

Caracterización de Filtros Digitales en Tiempo Real para Computadoras Digitales

Characterization of Real Time Digital Filters to Digital Computers

J. J. Medel, Pedro Guevara López, Alberto Flores Rueda

(jjmedel.guevara.aflores@pollux.cic.ipn.mx,

Laboratorio de Electrónica, Centro de Investigación en Computación – IPN

C. P. 07738 México D. F.,

Artículo recibido en enero 21, 2004; aceptado en febrero 23, 2004

Resumen

La clase de Filtros Digitales que interactúa con procesos dinámicos, y que emite respuestas de alta calidad, con restricciones de tiempo y de sincronía en forma crítica se les conocerá como FDTR (Filtros Digitales en Tiempo Real). En este artículo se desarrollarán los siguientes temas:

- Estado del arte sobre Teoría de Filtrado, teoría de Tiempo Real y teoría de filtrado en Tiempo Real,
- Conceptos teóricos que permitan describir a un FDTR a través de sus propiedades locales y globales,
- Los aspectos de implementación de un FDTR para estimar los parámetros de un sistema real a través del modelo de un motor de C.C. Presentando un análisis exhaustivo de las tareas concurrentes, de las restricciones de precedencia y de los tiempos dentro de una PC.

Palabras clave: Filtro digital, sincronía, Tiempo Real, tarea, intervalo, convergencia.

Abstract

Digital Filters that interact with dynamic process, produce high quality responses with synchrony and time constraints in critical form. This kind of filters have been known as RTDF (Real-time Digital Filters). This paper explain the next topics:

- Art of state about Filters theory, Real-time theory and Real-time Digital Filter theory.
- Development of theoretical concepts to describe local and global properties of RTDF.
- Implementation of RTDF like as parameter estimator of a D. C. motor. Including an exhaustive analysis of concurrent tasks, precedence constraints and timing constraints in a PC.

Keywords: Digital filter, synchrony, Real-time, tasks, convergence.

1. Introducción

Los Filtros Digitales en general tienen aplicaciones en los procesos industriales, en sistemas de control y de monitoreo de acuerdo a Buttazzo (1997) y Chui & Chen (1999); por ejemplo, se encuentran en: Plantas químicas y nucleares, procesos de producción, bolsas de aire de seguridad para automóviles, sistemas de inyección de combustible, análisis de voz, sistemas de control de vuelo, adquisición de datos, aplicaciones médicas, sistemas de telecomunicación, análisis de trayectorias de misiles y cohetes espaciales, entre otras muchas aplicaciones. Estos filtros interactúan con el mundo físico y no deben fallar ni en calidad de respuesta ni en tiempo de respuesta, ya que de otra manera, podrían ocasionar graves daños al proceso con que interactúan. Por este motivo a esta clase de filtros se les denominará como Filtros Digitales en Tiempo Real (FDTR).

Los Filtros Digitales en Tiempo Real pueden implementarse como sistemas dedicados (embedded systems) de acuerdo a lo escrito por Burns y Wellings (1997) a través de microcontroladores, DSP's, etc; o en computadoras con Sistemas Operativos de Tiempo Real (SOTR), como es el caso de este trabajo.

Para describir a un Filtro Digital en Tiempo Real es necesario conocer las teorías de Filtrado Digital y de Tiempo Real. En las secciones 1.1 y 1.2 se describen un conjunto de referencias que han dado sustento a ambas teorías de acuerdo a diversos autores. En la sección 2 se da un panorama del estado del arte de los FDTR. En la sección 3 se desarrolla el marco teórico de los FDTR con restricciones locales y globales para procesos dinámicos lineales estacionarios e invariantes en el tiempo, mostrando un diagrama de implementación. En la sección 4 se presenta la implementación de un FDTR para la estimación

de parámetros de un motor de C.C., desarrollando un análisis del conjunto de las tareas concurrentes, de las restricciones de precedencia y tiempos dentro de la PC; verificando que el filtro propuesto cumpla con las cualidades de un FDTR conforme a la sección 3. En la sección 5 se presentan las conclusiones de este trabajo, y en la sección 6 se incluye la bibliografía.

1.1. Filtros Digitales

El término *filtro* se usa de acuerdo a Haykin (1991) y Grewal (1993) para describir un dispositivo que en forma de *hardware* o *software* se aplica para:

- Eliminar el ruido de los datos de un sistema,
- Extraer información de acuerdo a alguna característica del sistema,
- Predecir el comportamiento del sistema a analizar,
- Reconstruir el comportamiento del sistema a analizar.

