedicado,  $r_{ij}$  es el requerimiento de recurso: el recurso (si se requiere) que es requerido durante la fase  $j$ -ésima de la tarea  $T_i$  y  $p_i$  es el periodo de la tarea: el tiempo de inter-arribo mínimo de la tarea  $T_i$ . En el documento describe tareas esporádicas, si embargo su comportamiento en el tiempo no es descrito, su principal interés es la carga de trabajo para el procesador y la planificación.

S. K. Baruah, A. K. Mok y L. E. Rosier (Preemptively Scheduling Hard-Real-Time Sporadic Task on One Processor) en [BMR90] comentan: Una tarea esporádica  $T_i$  está caracterizada por tres elementos: tiempo de ejecución  $e_i$ , un plazo  $d_i$ , y un periodo mínimo de separación  $p_i$ , con  $e_i \leq d_i$  y  $e_i \leq p_i$ , de tal forma que  $T_i = (e_i, d_i, p_i)$ ,  $i \leq l \leq n$ . En este caso, las tareas se representan con tres valores.

Patricia Balbastre (Modelo de Tareas para la Integración del Control y la Planificación en sistemas de Tiempo Real) en [Bal02] propone un modelo para  $n$  tareas periódicas:

$$T^C = (T_1^C, T_2^C, \dots, T_n^C)^*, \quad (3.5)$$

de las cuales cada  $T_i$  es descrito:

$$T_i^C = (C_i, D_i, P_i, O_i, OCW_{(i)}), \quad (3.6)$$

Que en forma simplificada tiene la forma:

$$T_i^C = (C_j, D_j, P_j, O_j). \quad (3.7)$$

Denominando a (2.7) como el Modelo de tareas con el cómputo restringido a ventanas temporales, donde OCW es la ventana de cómputo opcional y se obtiene:

$$OCW_{(i)} = \left\{ C_i + \Delta C_i, \left\langle \frac{n_i}{m_i} \right\rangle \right\}. \quad (3.8)$$

Y la ventana de cómputo opcional consecutiva de una tarea es

$$OCW_{(i)} = \left\{ C_i + \Delta C_i, \left\langle \frac{n_i}{m_i} \right\rangle \right\}, \quad (3.9)$$

Donde  $\left\langle \frac{n_i}{m_i} \right\rangle$  indica el número  $n$  de activaciones de un total de  $m$  que una tarea

puede ejecutarse con un tiempo de cómputo superior al inicialmente signado,  $\left\langle \frac{n_i}{m_i} \right\rangle$

indica que una tarea cumple la restricción temporal  $n$  sobre  $m$  consecutivo. Si, en cualquier ventana de  $m$  invocaciones consecutivas de una tarea, puede ejecutarse con un tiempo de cómputo superior al normal, al menos en  $n$  activaciones consecutivas.

$\Delta C$  es el tiempo de cómputo a incrementos.  $O_i$  es el offset (fase) del tiempo de arribo.

Si la tarea es planificable, entonces  $T_i^C$  se descompone en tareas simples sin el parámetro OCW.

$$\begin{aligned} T_i &= \{C_i, D_i, P_i, 0\} \\ T_{ij} &= \{\Delta C_i, D_i, m_i P_i, j P_i\}, \forall 0 \leq j \leq n_i \end{aligned} \quad (3.10)$$

Si la tarea no cumple la restricción OCW, la cual implica que la tarea no es planificable, se considera que sólo puede ejecutarse en su versión básica (sin cómputo opcional):

$$T_i = \{C_i, D_i, P_i, O\}. \quad (3.11)$$

El modelo que presenta esta autora para describir las TTR es

$$T_i = \{C_i, D_i, P_i, O\}, \quad (3.11)$$

ya que los demás modelos involucran prioridades, recursos compartidos, etc., los cuales afectan el tiempo de ejecución y no describen las características de los tiempo de arribo.

Guevara López, J.J. Medel Juárez y A. Flores Rueda ( Modelo ARMA para Caracterizar Tareas en Tiempo Real) en [GMF02], proponen un modelo regresivo con promedios móviles para caracterizar los tiempos de arribo dinámico en las Tareas en Tiempo Real. El modelo es el siguiente:

$$x_{k+1} = a_k x_k + \omega^1_k, \quad (3.12)$$

$$y_k = x_k + \omega^2_k, \quad (3.13)$$

donde los ruidos interno y externo  $\omega^1_k$  y  $\omega^2_k$  que dependen del sistema computacional y del medio ambiente respectivamente, no están correlacionados entre sí, pero si con la señal generadora de tareas  $y_k$  ( $y_k$  es el tiempo de arribo). Las dinámicas internas descritas por  $x_k$  permiten a través del parámetro  $a_k$  y el ruido interno  $\omega^1_k$  modelar las Tareas en Tiempo Real. Se considera que el tiempo de ejecución y el plazo son constantes.

K. Ramamritham y J. A. Stankovic (Scheduling Algorithms and Operating Systems Support for Real-Time Systems) en [RS94], desarrollaron un estudio sobre los algoritmos de programación para sistemas en Tiempo Real. En varias partes de ese documento se menciona a las Tareas en Tiempo Real, pero en ningún lugar se hace referencia algún modelo.

Tarek F. Abdelzaher (A Schedulable Utilization Bound for Aperiodic Tasks) en [Abd00] realiza una representación de una tarea y aclara que los tiempos de arribo y plazos son asignados en forma arbitraria. El tiempo de arribo de una tarea  $T_i$  se denota con  $A_i$ , su tiempo de ejecución (posiblemente desconocido para el SO) 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 promedio de utilización del procesador  $U_i$  contribuido por esta tarea es  $U_i=C_i/D_i$  en el intervalo entre su arribo y el plazo. La representación es:

$$T_i=(A_i, C_i, D_i), \quad (3.14)$$

donde  $A_i$  es el tiempo de arribo,  $C_i$  el tiempo de ejecución,  $D_i$  el plazo.

M. Spuri y G Buttazzo (Scheduling Aperiodic Tasks in Dynamic Priority Systems) en [SB97] comentan que para insertar las tareas en un planificador dinámico, se debe considerar: a) Todas las tareas  $\tau_i: i=1, \dots, n$  tienen plazos críticos; b) Todas las tareas aperiódicas  $J_i: i=1, \dots, m$  no tienen plazos; c) Cada tarea periódica  $\tau_i$  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; d) Todas las tareas periódicas son activadas simultáneamente al tiempo  $t=0$ ; por ejemplo, la primer instancia de cada tarea periódica tiene un tiempo de arribo  $r_i(0)=0$ ; e) El tiempo de arribo de la  $k$ -ésima instancia periódica está dado por :

$$r_i(k)=r_i(k-1)+T_i; \quad (3.15)$$

f) el plazo de la  $k$ -ésima instancia periódica está dado por :

$$d_i(k)=r_i(k)+T_i; \quad (3.16)$$

g) el tiempo de arribo de cada tarea aperiódica es desconocido; h) el peor tiempo de ejecución de cada tarea aperiódica se considera conocido en su tiempo de arribo. Tratando en este trabajo en forma separada a las tareas periódicas y las aperiódicas. Sin embargo toma en cuenta los tiempos de arribo de forma absoluta de la siguiente manera: 1) El tiempo de arribo de la  $k$ -ésima instancia periódica está dado por:

### Capítulo 3. Estado del Arte del Modelado de Tiempos de Arribo en Tareas en Tiempo Real

$$r_i(k) = r_i(k-1) + T_i; \quad (3.15)$$

2) el plazo de la  $k$ -ésima instancia periódica está dado por :

$$d_i(k) = r_i(k) + T_i; \quad (3.16)$$

por lo que considera que el plazo es igual al periodo. Para tareas periódicas considera:

a) Todas las tareas  $\tau_i: i=1, \dots, n$  tienen plazos tipo “hard”, 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, b) Todas las tareas periódicas son activadas simultáneamente al tiempo  $t=0$ ; por ejemplo, la primer instancia de cada tarea periódica tiene un tiempo de arribo  $r_i(0)=0$ , c) El tiempo de arribo de la  $k$ -ésima instancia periódica está dado por (2.15), d) El plazo de la  $k$ -ésima instancia periódica está dado por (2.16). Para tareas aperiódica considera: a) Todas las tareas aperiódicas  $J_i: i=1, \dots, m$  no tienen plazos; b) El tiempo de arribo de cada tarea aperiódica es desconocido; c) El peor tiempo de ejecución de cada tarea aperiódica se considera conocido en su tiempo de arribo.

J. K. Dey, James Kurose, Don Towsley (On-Line Processor for a Class of IRIS (Increasing Reward with Increasing Service) Real Time Tasks) en [DKT93] introducen un modelo de planificación para Tareas es 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 mas tiempo se le conceda, más preciso es el cálculo. El valor de la “recompensa” está asociado a una función no decreciente y cóncava.

John P. Lehoczky (Real Time Queuing Theory. Carnegie Mellon University, Pittsburgh) en [Leh96] propone la teoría de colas y su aplicación a la forma de atención de las tareas, se asume que el sistema tiene una cola de atención y basado en esto se le asignan prioridades a las tareas.

Pedro Mejía-Alvarez, R. Melheim, D. Mossé y H. Aydin (An Incremental Server for Scheduling Overloaded Real-Time Systems) en [MMMA03], considera a las tareas

periódicas expulsivas ejecutándose en un procesador. Las tareas son independientes y no tienen restricciones de precedencia. Se considera que la tarea  $\tau_i$  llega en el tiempo  $a_i$ . El “tiempo de vida” de cada tarea  $\tau_i$  consiste de un número fijo de instancias  $r_i$ , después de cada  $r_i$  instancias la tarea deja el sistema. El tiempo de inter-arribo entre dos instancias consecutivas  $\tau_x$  y  $\tau_y$  se define como:

$$l_{xy} = a_y - a_x. \quad (3.17)$$

Considera que el plazo  $d_i$  de la tarea  $\tau_i$  es igual al periodo  $T_i$  de cada instancia y que el tiempo de cálculo  $C_i$  es el peor tiempo de cálculo. El tiempo de ejecución  $C_i$  consiste en una parte obligatoria de duración  $m_i$  y una parte opcional de duración  $p_i$ .

S. Lauzac, R. Malheim y D. Mossé (An Improved Rate-Monotonic Admission Control And Its Applications) en [LMM] usan el modelo de Liu y Layland de tareas periódicas donde la tarea está representada con su periodo  $T$  y su tiempo de cálculo y asume que el plazo es igual al próximo tiempo de inter-arribo. Las tareas periódicas  $\tau_i$  se caracterizan por un periodo  $T_i$  y un peor tiempo de cálculo  $C_i$ .

Entre los autores que mencionan varios tipos de representaciones de comportamiento está Adas, que en [Ada97] presenta varios modelos de representación de tráfico en redes de banda ancha, entre los modelos que menciona se encuentra el modelo ARMA que es el utilizado para el desarrollo del modelo propuesto en esta tesis. El modelado de tráfico es importante debido a los distintos tipos de aplicaciones que están surgiendo con necesidad de Calidad de Servicio, Tiempo Real y Prioridad de Servicio entre ellos el Video con VBR (*Variable Bit Rate*). Otros autores que usan el modelo ARMA son M. S. Squillante, D D Yaoy y L. Zhang en [SYZ] dónde lo usan para modelar tráfico de *Web*.

Pizolo, Hosking y Vitek en [PHV07] Hierarchical Real-Time Garbage Collection, tartan la reducción en tiempos debida a los colectores de basura de software y hacen una propuesta de múltiples colectores de basura. En este documento se resaltan las grandes pérdidas de tiempo disponible para cálculo que se tienen con un solo colector de basura, dejando muy poco de éste para el uso de las tareas conrestricciones temporales. Presentan algunas variaciones del “RTSJ” (Especificaciones para Java de Tiempo Real, por sus siglas en inglés). Obtienen como resultado la variación considerable de los tiempos de atención disponibles y los presentan en gráficas.