El proceso de filtrado se basa en:

- Monitorear** los estados que emite el sistema a analizar, (de acuerdo a la Fig. 1.1, el estado observable es $y(k)_i$),
- Predecir o identificar los ruidos** que pueden ser originados por causas como (ver Fig. 1.1):
 - Errores propiciados por el emisor (el sistema),
 - Errores de lectura provocados en el receptor,
 - Influencia del medio ambiente que rodea tanto al emisor como al receptor.

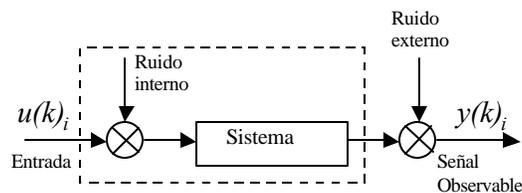


Fig. 1.1 Sistema con ruidos internos y externos.

Dentro de la teoría de filtrado existen dos áreas básicas de acuerdo a Haykin (1991), Medel (2002):

- Estimación.** Es el área que se encarga de describir la dinámica de los parámetros del sistema a través de sus señales observables (Kailath, 1980 y Medel 2002). Ver Fig. 1.2.
- Identificación.** Es el área que se encarga de describir la dinámica de los estados del sistema a través de sus señales observables (Kailath, 1980 y Medel 2002). Ver Fig. 1.2.

Para realizar un filtrado se requiere pasar por los siguientes pasos para obtener un modelo que describa de una manera aproximada el comportamiento del sistema, a través de la señal observable, sin alterar su dinámica:

- Planeación experimental,
- Selección de la estructura del modelo,
- Estimación e Identificación,
- Validación del modelo.

De acuerdo al espacio de tiempo donde se obtiene la información, la operación de filtrado puede clasificarse de la siguiente manera de acuerdo a Medel (2002):

- Filtrado.** Es la operación de extracción de información de interés del sistema en el intervalo k , a través de los datos adquiridos en el intervalo k por parte del receptor.

- II) **Reconstrucción.** Es la operación de extracción de información de interés del sistema en un intervalo $(k-n)$ para todo $n \in \mathbb{R}^+$; considerando la información recibida hasta el intervalo k .
- III) **Predicción.** Es la operación de descripción del comportamiento del sistema en un intervalo $(k+n)$, para todo $n \in \mathbb{R}^+$; considerando la información recibida hasta el intervalo k .

El error del filtrado o funcional $J(k)_i$ (error de estimación y/o error de identificación) es desarrollado bajo algún criterio para minimizar su valor medio (Haykin, 1991), de tal manera que la convergencia a un valor se dé en un tiempo finito y conocido (Young (1982)). Este criterio en general queda definido por la diferencia entre la señal filtrada y la señal original, aplicando a este resultado el segundo momento de probabilidad de acuerdo a Haykin (1991), Medel (2002). Como ejemplo de filtros digitales se tienen: los filtros de Wiener, Kalman (Haykin (1991)) y el filtro de Medel y Poznyak (Medel & Poznyak, 2001).

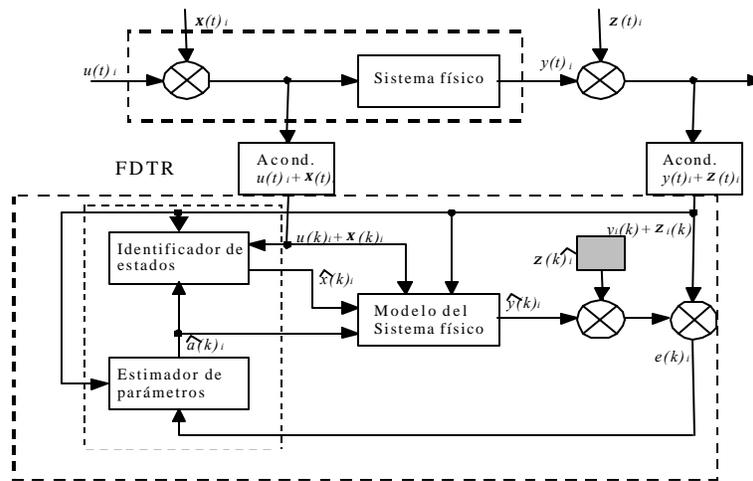


Fig. 1.2 Esquema general de un Filtro Digital funcionando como estimador de parámetros e identificador de estados.