Auerbach, Baco, Iercan, Kirsch, Rajan, Röck y Trummer en [ABIKRRT07] *Java takes flight: Tiem-portable real-time programing with exatasks*, menciona las variaciones en tiempos de arribo de tareas con restricciones temporales y mencionan los peóres tiempos de ubicación (WCA, por sus siglas en inglés) en memorias de los equipos de cómputo. También el colector de basura es mencionado como partícipe de las variaciones.

Sun, Belding-Royer, Gao y Kempf en [SBGK07] *Real-time traffic support in heterogeneous mobile networks* mencionan los tiempos de latencia en redes móviles heterogeneas. Muestran las variaciones en los tiempos de latencia en varias situaciones y presentan las diferencias obtenidas en tráfico con calidad de servicio (QoS) y otro tipo de tráfico.

Giese y Henkler en [GH06] *Architecture-driven Platform independent deterministic replay for distributed hard real-time systems* mencionan que el problema de concurrencia por gran número de fuentes no determinísticas, la hacen impredecible.

Huang, Guérin y Gupta en [HGG06] *Supporting exceso real.time traffic with active drop queue* solucionan los tiempos de latencia para tráfico en Tiempo Real como voz sobre IP (VoIP) y video con un mecanismo de administración de funcionamiento del planificador y del buffer. Separan el tráfico en red con tres buffer diferentes Uno para el tráfico en Tiempo Real, otro para el tráfico de exceso de las tareas en Tiempo Real

y un tercero para cualquier otro tipo de tráfico. Se menciona que el objetivo es garantizar la frecuencia así como el retraso de las solicitudes.

Tan y Money en [TM07] *Timing análisis for preemptive multitasking real-time systems with caches* tartan los peores casos para tiempos de respuesta (por sus siglas en inglés WCRT). Mencionan múltiples tareas en Tiempo Real  $T_1, T_2, \dots, T_{n-1}$  donde  $n$  es el número de tareas. Cada tareas  $T_i$  tiene un prioridad  $p_i$  y periodo  $P_i$ . Cada tareas  $T_i$  arriba al principio de su periodo y debe concluir antes del final de su periodo. Comentan también que el WCET se ve afectado por las distintas tareas que han sido desalojadas y que esperan turno. Algo similar lo trata Tan en [T05] *Cache design and timing análisis for preemptive multitasking real-time uniprocessor systems*, Siendo éste el trabajo de partida del primero. .

In-Koo Kang, Dong-Hyuck Im, Heung-Kyu Lee y Young-Ho Suh en [KILS06] *Implementation of Real-Time Watermarking Scheme for High-Quality Video* Proceeding of the 8th workshop on Multimedia and security, mencionan que la velocidad para realizar las marcas en los videos deben de tener un máximo de tiempo ya que de lo contrario se presentarían retrasos en la transmisión de las tramas. Por ejemplo mencionan que para el caso de 30 fps (imágenes por segundo por sus siglas en inglés), una imagen debe ser decodificada y desplegada en aproximadamente 0.03 segundos y es obvio que para dejar una marca en la imagen debe ser mucho menor. En el artículo presentan las ecuaciones con las que se calculan los retrasos en la realización de cada proceso.

Calandrino, Baumberger, Li, Hahn y Anderson en [CBLHA06] *Soft Real-time scheduling on performance asymmetric multicore platforms* tartan tareas en Tiempo Real no críticas es sistemas con diversas unidades de procesamiento. Menciona que hay sistemas que deben soportar cargas de Tiempo Real así como cargas sin tantas restricciones de tiempo. Tales aplicaciones se pueden apreciar en dispositivos multipropósito que por ejemplo pueden dar los servicios de: aplicaciones de



multimedia, como accediendo video por demanda, accediendo a video almacenado y grabando imágenes; monitoreo de otros sistemas del hogar como la temperatura de los cuartos y reaccionando adecuadamente; proveer de capacidad de procesamiento para terminales de usuario y dando la funcionalidad de una P.C. Consideran el tráfico de multimedia como de Tiempo Real de tipo no crítico (soft). Se mencionan los retardos que se pueden presentar para los tiempos de inicio. El modelo que se menciona en este artículo para las tareas, aparentemente sólo toma la frecuencia de arribo, menciona que el tiempo de ejecución se mejora comparado con el tiempo en espera de ser atendido y las restricciones temporales de las instancias no son tratadas.

Van Tan, Dae-Seung, Myung-Kuyn y Myeong-Jae en [VDMM06] *Hard and soft real time based on switched Ethernet* comentan que el protocolo 802.11 no fue hecho para soportar trafico en Tiempo Real, sin embargo realizan un estudio y hacen adecuaciones para calcular el uso de red para las tareas en Tiempo Real. El modelo de las tareas en Tiempo Real que usan  $\{T_{pd,i}, C_i, T_{d,i}\}$  se forma con un periodo  $T_{pd,i}$ , un tiempo de ejecución  $C_i$  y un plazo relativo  $T_{d,i}$  que es usado por el algoritmo de planificación EDF, por tanto tenemos. El periodo de la tarea  $T_{pd,i}$  lo calculan con  $T_{pd,i} = T_{u,i} + T_{d,i} + T_{sw}$ , donde  $T_{u,i}$  y  $T_{d,i}$  son el plazo de cada tarea en Tiempo Real para transitar en la red u  $T_{sw}$  es el retardo del conmutador (*switch*).

Ngolah, Wang y Tan en [NGT06] *Implementing the real-time processes of RTPA using Real-time Java* tartan la “Real Time Processor algebra” y lo comentan aplicándolo en Java en Tiempo Real. Tratan la estructura de representación para el algebra arriba mencionada y comenta que se usan los relojes de Tiempo Real de java. No comentan ningún modelo de tareas en Tiempo Real.

Morales, Vasques, Portugal y Fonseca en [MVPF06], *VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks*, tratan el tráfico en Tiempo Real usando un medio de transporte físico común entre

conexiones de tiempo-real y otras que no solicitan restricciones temporales. En la descripción de la simulación consideran cuatro tipos de tráfico: tiempo-real (comportamiento periódico), voz, video y *background*. La mejora de tiempo de respuesta la hacen con una estafeta (*token*) que mejora el tiempo de acceso al medio para los procesos con restricciones de tiempo. Los procesos con restricciones de Tiempo Real tienen tiempos de arribo caracterizados con un valor medio  $\mu$  y una desviación estándar  $\sigma$  donde  $\sigma/\mu \leq 1\%$ . Se considera que la longitud de los mensajes de los procesos en Tiempo Real son constantes.

Lin, Kaldewey, Povzner y Brandt en [LKP06] Diverse Soft Real-time Processing in an Integrated System tratan la forma en que son atendidas las tareas en tiempo-real de tipo *Firm*. Se menciona que dichas *Firm* pueden tener cierta tolerancia de pérdidas de información dadas por  $(mr, mn).firm$  permiten que un porcentaje  $mr$  de instancias pierdan sus plazos, siempre y cuando el número de plazos perdidos consecutivos no exceda  $mn$ .

Hyon-Young, Sung-Gi en [HS06] RED-RT for Congestion Control and Real Time Traffic Protection Proponen un RED-RT (*Random Early Detection Real-time*) que mejora el funcionamiento para Tiempo Real ya que el AQM (Active Queue Management) no lo consideraba. RED-RT considera el manejo del buffer de los sistemas de ruteo (routers) y asegura la calidad de las conexiones en Tiempo Real en situaciones sin congestión.

Tianzhou, Wei, Bin y Like en [TWBL06] *A Real-time Scheduling Algorithm for Embedded Systems With Various Resources Requirements* tratan un algoritmo de planificación, sin embargo mencionan poco o casi nada de las características de las tareas en Tiempo Real. Las características que mencionan de las tareas en Tiempo Real son las consideradas para su planificación: tiempo de inicio, prioridad, plazo y “combinaciones de otros atributos”. Los detalles de las combinaciones son

consideraciones realizadas como distancias entre procesos, en donde se ven diferencias o afinidades y de las cuales se decide su asignación.

Tao Xie y Xiao Qin en *Scheduling Security-Critical Real-Time Applications on Clusters* [XQ06] comentan que existen aplicaciones en donde la seguridad de los datos es igualmente importante como el cumplimiento de las restricciones temporales. Ellos usan un modelo de tareas en donde se adicionan datos como el nivel de seguridad y la cantidad de datos que se desés resguardar. Comentan que aún cuando es sistema de seguridad es importante, el tiempo de retardo o el aumento en el tiempo de ejecución no debe pasar los plazos establecidos. El modelo de la tarea en Tiempo Real que usan es el siguiente:  $T_i = (a_i, e_i, f_i, d_i, l_i, S_i)$ , donde  $a_i, e_i$  y  $f_i$  representan los tiempos de arribo, ejecución y los tiempos de finalizado respectivamente,  $d_i$  es el plazo,  $l_i$  denota la cantidad de datos a proteger y  $S_i$  el nivel de seguridad de los datos. Aún cuando menciona los tiempos de ejecución para el cifrado y seguridad de los datos y los encabezamientos necesarios para cada caso, no se mencionan ninguna característica adicional del modelo para los tiempos de arribo de la tareas.

Wei, Son y Satnkovic en [WSS06] *Real-time query processing for data stream*, Tratan el acceso a las bases de datos en tiempo-real. Menciona que dichas bases de datos son empleadas ampliamente en la la industria de comunicaciones en donde las restricciones temporales son muy importantes. El tiempo de comportamiento al que enfocan el trabajo es al de solicitudes de tiempo periódico. Las extensiones del lenguaje de búsquedas (query) en este trabajo son las siguientes: STIME, tiempo de inicio de una búsqueda periódica; ETIME, tiempo de finalizado de una búsqueda periódica; PERIODO, periodo de la búsqueda en caso de que aplique; PLAZO, plazo de la búsqueda (relativo al tiempo de inicio de la búsqueda) e IMPORTANCIA, nivel de importancia de la búsqueda. En esta representación, una búsqueda que no requiera tiempo-real marcará el PLAZO con un valor grande.

Cavanagh, Wang en [CW05] *Design of a real-time virtual machine (RTVM)* proponen una máquina virtual de tiempo-real para que las aplicaciones puedan ser portables siempre y cuando sean tareas no críticas. Sus servicios básicos son el manejo de interrupciones, manejo de eventos, administrador de memoria y el planificador.

Jia-Li en [Jia05] *XiaCPAHR: an concurrency control protocol against hard real-time transactions* trata la búsquedas críticas en tiempo-real, sin embargo se enfoca al problema de restricción de recursos y no al de restricciones temporales. Sin embargo no menciona la forma en que se presentan las solicitudes de búsqueda.

### **3.3 Conclusiones**

En el presente capítulo se ha podido observar que en pocos artículos se menciona de forma clara a las tareas en Tiempo Real y como se representan. Sin embargo algunos coinciden en que tienen al menos dos componentes, estando siempre presente el tiempo de arribo. Para el presente documento se tomara a la tarea en Tiempo Real como compuesta de tres elementos: tiempo de arribo, tiempo de ejecución y plazo. Es importante señalar que de estas características básicas se toman las consideraciones para calcular por ejemplo el tiempo de inicio, el tiempo de finalizado y otros, ya que en estos casos además de las tareas en Tiempo Real intervienen las características del equipo de cómputo y la forma en que se da atención a las tareas.

# CAPÍTULO 4 **Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes**

## **4.1 Introducción**

En el presente capítulo se desarrolla el modelo multivariable para tiempos de interarribo de tareas en Tiempo Real concurrentes. Este modelo se fundamenta en las concordancias con los datos obtenidos en el capítulo anterior y se busca tener los comportamientos descritos en los artículos mencionados. Con dicho modelo se busca representar las características más importantes de los tiempos de arribo de las tareas en Tiempo Real.

## **4.2 Definiciones Preliminares para la Presentación del Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes**

Las siguientes definiciones presentan algunas modificaciones con respecto a [Cruz2004] y [MGC2006] y servirán de base para proponer el modelo.

**Definición 1. (Tarea o Tarea en Tiempo Real)** Una Tarea en Tiempo Real (TTR) es una entidad ejecutable de procesamiento que al menos se caracteriza por un tiempo de arribo, un tiempo de ejecución y una restricción temporal y se representa con  $J_i$ , (donde la  $i$  significa “tarea  $i$ -ésima”). Cada TTR está formada por un conjunto de

## Capítulo 4. Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes

instancias  $j_{i,k}$ , tal que  $J_i = \{j_{i,k}\}$  con  $i, n \in Z^+$  (donde la  $k$  significa la  $k$ -ésima instancia).

**Definición 2. (Instancia)** Una instancia  $j_k$  es una unidad de trabajo de una tarea  $J_i$ , queda definida con  $j_{i,k} = (\pi_k, c_k, d_k)$  para  $i, k \in Z^+$  donde  $\pi_k$  es el tiempo de arribo relativo o interarribo de cada instancia; " $c_k$ " es el tiempo de ejecución de cada instancia y  $d_k$  es el plazo relativo de cada instancia.

**Definición 3. (Tiempo de arribo absoluto de una TTR).** El tiempo de arribo absoluto  $l_k$  de una instancia está definido como el tiempo en que la instancia pide atención al procesador en relación con el origen temporal de referencia. Tal arribo corresponde al índice  $k$  de la instancia  $j_{i,k}$  correspondiente a la tarea  $J_i$ .

**Definición 4. (Tiempo de arribo relativo o inter-arribo de una TTR).** El tiempo de interarribo o tiempo de arribo relativo  $\pi_k$  de una TTR en su instancia con índice  $k$  está definido como el tiempo en que dicha instancia pide atención al procesador, tomando como referencia el tiempo de arribo inmediato anterior o, de otra forma de la instancia con índice  $k-1$ . En el caso de muestreo de variables, El tiempo de inter-arribo puede estar relacionado con el criterio de Nyquist (ver [Nyq28]).

**Definición 5. (Tiempo de inicio de una TTR).** El tiempo de inicio  $s_k$  de una TTR  $J_i$  en su instancia con índice  $k$  está definido como el instante en que la instancia es atendida por el procesador con relación al origen temporal de referencia. El tiempo de inicio siempre será mayor o igual al tiempo de arribo absoluto o relativo, según la consideración.

#### Capítulo 4. Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes

**Definición 6. (Tiempo de ejecución de una TTR).** El tiempo de ejecución  $C_k$  de una TTR  $J_i$  en su instancia con índice  $k$  es el tiempo en que la instancia es ejecutada en el procesador sin tomar en cuenta cambios de contexto en caso de que existan, cumple con obtener una respuesta considerada correcta en algún sentido predefinido y representa un lapso de tiempo en el que no se consideran los desalojos de la instancia por el procesador.

**Definición 7. (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 totalmente su ejecución, tomando en cuenta los cambios de contexto si es que se realizaron.

**Definición 8 (Plazo ( $D_n$ )).** Es el tiempo máximo en el cual se desea concluir con los cálculos o uso de recursos y así cumplir con la restricción de tiempo impuesta por la dinámica del sistema.

**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 en el cual o después del cual la instancia con índice  $k$  de una TTR  $J_{i,n}$  debe finalizar.

**Definición 10. (Plazo absoluto máximo de una TTR).** El plazo absoluto máximo  $d_{k\_max}$  es el límite temporal superior en relación al origen temporal de referencia antes del cual la instancia con índice  $k$  de una TTR  $J_{i,n}$  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  $l_k$  en el cual o después del cual la instancia con índice  $k$  de una TTR  $J_{i,n}$  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 del cual la instancia con índice  $k$  de una TTR  $J_{i,n}$  debe finalizar.

### 4.3 Propuesta del Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes

A continuación se partirá de la definición de sistema de eventos discretos para darle forma al modelo de tiempos de arribo de tareas en Tiempo Real.

Un Sistema de Eventos Discretos (SED) [PB98] es un sistema dinámico que evoluciona en el tiempo con ocurrencias de eventos o posibles intervalos de tiempo irregulares. En este contexto, un sistema en Tiempo Real es un caso particular de un sistema de eventos discretos porque su comportamiento queda completamente representado como sigue:

**Definición 1. (Tareas en Tiempo Real Concurrentes).** Un Conjunto  $G$  de Tareas en Tiempo Real Concurrentes  $\{J_{ij}\}$ , está definido por un sistema de eventos discretos tal que:

$$G = (X, \varepsilon, f_e, g, E_v)$$

Donde:

$X$  : Es conjunto de valores de los tiempos de interarribo  $\{\pi_{i,k}\}$ ,  $\forall i, k \in N^+$ .

$\varepsilon$  : Es el conjunto de TTR (Tareas en Tiempo Real)  $\{J_i\}$ ,  $\varepsilon = \{J_i\}$

$f_e$  : Es la función de transición entre estados  $f_e : \pi_k \rightarrow \pi_{k+1}$



#### Capítulo 4. Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes

$g$  : Función que mapea de  $X$  a  $P(X)-\{0\}$ ,  $g: X \rightarrow P - \{0\}$ , donde  $P$  representa el conjunto potencia.

$E_v$ : Conjunto de trayectorias posibles desde  $\pi_{i,j}$  hasta  $\pi_{i,k+m}$  con  $i, k, m \in \mathbb{N}^+$ .

En el sistema quedan contenidas las características de cada una de las variables que intervienen en los TA de las TTR.

Lo anterior define el ambiente de las TTR y le da las características de un Sistema de Eventos Discretos. Ampliando las características de los elementos anteriormente mencionados tenemos:

$X$ :

Es el conjunto de valores de los TI  $\{\pi_{i,k}\}$ ,  $\forall i, k \in \mathbb{N}^+$ .  $X = \{\pi_i\}$ ,

$X \subset \mathbb{R}^+ \cup \{0\}$ ,  $k \in \mathbb{R}^+$ ,

Es un conjunto acotado por ambos lados,  $0 < x < \infty$ ,  $\forall x \in X$ . Los valores de  $X$ , deben contener todos los valores posibles de TI, en otras palabras, debe contener los subconjuntos  ${}_1X, {}_2X, \dots, {}_nX$  donde  ${}_mX$ ,  $\forall m \in \mathbb{Z}^+$  representa el subconjunto de los valores probables de TI para la tarea "m". Cada conjunto  ${}_mX$ ,  $\forall m \in \mathbb{Z}^+$  queda definido por una tarea, de la cual va a representar todos sus los valores posibles en cuanto a TI.

$\varepsilon$  :

Es el conjunto de TTR (Tareas en Tiempo Real)  $\{{}_iJ\}$ ,  $\varepsilon = \{{}_iJ\}$ ,  $\forall i \in \mathbb{N}^+$ ,

$0 < i$ ,

#### Capítulo 4. Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes

Cada TTR  $_i J$ ,  $\forall i \in N^+$ ,  $0 < i$ , tiene tres componentes que son un Tiempo de Interarribo, un Tiempo de Ejecución y un Plazo o sea:  $_i J = \{(_i \pi_k, _i c_k, _i d_k)\}$ ,  $\forall i, k \in N^+$ ,  $0 < i$

$f_e$ :

Es la función de transición entre estados  $f_e : \pi_k \rightarrow \pi_{k+1}$

El mapeo se hace con un modelo AR de primer orden y se comporta como un proceso markoviano. Para una sola TTR se puede representar por las ecuaciones

$$f_e : \pi_k \rightarrow \pi_{k+1} \quad (4.1)$$

$$f_e : \Pi_k \rightarrow \Pi_{k-1}$$

donde  $\pi_k$  representa una sola tarea y  $\Pi_k$  representa múltiples tareas. Todos los valores que se obtengan deben estar contenidos en X y dado que son valores estocásticos estacionarios, deben estar representados por un valor medio (media aritmética) y máxima desviación con repesto a éste.

$g$  :

Función que mapéa de X a  $P(X) - \{0\}$

$g : X \rightarrow P - \{0\}$ , donde  $P$  representa el conjunto potencia.

$E_v$ :

Conjunto de trayectorias posibles desde  $\pi_{i,j}$  hasta  $\pi_{i,k+m}$  con  $i, k, m \in N^+$ .

$$E_v = \{\pi_1, \pi_2, \pi_3, \dots\}$$

$E_a$ : Conjunto de trayectorias posibles de  $\pi_{i,j} \rightarrow \pi_{i,k+m}$  (que se aplican al caso en específico) con  $i, k, m \in N^+$ .

### Estabilidad

La estabilidad en este sistema de eventos discretos debe abordarse considerando que los valores que se producen específicamente por la función de transición deben ser acotados por un máximo y un mínimo o en el caso de representar los tiempos de arribo absolutos, tener alguna razón de aumento con significado válido. Por tanto el sistema se considera estable por el criterio de “estradas acotadas, salidas acotadas” (BIBO por sus siglas en inglés) y para conocerlas se deben conocer los valores máximos y mínimos para la función de transición, mismas que son los límites del conjunto  $X$  o las razones de cambio cada vez que se produzca un aumento en el valor [Oga98].

### Propuesta de estabilidad

Para un sistema cualquiera la estabilidad reside en la relación del comportamiento de las salidas con respecto a las entradas, lo que se conoce como función de transferencia [Oga98][[]]. En el caso del Sistema de Eventos Discretos el conjunto  $X$  representa los valores posibles para las salidas del sistema y la función de transición nos da la forma en que van sucediendo.

Dado que el conjunto  $X$  de valores posibles de salida es representativo del comportamiento de los tiempos de interarribo, estos deben manifestar el comportamiento de los tiempos de interarribo de tareas en Tiempo real de tipo periódicas, aperiódicas.

En los que coinciden varios autores es que los valores que se producen para los casos de TTR periódicas y aperiódicas están acotados, y en sus artículos proponen distintas

#### Capítulo 4. Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes

funciones para representar su comportamiento, que en el modelo de Eventos Discretos representa la función de transición

Por lo anterior se puede decir que la estabilidad del modelo es un mapeo de los valores producidos por la función de transición al conjunto  $X$  de valores posibles.

La función de transición  $f_e$  es un mapeo como sigue

$$f_e : \pi_k \rightarrow \pi_{k+1} \quad (4.2)$$

El sistema es estable si se cumple

$$\pi_k \in X \text{ y } \pi_{k+1} \in X, \forall k \in Z^+ \quad (4.2)$$

Donde:

$X$  representa los valores posibles para los tiempos de interarribo.

Y para los casos de tiempos de arribo absolutos,

$$f_e \in X, \{\Delta l\} \subset X_{\text{int}} \quad (4.2)$$

Donde:

$X_{\text{int}}$  representa los valores de la razón de cambio  $\Delta l$  entre arribos absolutos consecutivos.

Por lo tanto el sistema es estable siempre que alguno de los criterios anteriores se cumpla.

A continuación se tratarán dos criterios de estabilidad que se pueden aplicar a los tiempos de interarribo y posteriormente se abordará un tercer criterio relacionado con la noción de estabilidad para el caso específico de tiempos de arribo absolutos.