1.2. Sistemas en Tiempo Real (STR)

En esta sección se dará un panorama de los Sistemas en Tiempo Real (STR), para describir a un FDTR.

Un *proceso dinámico* del que se quiere extraer información a través de cualquiera de sus operaciones de filtrado en forma dinámica, correcta, y acotada en tiempo (Chui & Cheng, 1999), requiere de un Filtro Digital que funcione en Tiempo Real, y al cual lo conoceremos como Filtro Digital en Tiempo Real (FDTR), ya que éste es capaz de sincronizarse con el proceso físico.

Diversos autores (entre los que destacan: Martin (1980), (Burns & Wellings, 1997), (Heitmeyer y Mandrioli 1996), Gray (1999), De la Puente (2000), Stankovic (1992), entre otros.), 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 (ver Guevara y Medel (2003)):

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 (por ejemplo Haykin (1991) entre otros), c) Cumplimiento de las restricciones de tiempo del proceso dinámico, conforme a los conceptos mostrados en Nyquist (1928).

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 aquel sistema digital que se ajusta a las restricciones de tiempo impuestas por el sistema o *proceso dinámico*, de acuerdo a los criterios que se encuentran en Nyquist (1928) y Buttazzo (1997), y que se sincroniza con las dinámicas del sistema o proceso dinámico.

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[®], Lynx[®], RT-Linux[®], 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 (Ramamritham & Stankovic, 1994).

Una *PC* 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)*.) y procesa sus peticiones mediante *tareas*, en este caso *tareas de Tiempo Real* según Liu (2000) y (Burns & Wellings, 1997).

En general cada una de las variables del *proceso dinámico* (entradas, salidas y estados) son relacionadas 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. De acuerdo a (Burns & Wellings, 1997) 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* de acuerdo a se puede clasificar conforme al cumplimiento de las restricciones de tiempo generadas por el *proceso dinámico*, en:

- 1) **Determinístico:** Es aquel *STR* en el que las restricciones de tiempo del *proceso dinámico* generan las restricciones de tiempo del sistema digital en intervalos abiertos por la derecha. En caso contrario generaría respuestas no deseadas o fuera de tiempo de acuerdo a Buttazzo (1997),
- 2) **Probabilístico:** Es aquel *STR* en el que las restricciones de tiempo del *proceso dinámico* generan las restricciones de tiempo del sistema digital en intervalos cerrados y que en un sentido de probabilidad caen en un intervalo semi-abierto de acuerdo a Ash (1970), impuesto éste último por las restricciones de tiempo del *proceso dinámico*. En un sentido de probabilidad es aceptable (Liu & Layland, 1973).

Para que un sistema computacional rígido sea de Tiempo Real, se requiere que la tarea que no cumpla su plazo, den un resultado nulo, que permita cubrir las restricciones de tiempo del *proceso dinámico* según Buttazzo (1997). Cuando las restricciones temporales no son cumplidas, este tipo de *STR* no tiene sentido aunque no afecte al proceso dinámico.

Considerando que los filtros digitales pueden ser utilizados como sistemas de control o parte de ellos. De acuerdo a Buttazzo (1997), los sistemas de control se auxilian de los sistemas computacionales que operen en Tiempo Real, ya que es esencial garantizar un comportamiento *predecible*; de hecho la estabilidad de los sistemas no solo depende del buen desarrollo del algoritmo, sino también de las restricciones de tiempo con que cuenta para interactuar a tiempo con el proceso. Un Filtro Digital que tenga contacto con un proceso dinámico, requiere además de dar un buen resultado, entregarlo a tiempo, requiriendo para ello de la teoría de Tiempo Real. A esta clase de sistemas se les llama sistemas de control en Tiempo Real.

1.3. Filtrado Digital en Tiempo Real

Chui y Chen (1999) comentan que los filtros digitales como el de Kalman se implementan en aplicaciones de Tiempo Real gracias al reciente desarrollo de computadoras de alta velocidad: Expresan al Tiempo Real como sinónimo de rápido sin hacer un estudio completo del *STR* en el que se encuentra el filtro. Plantean problemas de seguimiento de trayectorias en 2D para aeronaves a través de radares, y las únicas consideraciones que toman son la velocidad del sistema de cómputo y la capacidad del filtro para expresarse en forma recursiva.

Gustafsson (2000) describe en su libro a los filtros digitales FIR (Finite Impulse Response) como “aplicaciones para procesamiento de señales estándares en Tiempo Real”, sin dar ningún argumento que valide sus afirmaciones.