### **Estabilidad del Modelo para tiempos de arribo de tareas en Tiempo Real Concurrentes en sentido de Lyapunov**

Un conjunto invariante cerrado  $X_m \subset X$  de  $G$  es estable en el sentido de Lyapunov [PB98] con respecto a  $E_a$  si para cualquier  $\varepsilon > 0$  es posible encontrar algún  $\delta > 0$  tal que cuando  $\rho(x_0, X_m) < \delta$  se tiene:

$$\rho(X(x_0, E_k, k), X_m) < \varepsilon \quad \forall E_k \quad (4.2)$$

Tal que

$$E_k E \in E_a(x_0), \quad k \geq 0 \quad (4.3)$$

Donde

$X_i \subset X$  es el subconjunto de estados posibles para cada tarea  $J_i$  (valores posibles de los tiempos de interarribo  $\pi_{i,k}$ ,  $\forall i, k \in Z^+$ ),

Es estable en sentido de Lyapunov con respecto a  $E_a$  como conjunto de trayectorias permitidas,

$E_k E$  representa una concatenación de cadenas.

$X(x_0, E_k, k)$  representa el estado final de la secuencia de eventos que comienza en  $x_0$  y termina en  $E_k$ .

$$X = \{\pi_k\}, \quad k \in N^+, \quad \pi_k \in R^+ \quad (4.4)$$

#### Capítulo 4. Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes

$$X = \{\pi_k\} \text{ con } \pi_{\min} \leq \pi_k \leq \pi_{\max}, \forall k \in N \quad (4.5)$$

De lo anterior se tiene que la estabilidad de los interarribos  $\pi_{i,k}, \forall i, k \in Z^+$  de TTR concurrentes  $J_i, \forall i \in Z^+$  depende de la permanencia de todos y cada uno de los valores dentro del subconjunto  $X_i$ .

De la ecuación:

$$\rho(X(x_0, E_k, k), X_i) < \varepsilon \quad \forall E_k \quad (4.6)$$

Se interpreta que la estabilidad del sistema está ligada a una región acotada por la distancia máxima de la trayectoria  $X(x_0, E_k, k)$  al conjunto  $X_i$ , siempre y cuando la distancia del primer valor de dicha cadena se encuentre en una región cercana al conjunto  $X_i$  y acotada por  $\delta$  como se indica en  $\rho(x_0, X_i) < \delta, \forall \delta \in R^+$ .

Como ya se tiene definidos los máximos y los mínimos posibles en los tiempos de interarribo es posible establecer la desigualdad

$$|x_0 - X_i| < |\pi_{\max} - \pi_{\min}| \quad (4.7)$$

y

$$|x_{m-1} - x_i| < |\pi_{\max} - \pi_{\min}| \quad (4.8)$$

Donde:

$|\pi_{\max} - \pi_{\min}|$  es la máxima diferencia posible entre los tiempos de interarribo

$|x_0 - x_i|$  es la diferencia entre el valor inicial de los tiempos de interarribo y el valor en la instancia con índice “ $i$ ”

## Estabilidad del Modelo para tiempos de arribo de tareas en Tiempo Real Concurrentes en un Sentido Amplio

Otro punto de vista de estabilidad se menciona en [Hay02] como siguiente:

En un sistema estocástico discreto estrictamente estacionario, la función de valor medio

$$\mu(n) = E[u(n)] \quad (4.9)$$

donde  $E$  denota el operador de esperanza estadística, la autocorrelación,

$$r(n, n-k) = E[u(n)u^*(n-k)], \quad k = 0, \pm 1, \pm 2, \dots, \quad (4.10)$$

donde el asterisco denota el complejo conjugado y al covarianza,

$$c(n, n-k) = E[(u(n) - \mu(n))(u(n-k) - \mu(n-k))^*], \quad k = 0, \pm 1, \pm 2, \dots, \quad (4.11)$$

asumen formas simples. Por ejemplo el valor medio asume un valor constante, por lo que se puede escribir:

$$\mu(n) = \mu, \quad \forall n \quad (4.12)$$

También se encuentra que las funciones de autocorrelación y la covarianza dependen solamente de la observación en el momento  $n$  y en  $n-k$ , o sea dependen de la secuencia de intervalos desde  $k$  hasta  $n$

$$r(n, n-k) = r(k), \quad (4.13)$$

$$c(n, n - k) = c(k) \quad (4.14)$$

Las ecuaciones anteriores no son suficientes para garantizar que el proceso es estrictamente estocástico, pero un proceso estocástico que es “no estrictamente estacionario” y que cumple estas condiciones se dice que es estacionario en un sentido amplio (wide-sense stationary) o estacionario de segundo orden. Un proceso estrictamente estacionario, es estacionario en sentido amplio [Dob53] si y solo si:

$$E[u(n)^2] < \infty, n \in Z^+ \quad (4.15)$$

La condición anterior se obtiene con  $r(n, n - k) = E[u(n)u^*(n - k)]$  cuando  $k=0$ , o sea

$$r(n, n) = E[u(n)u^*(n)] = E[u(n)^2] \quad (4.16)$$

aplicándolo al caso del sistemas de eventos discretos antes mencionado, ya que los valores de todos los tiempos de interarribo estan contenidos y acotados por valores conocidos, los valores de las esperanzas al menos están acotados por los valores máximos y mínimos de los tiempos de interarribo, que siempre son menores que infinito.

### **Estabilidad para los tiempos de arribo absolutos**

Para los tiempos de arribo absolutos, el criterio de estabilidad cambia, ya que el tiempo de arribo absoluto va aumentando conforma se incrementa el valor de  $k$ , por lo que los límites de  $X$  no están acotados.



#### Capítulo 4. Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes

Para este caso el criterio de estabilidad se basa en la noción de que el incremento del valor del tiempo de arribo absoluto para la instancia  $k$  aumenta de forma acotada, en otras palabras:

$$\Delta l_k = l_k - l_{k-1} \quad (4.17)$$

Dando como resultado a los tiempos de interarribo

$$\Delta l_k = \pi_k = l_k - l_{k-1} \quad (4.18)$$

Y por tanto:

$$\pi_k < \Delta l_k < \pi_k \quad (4.19)$$

Lo anterior limita los incrementos de los tiempos de arribo absolutos  $\Delta l$  en el instante  $k$  a los valores contenidos en el conjunto  $X$  (tiempos de interarribo), los cuales sí tienen cotas y delimita los valores posibles de  $\Delta l_k$  dentro de un criterio de estabilidad:

$$\pi_{\min} < \Delta l_k < \pi_{\max} \quad (4.20)$$

Lo anterior implica que el conjunto  $X$  incluye los valores posibles de  $\pi_k$  y los valores posibles de  $l_k$ .

Los valores posibles para los tiempos de interarribo  $\pi_k$  se encuentran contenidos en un subconjunto de  $X$ :

$$\pi_k \in X_\pi, X_\pi \subset X, \forall k \in Z^+ \quad (4.21)$$

Y

$$X_\pi = [\pi_{\min}, \pi_{\max}) \quad (4.22)$$

## **Capítulo 4. Modelo para Tiempos de Arribo de Tareas en Tiempo Real Concurrentes**

Por lo anterior podemos concluir que el sistema es estable siempre que se cumpla con que los valores producidos por la función de transición estén contenidos en el conjunto acotado de valores posibles de los tiempos de interarribo.

### **Conclusiones**

En el presente capítulo se presentó un modelo matemático para representar tiempos de interarribo. Mostró que el modelo tiene la capacidad de representación de un sistema monotarea así como modelos multitareas y que lo importante es conocer las cotas máximas y mínimas posibles para los tiempos de interarribo para poder determinar la estabilidad del sistema. Se presentaron adicionalmente tres criterios de estabilidad del sistema de eventos discretos, dos para los tiempos de interarribo y uno para los tiempos de arribo absolutos.

# CAPÍTULO 5 Ejemplos y Resultados Experimentales

## Introducción

En el presente capítulo se aplicará el modelo de representación para apreciar su uso y el significado de sus elementos. Se usarán como referencias algunos de los trabajos donde se menciona el modelo de los tiempos de interarribo. Es de gran importancia recalcar que la representación con el sistema de eventos discretos abarca la forma de encontrar los tiempos de arribo, así como los valores que se deben representar.

## Ejemplos

Partiendo de la definición de eventos discretos:

$$G = (X, \varepsilon, f_e, g, E_v) \quad (5.1)$$

Donde:

$X$  : Es conjunto de valores de los tiempos de interarribo  $\{\pi_{i,k}\}$ ,  $\forall i, k \in N^+$ .

$\varepsilon$  : Es el conjunto de TTR (Tareas en Tiempo Real)  $\{J_i\}$ ,  $\varepsilon = \{J_i\}$

$f_e$  : Es la función de transición entre estados  $f_e : \pi_k \rightarrow \pi_{k+1}$

$g$  : Función que mapea de  $X$  a  $P(X)-\{0\}$ ,  $g : X \rightarrow P - \{0\}$ , donde  $P$  representa el conjunto potencia.

$E_v$  : Conjunto de trayectorias posibles desde  $\pi_{i,j}$  hasta  $\pi_{i,k+m}$  con  $i, k, m \in N^+$ .

Podemos hacer las interpretaciones siguientes

### Ejemplo 1

Con los datos siguientes, representar las tareas en Tiempo Real con el modelo de eventos discretos y determinar si el sistema es estable.

Número de tareas:  $n, \forall n \in Z^+$

Tiempos de interarribo:  $f_e : \Pi_k = A(\Pi_{k+1} - W_{k-1}) + U + W_k, \forall k \in Z^+$

valores posibles de los tiempos de interarribo:  $\Pi_{i,k} \in [\Pi_{\min}, \Pi_{\max}]$

### Solución

Dado que la función de los tiempos de interarribo puede representar de una a múltiples tareas, se desarrollará partiendo de la ecuación representativa para los arribos de una sola TTR y posteriormente se generalizará para múltiples tareas.

- $X$ : conjunto de valores para los tiempos de interarribo  $\pi_{i,k} = [\pi_{\min}, \pi_{\max}]$ , donde el subíndice  $i$  representa la tarea y el subíndice  $k$  representa la instancia.

Dado que el modelo de tiempos de interarribo debe representar cierto comportamiento de momentos en que se solicita servicio al procesador y simula el comportamiento de un sistema implantado, las cotas máxima y mínima son conocidas.

**Máximos y Mínimos para los Valores Posibles de Tiempos de Interarribo**

La ecuación  $f_e : \Pi_k = A(\Pi_{k+1} - W_{k-1}) + U + W_k$  para una sola tarea  $f_e : \pi_k = a(\pi_{k+1} - w_{k-1}) + u + w_k$  representa un proceso estocástico con la característica de ser estacionario, por tanto se tiene que cada una de las tareas cumple con:

$$\pi_k = u + \hat{w}_k \tag{5.2}$$

Donde  $\hat{w}_k$  representa las variaciones provocadas por el parámetro de la ecuación y los ruidos inherentes al sistema.

$$\hat{w}_k = a_k(\pi_{k-1} - w_{k-1}) + w_k \tag{5.3}$$

Y dado que  $\pi_k$  está definido como un proceso estocástico estacionario cumple con tener una distribución de probabilidad constante a lo largo del tiempo igualmente es una cadena de Markoff ya que cumple con las condiciones de

$$P\{\pi_n = x_{i,n} | \pi_{n-1} = x_{i,n-1}, \dots, \pi_1 = x_{i,1}\} = P\{\pi_n = x_{i,n} | \pi_{n-1} = x_{i,n-1}\} \tag{5.4}$$

Donde  $P\{\}$  representa la probabilidad condicional.

**El valor mínimo de  $\pi_k$  se produce cuando**

- En el caso de TTR periódicas, donde el parámetro “a” es constante se tiene que el valor mínimo de los tiempos de interarribo se produce cuando

$$\hat{w}_k = 0 \quad (5.5)$$

Por lo que la ecuación

$$\pi_k = u + \hat{w}_k = u + 0 \quad (5.6)$$

Se convierte en

$$\pi_{\min} = \pi_k = u \quad (5.7)$$

- En el caso de TTR esporádicas, donde el valor de “a” es variable

$$\hat{w}_k = 0, \hat{w}_{k-1} = 0 \text{ y } a = a_{\min} \quad (5.8)$$

Por lo que la ecuación (1) se convierte en

$$\pi_k = a_{\min}(\pi_{k-1}) + u_k \quad (5.9)$$

El valor del tiempo de arribo mínimo queda en la vecindad del valor de  $u_k$ .

**El valor máximo de  $\pi_k$  se encuentra cuando**

Para tareas periódicas, de la ecuación (1) se tiene:

$$\hat{w}_k = \hat{w}_{\max} = a(\pi_{\max} - w_{\min}) + w_{\max} \quad (5.10)$$

Por lo que la ecuación de TI queda como sigue

$$\pi_{\max} = a(\pi_{\max} - w_{\min}) + w_{\max} + u \quad (5.11)$$

Si

$$w_{\min} = 0 \quad (5.12)$$

Entonces

$$\pi_{\max} = a\pi_{\max} + w_{\max} + u \quad (5.13)$$

Para tareas esporádicas, el valor del parámetro  $a$  es variable, por lo que de la ecuación (1) se tiene:

$$\hat{w}_k = \hat{w}_{\max} = a_{\max}(\pi_{\max} - w_{\min}) + w_{\max} \quad (5.14)$$

Y la ecuación de TI

$$\pi_{\max} = a_{\max}(\pi_{\max} - w_{\min}) + w_{\max} + u \quad (5.15)$$

Si

$$w_{\min} = 0 \quad (5.16)$$

Entonces

$$\pi_{\max} = a_{\max}\pi_{\max} + w_{\max} + u \quad (5.17)$$

Ahora que se encuentran acotados los valores posibles de  $\pi_k$  es posible definir el mapeo de los tiempos de interarribo por

$$\pi_k : \pi \rightarrow X \quad \text{ó} \quad \pi_k : k \rightarrow X \quad (5.18)$$

Donde:

$\pi$  representa  $\{\pi_k\}, \forall k = 1, 2, 3, \dots,$

$X \in R^+$  y representa el conjunto de todos los valores posibles que puede tomar  $\pi_k$ .

Para múltiples tareas se tiene que

$$\begin{aligned} {}_1\pi_k : \pi \rightarrow {}_1X \text{ ó } {}_1\pi_k : k \rightarrow {}_1X, \text{ para } {}_1X \subset X & \quad (5.19) \\ {}_2\pi_k : \pi \rightarrow {}_2X \text{ ó } {}_2\pi_k : k \rightarrow {}_2X, \text{ para } {}_2X \subset X & \\ \vdots & \\ {}_n\pi_k : \pi \rightarrow {}_nX \text{ ó } {}_n\pi_k : k \rightarrow {}_nX, \text{ para } {}_nX \subset X & \end{aligned}$$

Donde:

$\pi$  representa  $\{\pi_k\}, \forall k \in N^+,$

$X \in R^+$  y representa el conjunto de todos los valores posibles que puede tomar cualquier TI.

${}_1X, {}_2X, \dots, {}_nX \subset X$  cada uno de ellos representa el conjunto de valores posibles de las tareas  ${}_1J, {}_2J, \dots, {}_nJ$  respectivamente.

Dado que los valores que puede tomar cada uno de los TI de cada TTR queda contenido en un conjunto acotado, cada valor tiene una probabilidad que en suma deben de dar máximo “1” en cada subconjunto, por lo que se tiene un espacio de probabilidad, esto es:

$$\begin{aligned} X &= \{{}_1X, {}_2X, \dots, {}_nX\} & (5.20) \\ {}_1X &= \{x_1, x_2, \dots, x_m\} \\ {}_2X &= \{y_1, y_2, \dots, y_m\} \\ {}_nX &= \{z_1, z_2, \dots, z_m\} \end{aligned}$$



Se agrega ahora una medida a cada uno de los subconjuntos  $({}_n X, p)$ ,  $\forall n \in Z^+$  donde,  $p$  representa una medida y además es un mapeo

$$p : X \rightarrow R^+ \quad (5.21)$$

Como lo que se desea representar en cada subconjunto  ${}_n X$  es la ocurrencia de los TI, cada valor presente en dicho subconjunto debe tener cierta probabilidad dado que cada subconjunto debe representar el éxito de un evento para una variable estocástica, por tanto la máxima métrica es la del conjunto en sí y es de un valor  $p \leq 1$ .

Por lo que cumple con:

$$p(X) < \infty, p(\inf |X|) \leq 0 \quad (5.22)$$

Y además

$$0 \leq p \leq 1 \quad (5.21)$$

con lo que se tiene un espacio de probabilidad por cada uno de los subconjuntos.

$\varepsilon$  : Es el conjunto de TTR (Tareas en Tiempo Real)  $\{J_i\}$ ,  $\varepsilon = \{J_i\}$ ,  $\forall i = 0, 1, 2, \dots, n$

Si  $i=1$ , se hablaría de una sola tarea, si  $i > 1$ , entonces se estaría hablando de múltiples tareas, por tanto se tiene:

El modelo para representar la función de transición es el presentado en [Cruz2004] dado que representa un modelo de Tiempos de Arribo (TA) para una sola TTR. Dicho modelo usa un modelo ARMA para representar los TA, Tiempos de Interarribo, Tiempos de Inicio y Tiempos de Ejecución.

Los Tiempos de Interarribo (TI que estan representados por función de transición entre estados  $f(k)$ ) quedan representados como sigue:

$$\pi_k = a_k(\pi_{k-1} - w_{k-1}) + u_k + w_k \quad (5.22)$$

Donde:

- $a_k$  es el parámetro del sistema,
- $\pi_{k-1}$  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.

El valor del parámetro " $a_k$ " puede ser constante como en el caso de las Tareas Periódicas o puede ser variable como en el caso de Tareas Esporádicas.

Esta función es representada en el modelo de eventos discretos como  $f_e$  como sigue:

$$f_e : \{\pi_k\} \rightarrow \{\pi_{k+1}\} \quad (5.23)$$

Para representar múltiples tareas basándonos en la ecuación [1], la representación de tareas independientes queda:

Para tareas " $J_1, J_2, \dots, J_n$ " tenemos que:

$$\begin{aligned}
 {}_1\pi_k &= {}_1a_k ({}_1\pi_{k-1} - {}_1w_{k-1}) + {}_1u_k + {}_1w_k \\
 {}_2\pi_k &= {}_2a_k ({}_2\pi_{k-1} - {}_2w_{k-1}) + {}_2u_k + {}_2w_k \\
 &\vdots \\
 {}_n\pi_k &= {}_na_k ({}_n\pi_{k-1} - {}_nw_{k-1}) + {}_nu_k + {}_nw_k
 \end{aligned} \tag{5.24}$$

Y por tanto es posible representar lo anterior en forma matricial como:

$$\begin{bmatrix} {}_1\pi_k \\ {}_2\pi_k \\ \vdots \\ {}_n\pi_k \end{bmatrix} = \begin{bmatrix} {}_1a_k & 0 & \cdots & 0 \\ 0 & {}_2a_k & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & {}_na_k \end{bmatrix} \begin{bmatrix} {}_1\pi_{k-1} - {}_1w_{k-1} \\ {}_2\pi_{k-1} - {}_2w_{k-1} \\ \vdots \\ {}_n\pi_{k-1} - {}_nw_{k-1} \end{bmatrix} + \begin{bmatrix} {}_1u_k \\ {}_2u_k \\ \vdots \\ {}_nu_k \end{bmatrix} + \begin{bmatrix} {}_1w_k \\ {}_2w_k \\ \vdots \\ {}_nw_k \end{bmatrix} \tag{5.25}$$

Dado que las tareas son independientes, en la matriz de parámetros sólo existen elementos diferentes de cero en la diagonal principal. Las tareas que se busca representar son periódicas y esporádicas, por lo que sus comportamientos en cuanto a Tiempos de interarribo están acotados tanto para el tiempo mas corto como para el tiempo mas largo. Lo anterior se traduce de cada una de sus definiciones.

Este modelo queda representado por la segunda función de transición

$$f_e : \{\Pi_k\} \rightarrow \{\Pi_{k+1}\} \tag{5.26}$$

La ecuación (1) representa valores de tiempos de interarribo estocásticos con elementos acotado y que deben estar definido por un valor medio aritmético (primer momento de probabilidad) y una varianza (segundo momento de probabilidad).

Para tareas dependientes se tiene una relación entre los arribos de dos o mas tareas. Dado que una tarea puede decidir cual otra tarea ejecutar, puede

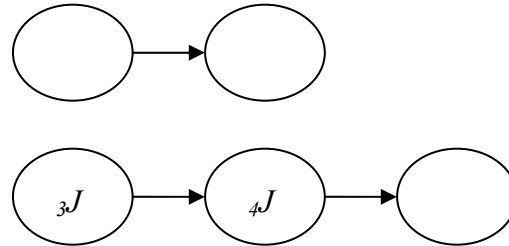


Figura 5- 1.- Gráfica de Tareas dependientes

representarse dependencias con grafos dirigidos o con matrices de precedencia:

	$1J$	$2J$	$3J$	$4J$	$5J$
$1J$		1			
$2J$					
$3J$				1	2
$4J$					1
$5J$					

Tabla 5- 1.- Matriz de adyacencias para tareas dependientes

Donde el valor “1” representa precedencia inmediata, los valores mayores a “1” indican la “distancia” en tareas que preceden. Las casillas vacias indican la no dependencia y tienen valor igual a cero (“0”), en la tabla se dejan en blanco para resaltar las dependencias.

Los tiempos de arribo de las TTR dependientes tienen la característica de depender de los tiempos de finalizado de las tareas precedentes, por lo que su representación debe relacionar los tiempos de finalizado de otras TTR.

Tenemos en el ejemplo anterior cinco tareas  ${}_1J, {}_2J, {}_{31}J, {}_{41}J, {}_5J$ , por lo que  ${}_1J = \{\pi_k\}, {}_2J = \{\pi_k\}, \dots$

$$\begin{bmatrix} {}_1\pi_k \\ {}_2\pi_k \\ {}_3\pi_k \\ {}_4\pi_k \\ {}_5\pi_k \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}_1f_k \\ {}_2f_k \\ {}_3f_k \\ {}_4f_k \\ {}_5f_k \end{bmatrix} + \begin{bmatrix} {}_1\gamma_k \\ {}_2\gamma_k \\ {}_3\gamma_k \\ {}_4\gamma_k \\ {}_5\gamma_k \end{bmatrix} \quad (5.27)$$

Donde cada uno de los tiempos  ${}_nf_k$  representa el tiempo de finalizado de la tarea “n” en la instancia “k”.