Papoulis y Bertrán (1989) hacen una cuantificación de los tiempos de desarrollo de aplicaciones (filtros digitales) para el manejo de señales desde un punto de vista práctico, donde consideran:

- a) tiempos de ejecución del algoritmo de filtrado,
- b) el procesador,
- c) tiempos de conversión A/D y D/A,

- d) diagramas de tiempos y
- e) restricciones de precedencia.

Kuo (1996) da algunos ejemplos sencillos de sistemas (aplicables a filtrado) con restricciones de tiempo impuestas por el mundo físico y coincide con lo que se comenta en Buttazzo (1997), descrito en la sección anterior.

Liu (2000) trata de establecer un lazo entre la teoría de filtrado y la de los STR, pero solo menciona al Filtro de Kalman dentro de su introducción sin explicar si lo implementó o no y cuál fue el proceso de implementación.

Baras (1999) realiza un estudio sobre procesamiento de señales en Tiempo Real, expone algunas técnicas de filtrado (incluso usando redes neuronales) y justifica que éstas son en Tiempo Real solo si son capaces de expresarse en ecuaciones recursivas de acuerdo a (Chui & Cheng, 1999).

Como se observa, diferentes autores han descrito a los filtros digitales como sistemas de Tiempo Real (FDTR), dando algunas de sus características, lo que motiva a la realización de este trabajo. Buscando describirlos a través de este conjunto de propiedades teóricas y prácticas como las mencionadas en Medel, Guevara y Flores (2003).

2. Filtro Digital en Tiempo Real (FDTR)

La contribución principal de este artículo consiste en la formalización del Filtrado Digital en Tiempo Real a través de sus características temporales en función de las definiciones, teoremas, corolarios y comentarios descritos en esta sección basándose en el estado del arte de la sección 1.3.

Aquella clase de filtros digitales que tengan un contacto con procesos dinámicos, requieran proporcionar resultados correctos acotados en el tiempo. Estos filtros serán llamados de ahora en adelante Filtros Digitales en Tiempo Real (FDTR).

Definición 1. (Filtro Digital en Tiempo Real FDTR). Un FDTR es un Filtro Digital que satisface las condiciones de un sistema de Tiempo Real de acuerdo a Buttazzo (1997), y debe cumplir (ver Medel, Guevara y Flores (2003)):

- a) Extracción y emisión de información sincronizada con el tiempo de evolución del proceso considerando al criterio de Nyquist (1928),
- b) Emisión de respuestas correctas (por ejemplo, los criterios usados en Haykin (1991) entre otros),
- c) Respuestas acotadas en tiempo a través de intervalos semi-abiertos en forma local y global de acuerdo a las restricciones temporales del proceso dinámico, conforme al criterio de Nyquist (1928) y algún criterio de convergencia,
- d) Capacidad de expresarse en forma recursiva de acuerdo a los conceptos de Baras (1999), y (Chui & Cheng, 1999). Modelos obtenidos a través de ecuaciones en diferencias finitas (Jordan (1960)).

2.1. Características temporales de los FDTR como parte de un sistema computacional

Al implementar un FDTR en una computadora monoprocador con una plataforma de Tiempo Real, los algoritmos del Filtro Digital son ejecutados por *tareas* acotadas por *plazos* semi-abiertos de tiempo; se debe de considerar:

- a) Un FDTR puede consistir en una sola tarea o en un conjunto de tareas de acuerdo al número de variables y a la construcción del sistema; esto quiere decir que no necesariamente es concurrente o paralelo: ninguna de las definiciones de los autores citados en la sección 1.2 lo expresa así.
- b) La asignación de las restricciones de tiempo a cada tarea e instancia de tarea,
- c) El uso de operaciones predecibles que permitan garantizar las restricciones de tiempo definidas en el punto a) de esta sección.

- d) De acuerdo a la Definición 1, la respuesta temporal de un FDTR no es ni rápida ni instantánea. Es decir, un filtro digital no es de tiempo real solo por implantarlo en un DSP, sistema operativo de tiempo real como QNX o un paquete visual como RT-LabView o Matlab/Simulink/RTW, entre otros, si no se hace un análisis temporal formal.