Representamos los tiempos de finalizado con los tiempos de ejecución

$${}_nf_k = {}_nc_k + {}_ns_k + {}_n\omega_k \quad (5.28)$$

Donde  ${}_nc_k$  es el tiempo de ejecución,  ${}_ns_k$  es el tiempo de inicio de la tarea “n” en la instancia “k” y  ${}_n\omega_k$  son consideraciones temporales adicionales.

$${}_1c_k = {}_1b_k ({}_1c_{k-1} - {}_1\omega_{k-1}) + {}_1c_k + {}_1\omega_k \quad (5.29)$$

$f_e$ : Es la función de transición entre estados  $f_e = \pi_k = a(\pi_{k+1} - w_{k-1}) + u + w_k$ ,

Donde ;

$\pi_k$  es el tiempo de interarribo para la instancia k,

$\pi_{k-1}$  es el tiempo de interarribo inmediato anterior,

$w_k$  es el ruido en la instancia k,

$w_{k-1}$  es el ruido en la instancia inmediata anterior k-1,

u es la entrada al sistema o el número de referencia k,

$a$  es el parámetro de sistema,

Y se cumple también para tareas múltiples  $f_e = \Pi_k = A(\Pi_{k+1} - W_{k-1}) + U + W_k$

$g$  : Función que mapea de  $X$  a  $P(X) - \{0\}$ ,  $g : X \rightarrow P - \{0\}$ , donde  $P$  representa el conjunto potencia.

$E_v$  : Conjunto de trayectorias posibles desde  $\pi_{i,j}$  hasta  $\pi_{i,k+m}$  con  $i, k, m \in \mathbb{N}^+$ .

Por tanto la representación es

$$G = (X, \varepsilon, f_e, g, E_v) \quad (5.30)$$

Donde:

$X$ : Es el conjunto de valores posibles de los tiempos de interarribo

$$X = \{\pi_{i,k}\} = \{\pi_{i,\min}, \pi_{i,\max}\}, \forall i, k \in \mathbb{N}^+.$$

$\varepsilon$  : Es el conjunto de TTR (Tareas en Tiempo Real)  $\{J_i\}$ ,  $\varepsilon = \{J_i\}$ ,  $\forall i \in \mathbb{N}^+$

$f_e$ : Es la función de transición entre estados

$$f_e = \Pi_k = A(\Pi_{k+1} - W_{k-1}) + U + W_k \text{ o } f_e : \pi_k = a(\pi_{k+1} - w_{k+1}) + u + w_k$$

$g$  : Función que mapea de  $X$  a  $P(X) - \{0\}$ ,  $g : X \rightarrow P - \{0\}$ , donde  $P$  representa el conjunto potencia.

$E_v$  : Conjunto de trayectorias posibles desde  $\pi_{i,j}$  hasta  $\pi_{i,k+m}$  con  $i, k, m \in \mathbb{N}^+$ .

El sistema es estable dado que se considera una cota mínima así como una cota máxima para el conjunto  $X$ , siempre que el valor del parámetro  $a$  en la función de transición entre estados, sea menor que 1 y produzca un salida acotada por  $\{\pi_{i,k}\} = [\pi_{\min}, \pi_{\max}]$ .

Gráficas de la simulación de tiempos de arribo para una TTR:

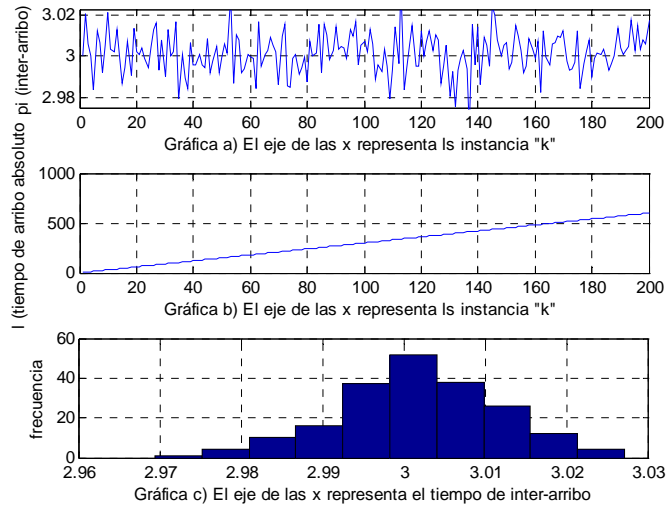


Figura 4.- Gráficas de la simulación de tiempos de arribo para una TTR periódica

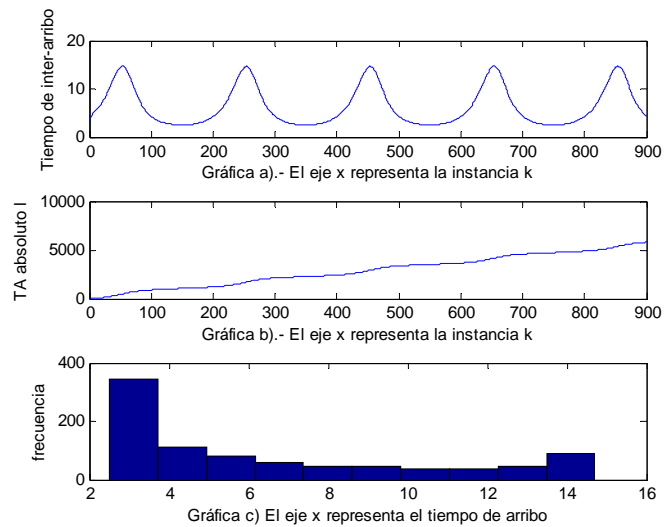


Figura 5.- Gráficas de la simulación de tiempos de arribo para una TTR periódica

## Ejemplo 2

Sea una Tarea en Tiempo Real  $J$  con  $n \in \mathbb{N}^+$  instancias y una función de tiempos de arribo absolutos  $l_k = l_{k-1} + \pi_k$  (para  $l_0 = 0$ ) y  $\pi_k = [\pi_{\max}, \pi_{\min}]$ ; modelar la tarea como un evento discreto.

### Solución

Para el valor del conjunto  $X$  se considera que debe incluir los valores posibles de los tiempos de interarribo  $\pi_k = [\pi_{\max}, \pi_{\min}]$  y los valores subsecuentes de  $l_k$ , por lo tanto

$$\pi_{\min} \leq l_1 < \pi_{\max} \quad (5.31)$$

$$l_1 \in X_\pi \quad (5.32)$$

Lo cual hace que el conjunto de tiempos de interarribo sea un subconjunto de los tiempos de arribo absolutos:

$$X = \{l_k\} = \{X_\pi, l_k\}, \quad \forall i, k \in \mathbb{N}^+,$$

$$\varepsilon = \{J\} \text{ porque es una sola tarea,}$$

$$f_e: l_{k-1} \rightarrow l_k,$$

$$f_e = l_k \Rightarrow f_e = l_{k-1} + 2,$$

$$g = \{l_k\} \rightarrow P(l_k) - \{0\}$$

$$E_v = \{ l_1 \rightarrow l_2, l_2 \rightarrow l_3, l_{k-1} \rightarrow l_k,$$

$$l_1 \rightarrow l_2, l_1 \rightarrow l_2 \rightarrow l_3, l_1 \rightarrow \dots \rightarrow l_n,$$

$$\}$$



### Análisis de estabilidad

El criterio de estabilidad se basa en la noción de que el incremento del valor del tiempo de arribo absoluto para la instancia  $k$  aumenta de forma acotada, en otras palabras:

$$\Delta l_k = l_k - l_{k-1} \quad (5.33)$$

Dando como resultado a los tiempos de interarribo

$$l_k - l_{k-1} = (l_{k-1} + \pi_k) - l_{k-1} \quad (5.34)$$

$$l_k - l_{k-1} = \pi_k \quad (5.35)$$

Y por tanto:

$$\pi_k < \Delta l_k < \pi_{k+1} \quad (5.36)$$

Lo anterior limita los incrementos de los tiempos de arribo absolutos  $\Delta l$  dentro de un criterio de estabilidad y por tanto es un sistema estable.

### Ejemplo 3

Sea una Tarea en Tiempo Real  $J$  con  $n \in \mathbb{N}^+$  instancias y una función de tiempos de arribo absolutos  $l_k = l_{k-1} + \pi_k$  y  $\pi_k = 2$  U.T. (Unidades Temporales); modelar la tarea como un evento discreto y obtener las gráficas correspondientes.

**Solución**

$l_k = l_{k-1} + \pi_k$  y  $\pi_k = 2$  U.T. (Unidades Temporales) representa los tiempos de arribo absolutos para una tarea con comportamiento periódico, con periodo 2 U.T. El conjunto X se define como  $X = \{l_k\}$ , donde  $l_k = k * 2, \forall k \in N^+$ .

$$X = \{l_k\} \quad \forall i, k \in N^+,$$

$\varepsilon = \{J\}$  porque es una sola tarea,

$$f_e: l_{k-1} \rightarrow l_k,$$

$$f_e = l_k \Rightarrow f_e = l_{k-1} + 2,$$

$$g = \{l_k\} \rightarrow P(l_k) - \{0\}$$

$$E_v = \{ l_1 \rightarrow l_2, l_2 \rightarrow l_3, l_{k-1} \rightarrow l_k, \\ l_1 \rightarrow l_2, l_1 \rightarrow l_2 \rightarrow l_3, l_1 \rightarrow \dots \rightarrow l_n, \\ \}$$

**Análisis de estabilidad**

Como las variaciones entre los tiempos de interarribo son:

$$\Delta l_k = l_k - l_{k-1} = 2 * k - 2 * (k - 1) \tag{5.37}$$

$$\Delta l_k = l_k - l_{k-1} = 2 \tag{5.38}$$

El sistema se considera estable porque las variaciones están contenidas en un conjunto acotado.

Dando como resultado a los tiempos de interarribo

$$\Delta l_k = \pi_k = l_k - l_{k-1} \tag{5.39}$$

Y por tanto:

$$\pi_k < \Delta l_k < \pi_{k+1} \quad (5.40)$$

Lo anterior limita los incrementos de los tiempos de arribo absolutos  $\Delta l$  en el instante  $k$  a los valores contenidos en el conjunto  $X$  (tiempos de interarribo), los cuales sí tienen cotas y delimita los valores posibles de  $\Delta l_k$  dentro de un criterio de estabilidad:

$$\pi_{\min} < \Delta l_k < \pi_{\max} \quad (5.41)$$

Lo anterior implica que el conjunto  $X$  incluye los valores posibles de  $\pi_k$  mas los valores posibles de  $l_k$ .

Los valores posibles para los tiempos de interarribo  $\pi_k$  se encuentran contenidos en un subconjunto de  $X$  :

$$\pi_k \in X_\pi, X_\pi \subset X, \forall k \in Z^+ \quad (5.42)$$

Y

$$X_\pi = [\pi_{\min}, \pi_{\max}) \quad (5.43)$$

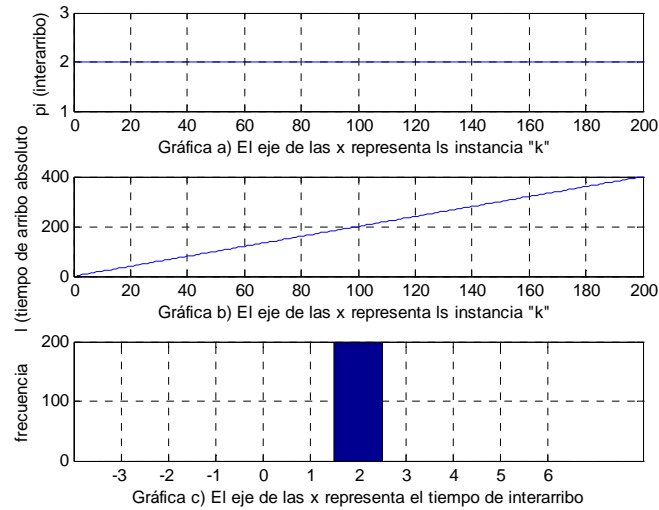


Figura 6.- Figura de tiempos de interarribo para  $l_k = l_{k-1} + \pi_k$  y  $\pi_k = 2$  U.T.

### Conclusiones

En el presente capítulo se aplicó el modelo matemático para tiempos de arribo de tareas en la representación de tres ejemplo para reresetar el comportamiento de tareas en Tiempo Real, logrando una representación completa del entorno relacionado con los tiempos de interarribo así como de los tiempos de arribo absolutos y se aplicó algún criterio para determinar su estabilidad, llegando así que aun cuando el conjunto X del modelo de los tiempos discretos no sea acotado, el sistema puede ser estable.

## Conclusiones Generales

En los primeros capítulos se revisaron las bases y varios artículos relacionados con los Sistemas en Tiempo real así como de Tareas en Tiempo real, mismos que sirvieron como base para definir las características apropiadas para la representación matemática de este trabajo. En dicho estudio se encontró que las TTR pueden quedar definidas con tres valores: tiempo de arribo, tiempo de ejecución y plazo

Los tiempos de arribo definen la forma en que los procesos solicitan tiempo de uso de procesador y un planificador debe elegir el momento en que se les va a dar esa atención.

Como los tiempos de arribo presentan diversos comportamientos, se llegó a establecer que los tiempos de arribo están contenidos en un conjunto acotado tanto para un valor máximo como para un valor mínimo y la forma en que estos valores se presenta depende de la función de transición. El sistema se considera estable siempre que la función de transición produzca valores en el intervalo definido, de otra manera el sistema se considera inestable.

Una de las ventajas del modelo es que no limita la forma en que se define la función de transición, sólo menciona que existe una y que los valores que se obtengan con ella deben estar contenidos en un intervalo acotado definido en el conjunto de estados posibles.

En el último capítulo se aplicó el modelo matemático para tiempos de arribo de tareas en la representación de tres ejemplos para representar el comportamiento de tareas en Tiempo Real, logrando una representación completa del entorno relacionado con los tiempos de interarribo así como de los tiempos de arribo absolutos y se aplicaron los criterios para determinar la estabilidad, concluyendo que el sistema es estable siempre que cada uno de los valores posibles de los tiempos de interarribo se encuentren

contenidos en un conjunto de valores acotados con un valor máximo y otro mínimo, en otras palabras el conjunto acotado representa un intervalo y para el caso de los tiempos de arribo absolutos se concluye que aun cuando el conjunto  $X$  del modelo de los tiempos discretos no sea acotado, el sistema puede ser estable cuando todos los valores de la razón de cambio  $\Delta I$  se encuentran contenidos en un conjunto acotado.

### Trabajos Futuros

Generalizar el concepto de planificador partiendo de la definición de sistema de, eventos discretos para tareas en Tiempo real, Formalización matemática del entorno de sistemas en Tiempo real.

## Glosario

**Diferencias finitas.** Aproximación de la derivada. Tangente a una curva, aproximada mediante valores de  $x$  muy pequeños donde la diferencia entre  $x_0$  y  $x_1$  (incremento  $\Delta x$ ) no tiende a cero, sino a un valor significativo.

**Firm.** Se refiere al Sistema de Tiempo Real en el que los procesos no satisfacen sus restricciones de tiempo, y ya no importa que el proceso se ejecute después; y aún es aceptable.

**Hard.** Se refiere al sistema de Tiempo Real en el que sus procesos deben satisfacer sus requerimientos de tiempo siempre.

**Instancia.** Es una unidad de trabajo  $j(k)$  de una tarea  $J_i$ , que queda definida con  $j(k)=f(l(k),c(k),d(k))$  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  $d(k)$  es el plazo de cada instancia; todo eso dentro de un intervalo de evolución con índice  $k$ .

**Kernel.** Es el núcleo del sistema operativo y es un programa que siempre está residente en memoria. Brinda recursos como: control de recursos del hardware, control de los periféricos, control de usuarios y manejo del sistema de archivos.

**Perturbación:** Es una señal que tiende afectar adversamente el valor de la salida de un sistema.

**Planificador.** Es un criterio predefinido que asigna procesos a los recursos en forma ordenada, tal que cada tarea es ejecutada hasta completarse. Un planificador puede

hacer que un proceso pase del estado listo al estado ejecutándose y viceversa, dándole un lugar específico dentro de la cola de listos.

**Plazo relativo mínimo.** Es el tiempo mínimo en el que se puede obtener una respuesta local por parte del FDTR, sin alterar negativamente la dinámica o comportamiento del sistema físico.

**Plazo relativo máximo.** Es el tiempo máximo en el que se puede obtener una respuesta local por parte del FDTR, sin alterar negativamente la dinámica del proceso.

**POSIX.** Estándares de Interfaces de Sistemas Operativos Abiertos Portables.

**Prioridad.** La prioridad es el nivel de preferencia de ejecución que tiene una tarea en un STR.

**Proceso estocástico.** El proceso estocástico  $X$  se define como una colección de variables aleatorias  $\{x_1, x_2, \dots, x_n\}$  que obedecen a una ley de distribución  $F(X)$ , tal que  $\{x_n: n \in \mathbb{N}\}$  ordenados por un conjunto de parámetros  $n$ .

**Sincronía.** Coordinación de varias causas en el mismo intervalo temporal.

**Sistema en Línea.** Es aquel sistema que siempre debe estar encendido, disponible y generalmente conectado a una red de computadoras y depende de la capacidad del hardware para atender peticiones de servicio.

**Sistema en Tiempo Real (STR).** Es aquel sistema digital que obedece tres condiciones: a) Interacción con el mundo físico, b) Emisión de una respuesta correcta de acuerdo a algún criterio preestablecido, c) Cumplimiento de las restricciones de tiempo del proceso físico con que interactúa.



**Sistema Lineal (SL).** Es aquel sistema que tiene una relación de entrada/salida y cumple las condiciones de causalidad y homogeneidad.

**Soft.** Se refiere al sistema de Tiempo Real en el que sus procesos pueden satisfacer sus restricciones de tiempo sólo en un cierto porcentaje de veces.

**SOTR.** Sistemas Operativos en Tiempo Real.

**Tarea en Tiempo Real (TTR).** Es una entidad ejecutable de trabajo  $J_i$  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  $i, k \in \mathbf{Z}^+$ .

**Tarea en Tiempo Real Periódica (TTRP).** Es aquella tarea donde todas sus instancias tienen tiempos de arribo relativo próximos a un período  $T$  y los tiempos de arribo absolutos crecen en forma monotonía respecto a la aparición de las instancias.

**Tareas Concurrentes (TC).** Son un conjunto de tareas que se ejecutan en un plazo máximo y en forma no secuencial en una computadora tomando recursos en instantes diferentes. No pueden existir tareas concurrentes sin un planificador predefinido.

**Tiempo de arribo o llegada.** También llamado tiempo de arribo absoluto, es el tiempo con respecto al inicio del proceso en que la instancia de una tarea pasa a formar parte de la cola de listos. Es decir, es un tiempo absoluto en que la instancia queda formada hasta ser atendida

**Tiempo de ejecución.** Es el tiempo que el procesador se encarga de ejecutar una instancia de una tarea (o conjunto de instancias sin interrupción).

**Tiempo de inicio o tiempo de liberación.** Es el tiempo en que la instancia de una tarea inicia su ejecución.

**Tiempo de arribo relativo de una TTR.** El tiempo de arribo relativo o tiempo de arribo relativo de una instancia  $k$  está definido como el tiempo en que la instancia pide atención al procesador en relación al arribo absoluto de la instancia  $k-1$ .

**Tiempo de finalizado.** Es el tiempo en que la instancia de una tarea termina su ejecución dentro del intervalo  $k$ .

**Tiempo prematuro.** Es el tiempo que ocurre cuando la instancia de una tarea finaliza antes de su plazo mínimo.

**Tiempo de retardo.** Es el tiempo en que la instancia de una tarea fue terminada dentro de su plazo absoluto máximo, en caso contrario, la tarea terminó después del intervalo  $k$ , y cuenta con un retardo.

**Variable aleatoria.** Se llama variable aleatoria a toda variable que, como resultado de un experimento, no toma más que un valor cualquiera, y como resultado de varios experimentos, puede tomar valores diferentes.

**Bibliografía**

- [Ada97] A. Adas, Traffic Models in Broadband Networks. 1997.
- [Abd00] T. F. Abdelzaher, A Schedulable Utilization Bound for Aperiodic Tasks, 2000.
- [ABIKRRT07] Auerbach, Baco, Iercan, Kirsch, Rajan, Röck y Trummer, *Java takes flight: Tiem-portable real-time programing with exatasks*, ACM, 2007.
- [Ash70] Ash R. “Real Analysis and Probability”, Ed. Academic Press, USA, 1970.
- [ASLZL03] T. Abdelzaher, John Stankovic, Chenyang Lu, Ronghua Zhang, Ying Lu. Feedback Performance Control in Sofware Services, 2003.
- [AMMM99] H. Aydin, R. Melhem, D. Mossé y P. Mejia-Alvarez, Optimal Reward-Based Scheduling for Periodic Real-Time Tasks, 1999.
- [Bal02] Patricia Balbastre Betoret. Modelo De Tareas Para La Integración Del Control Y La Planificación En Sistemas De Tiempo Real, tesis doctoral, Universidad Politécnica de Valencia. 1002.
- [BMR90] S.K. Baruah, A. K. Mok y L. E. Rosier. Preemptively Scheduling Hard-Real-Time Sporadic Tasks on One Processor, Real-Time Systems Symposium, 1990, IEEE Computer Society Press.
- [BW97] Burns A. and Wellings A. (1996) *Real-time systems and programming languages*. University of York, Addison Wesley.
- [But97] Buttazzo G. (1997) *Hard real-time computing systems*. Scuola Superiore S. Anna, Kluwer Academic Publishers.
- [CA97] S. Choi y A. K. Agrawala. Scheduling Aperiodic and Sporadic Tasks in Hard Real-Time Systems. 1997.
- [Cai86] Caines P. (1986). “*Linear Stochastic Systems*”. Ed. Wiley, Canada, 1986.
- [CBLHA07] Calandrino, Baumberger, Li, Hahn y Anderson *Soft Real-time scheduling on performance asymmetric multicore platforms*, Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium Year of Publication: 2007

- [CE] A. Cervin y J. Eker. A Computational Model for Real-Time Control Tasks. Department of Automatic Control, Lund Institute of Technology, Sweden.
- [CV06] A. Cuneo y J. Vitek, A New Approach To Real-Time Checkpointing, ACM 2006.
- [CW05] Cavanagh, Wang, Design of a real-time virtual machine (RTVM), Canadian Conference on Electrical and Computer Engineering, 2005. Mayo 2005.
- [Dav] M.H.A. Davis, Linear Estimation and Stochastic Control, London Chapman and Hall, A halsted Press Book, John Wiley and Sons, New York.
- [DKT93] J. K. Dey, J. Kurose, D. Towsley. On-Line Processor Scheduling for class of IRIS (Increasing Reward with Increasing Service) Real-Time Tasks. Computer Networks and Performance Evaluation Laboratory, Department of Computer Science, University of Massachusetts, Amherst. Enero de 1993.
- [Dob53] L. J. Doob, Stochastic Processes, Willey, New York 1953.
- [GH06] Giese y Henkler en *Architecture-driven Platform independent deterministic replay for distributed hard real-time systems*. ACM 2006.
- [GL99] M. K. Gardner y J. W. S. Liu. Performance of Algorithms for Scheduling Real-Time Systems with Overrun and Overload, 1999.
- [GM03] Pedro Guevara López y José de Jesús Medel Juárez. Introducción a los Sistemas de Tiempo Real. libro publicado por la editorial Politécnico, registro de Derechos de Autor 03-2003-012912240400-01, México. 2003.
- [GMF02] P. Guevara López y J. J. Medel Juárez A. Flores Rueda. Modelo ARMA para caracterizar Tareas en Tiempo Real. 2002.
- [Gul01] P. Guevara (2001) Un Predictor Planificador para tareas esporádicas con arribo estocástico telecontrol y telemonitoreo. Tesis de Maestría en Ciencias de la Computación, CIC-IPN.
- [Gue99] P. Guevara (1999) Control de motores de corriente continua en Tiempo Real con capacidad de relativos. CIDET.
- 55
- [Hay91] Haykin S. Adaptive Filter Theory, Prentice Hall Information and System science series, 1991.

- [Hay02] Haykin S. Adaptive Filter Theory, Prentice Hall Information and System science series, 2002.
- [HGG06] Huang, Guérin y Gupta en *Supporting exceso real.time traffic with active drop queue*, IEEE/ACM Transactions on Networking (TON) 2006.
- [HS06] Hyon-Young, Sung-Gi, *RED-RT for Congestion Control and Real Time Traffic Protection*, [Computer and Information Technology, 2006. CIT '06. The Sixth IEEE International Conference on](#) ISBN: 0-7695-2687-X, Sept. IEEE 2006.
- [JBB94] K. Jeffay, D. Becker y D. Bennett. The Design, Implementation and Use of a Sporadic Tasking Model. Abril, 1994.
- [Jef92] K. Jeffay. Scheduling Sporadic Tasks with Shared Resources in Hard-Real-Time Systems. 1992.
- [JSM91] K. Jeffay, D. F. Stanat y C. U. Martel. On Non-Preemptive Scheduling of Periodic and Sporadic TAsks 1991.
- [Kan] Kannan D., An Introduction to Stochastic Processes, Elsevier North Holland, Inc.
- [KILS06] In-Koo Kang, Dong-Hyuck Im, Heung-Kyu Lee y Young-Ho Suh en *Implementation of Real-Time Watermaking Scheme for High-Quality Video* Proceeding of the 8th workshop on Multimedia and security, 2006.
- [Lee98] E. A. Lee. Modeling Concurrent Real-Time Processes Using Discrete Events, Annals of SIGMETRICS 1997.
- [Leh96] J. P. Lehoczky. Real Time Queueing Theory. IEEE, 1996.
- [Liu00] Liu J. (2000). "Real-time Systems" Ed. Prentice Hall USA.
- [Lju99] Lennart Ljung: System Identification - Theory For the User, 2nd ed, PTR Prentice Hall, Upper Saddle River, N.J., 1999.
- [LKP06] Lin, Kaldewey, Povzner y Brandt, *Diverse soft real-time Processing in an integrated system*, Proceedings of the 27th IEEE International Real-Time Systems Symposium, ISBN ~ ISSN:1052-8725 , 0-7695-2761-2 , IEEE Computer Society , 2006.
- [Lop00] López A. (2000). Sistema para adquisición y envío de señales analógicas en Tiempo Real. Tesis de maestría en ingeniería de cómputo, Centro de Investigación

en Computación, IPN.

[LL73] C. Liu y J. Layland. Scheduling Algorithms for multiprogramming in Hard Real-Time Environments. 1973.

[LMM] S. Lauzac, R. Malheim, Daniel Mossé. An Improved Rate Monotonic Admission and its Applications..

[Mit97] Isi Mitrani. Probabilistic Modelling. Ed. Cambridge University Press. 1997.

[MMA03] P. Mejía-Alvarez, R. Melheim, D. Mossé, H. Aydin. An Incremental Server for Scheduling Overloaded Real-Time Systems. 2003.

[MVPF06] Ricardo Moraes; Francisco Vasques; Paulo Portugal; Jose Alberto Fonseca, VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks, IEEE 2006.

[MVPF06] Ricardo Moraes; Francisco Vasques; Paulo Portugal; Jose Alberto Fonseca, VTP-CSMA: A Virtual Token Passing Approach for Real-Time Communication in IEEE 802.11 Wireless Networks, IEEE 2006.

[MC97-1] A. K. Mok y D. Chen. A General Model For Real-Time Tasks. Technical Report , University of Software Vol. 23 No. 19, 1997.

[MC97-2] A. K. Mok y D. Chen. A Multiframe Model for Real Time Tasks, IEEE Transactions on (Proyecto para Jóvenes Investigadores) CINVESTAV-IPN, Sección de Computación.

[MG03] J.J. Medel y P. Guevara, “*Constraints for real-time digital filters*” Automatic Control and Computer Sciences AVT No. 5 pages 63-69 ISSN 0132-4160, Septiembre 2003, (ISI), Rusia.

[NGT06] Ngolah, Wang, *Implementing the real-time processes of RTPA using Real-time Java*, Canadian Conference on Electrical and Computer Engineering, ISBN: 1-4244-0038-4, Mayo 2006.

[Nyq28] Nyquist H. “Certain Topics in Telegraph Transmission Theory”. AIEE Transactions, USA 1928.

[Oga98] K. Ogata. Sistemas de Control en Tiempo Discreto, Segunda edición, 1996. Madrid, 1998.

[PHV07] F. Pizolo, A.L. Hosking, J. Vitek. Hierarchical Real-Time Garbage

Collection, ACM 2007.

[Pue98] J. A. De la Puente. Introducción a los sistemas de Tiempo-Real. Universidad Politécnica de Madrid.

[PB98] K. M. Pasi6n y K. L. Burgu6s, Stability An6lisis of Descrete Event Systems, John Wilwy and sons, Inc., 1998.

[RD94] P. Ramanathan y D. Kang. A Generalized Guarantee Model For Servicing Sporadic Tasks with Firm Deadlines. Proceedings of the IEEE, Jan 1994.

[RIC02] Blanca Alicia Rico Jim6nez, Simulacion En Tiempo Real De Una Maquina Rotatoria Para Conversion De La Energia Electrica (C.C./C.C.), Con Una Interfaz Grafica Para Monitoreo A Distancia, tesis de Maestría del CIC, IPN, 2002.

[Roz77] Yuriy A. Rozanov, Innovation Processes. V.H. Winston & Sons, Washington, D.C. 1977.

[RS94] K. Ramamritham y J. Stankovic (1996). Scheduling Algorithms and Operating Systems support for real-time systems, (1994).

[SB96] M. Spuri y G Buttazzo, Scheduling Aperiodic Tasks in Dynamic Priority Systems. ITP An Intenational Thomson Publishing Company, 1996.

[SK96] Robert D. Strum y Donald E. Kirk, Contemporary Linear Systems Using Matlab 4.0, PWS [SSRB98]

[SBGK07] Sun, Belding-Royer, Gao y Kempf *Real-time traffic support in heterogeneous mobile networks*. ACM 2007

[SSRB98] J. A. Stankovic, Marco Spuri, Krithi Ramamritham, Giorgio C. Buttazzo. Deadline Scheduling for Real-Time Systems, EDF and Related Algorithms. Kluwer Academic Publishers, 1998.

[XQ06] Xie y Xiao Qin en Scheduling Security-Critical Real-Time Applications on Clusters, IEEE Transactions on Computers, July 2006.

[T05] Tan, *Cache design and timing an6lisis for preemptive multitasking real-time uniprocessor systems*, tesis de doctorado, Georgia Institute of technology, Atlanta, GA, 2005.

[TDSSWL95] T.S. Tia, Z. Deng, M. Storch, J. Sun, L.C. Wu, J.W.S. Liu. Probabilistic Performance Guarantee for Real-Time Tasks with Varying

Computation Times, 1995.

[Tho99] M. E. Thomadakis. The Mixed Task Scheduling Problem in Open Computing Systems. presentation, [Second Annual Workshop on Computer Science and Engineering Program](#), held at the Computer Science Department, Texas A&M University, College Station, TX 77843-3112, on April 8, 1999.

[TM07] Tan y Money *Timing análisis for preemptive multitasking real-time systems with caches*, ACM Transactions on Embedded Computing Systems (TECS), 2007.

[TWBL03] Tianzhou, Wei, Bin y Like *A Real-time Scheduling Algorithm for Embedded Systems With Various Resources Requirements*, [2006 International Workshop on Networking, Architecture, and Storages \(IWNAS'06\)](#), IEEE 2006.

[VDMM07] Van Tan, Dae-Seung, Myung-Kuyn, Myeong-Jae *Hard and soft real time based on switched Ethernet*, The 1<sup>st</sup> International Forum on Strategic Technology, ISBN: 1-4244-0427-4, Oct. 2006,

[VIC] Juan Roberto Vicuña Heredia, *Guía para el desarrollo del Administrador de Recursos para una tarjeta de entrada – salida de señales en QNX y su implementación en un sistema de Control en Tiempo Real*, Tesis de maestría, CIC, IPN.

[WSS06] Wei, Son y Satnkovic, *Real-time query processing for data stream*, IEEE 2006.

[Jia05] Jia-Li Xia *CPAHR: an concurrency control protocol against hard real-time transactions*, Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 2005.

[Zill98] Dennis G. Zill. *Ecuaciones Diferenciales con Aplicaciones Segunda Edición*. Grupo Editorial Systems. Addison-Wesley Publishing Company, 1998.

[1] *Diccionario de la Lengua Española con appendices gramaticales Larousse 38<sup>a</sup> impresión*.

[2] *Diccionario Enciclopédico Quillet en ocho tomos*, Editorial Argentina Arístides Quillet





## Anexos

### Productos obtenidos

#### Libros

**Temas Selectos de Sistemas en Tiempo Real**, con registro de autor y en proceso de publicación.

#### Revistas con arbitraje internacional

*“Modelo Dinámico para una Tarea en Tiempo Real”*. Revista Computación y Sistemas, ISSN 1405-5546, Vol. VIII No. 1, México, Septiembre de 2004.

#### Congresos nacionales e internacionales con memorias publicadas

*“Modelo Estocástico para Tiempos de Arribo en Tareas de Tiempo Real”*. Conferencia Internacional de Control, Instrumentación Virtual y Sistemas Digitales. CIC-INDI 2004, México D.F. Sep. de 2004.

*“Estimación de los Tiempos de Arribo en Tareas de Tiempo Real Periódicas con Jitter Estocástico, Utilizando la Técnica de Variable Instrumental”*. XIX Congreso Nacional de Instrumentación organizado por la SOMI, Pachuca Hidalgo, Octubre de 2004.

*“Estimadores de Parámetros en Tiempo Real (EPTR) y su Aplicación en Sistemas Estocásticos Lineales Estacionarios con Parámetros Variantes en el Tiempo”*. XIX Congreso Nacional de Instrumentación organizado por la SOMI, Pachuca Hidalgo, Octubre de 2004.

*“Matricial Estimation for Start Times with Stochastic Behavior by Periodic Real Time Tasks in a Concurrent System”*. WSEAS Transactions on Mathematics, ISSN 1109-2769, Issue 3, Volume 4, pages 242-244, Greece July 2005.

*“Matricial Estimation of Start Times With Stochastic Behavior of Periodic Real Time Tasks in a concurrent System”* 5th WSEAS International

Conference on Signal Processing, Computational Geometry & Artificial Vision (ISCGAV '05), Malta , September 2005.

*“Modelado de los Tiempos de Ejecución para una Tarea con  $n$  Instancias”*, XVIII Congreso Nacional y IV Congreso Internacional de Informática y Computación, octubre de 2005 Torreón Coahuila, México.

*“La Investigación Musical como Formación Profesional y Rescate Cultural”*. 1er. Congreso Internacional y Segundo Simposium de Orientación Educativa y Vocacional, Universidad Autónoma de Baja California. Mexicali, Baja California, Marzo de 2007.