Se considera que el filtro evoluciona dentro de intervalos $k+i$, $i \in \mathbb{I}[0,n]$; acotados por el periodo de muestreo (T), del proceso dinámico. La caracterización temporal del comportamiento de un FDTR se describe a continuación:

Definición 2. (Características temporales de las tareas de un FDTR). Las *tareas* de un FDTR tienen un conjunto de características temporales dentro de los intervalos de tiempo k , con $k=0, \dots, n$ (ver Medel, Guevara y Flores (2003)):

- Tiempo de arribo o llegada* ($l(k)_i$) (Buttazzo (1997)): Es el tiempo absoluto con respecto al inicio del proceso en que una tarea pasa a formar parte de la *cola de listos*. Es decir, es un tiempo absoluto en que la tarea queda formada hasta ser atendida. Cumpliendo: $l(k)_i > 0 \quad \forall k \in \mathbb{I} \mathbb{N}^+$.
- Tiempo de ejecución* ($C(k)_i$) en cada intervalo: Es el tiempo que el procesador se encarga de ejecutar una tarea (o conjunto de tareas sin interrupción), donde $C(k)_i \in \mathbb{R}$.
- Plazo relativo mínimo* ($D(k)_i_{min}$) dentro de cada intervalo: Es el tiempo mínimo en el que se puede obtener una respuesta por parte del FDTR, sin alterar negativamente la dinámica o comportamiento del sistema físico de acuerdo a Kuo (1996) y a Buttazzo (1997). El plazo relativo mínimo cumple $D(k)_i_{min} \in \mathbb{R}$.
- Plazo relativo máximo* ($D(k)_i_{max}$): Es el tiempo máximo en el que se puede obtener una respuesta por parte del FDTR, sin alterar negativamente la dinámica del proceso de acuerdo a Kuo (1996) y Buttazzo (1997). El plazo relativo máximo cumple $D(k)_i_{min} \leq D(k)_i_{max}$.
- Tiempo de inicio o tiempo de liberación* ($s(k)_i$). Es el tiempo en que una tarea inicia su ejecución. Y cumple: $l(k)_i \leq s(k)_i < l(k)_i + D(k)_i_{max}$.
- Tiempo de finalizado* ($f(k)_i$) en cada intervalo k : Es el tiempo en que una tarea termina su ejecución dentro del intervalo k . La condición principal es $f(k)_i \in [ld(k)_i_{min}, LD(k)_i_{max}]$, con $ld(k)_i_{min} := l(k)_i + D(k)_i_{min}$, y $LD(k)_i_{max} := l(k)_i + D(k)_i_{max}$.
- Tiempo de retraso* ($L(k)_i$) en cada intervalo k : Es el tiempo definido por $L(k)_i := \lceil LD(k)_i_{max} - f(k)_i \rceil$. Si $LD(k)_i_{max} > f(k)_i$ significa que la tarea fue terminada dentro de su plazo absoluto máximo, en caso contrario, la tarea terminó después del intervalo k , y cuenta con un retraso.
- Tiempo prematuro* ($P(k)_i$) en cada intervalo k : Es descrito por la siguiente expresión $P(k)_i := \lceil ld(k)_i_{min} - f(k)_i \rceil$. Si $f(k)_i \geq ld(k)_i_{min}$.
- Plazo de convergencia* (d_i): Se define como el tiempo máximo en que el FDTR debe converger a un ϵ . Está determinada por la dinámica del mundo real.

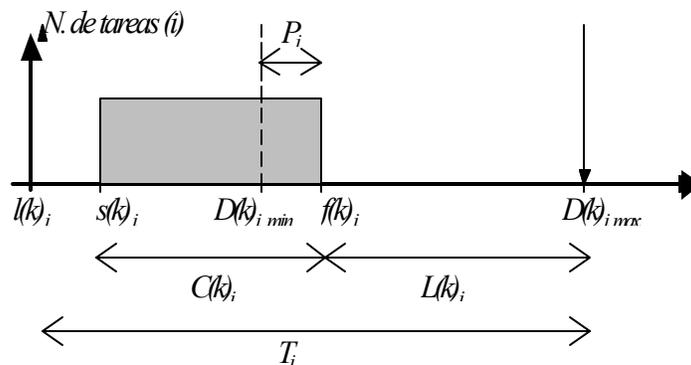


Fig. 2.1. Tarea de Tiempo Real en un FDTR.

Comentario 1: Se llama procesador a un recurso activo de acuerdo a Liu (2000).

Comentario 2: Los tiempos absolutos pueden referenciarse a las 0:00 del 1/1/70 del meridiano de Greenwich como lo hacen algunos relojes de Tiempo Real.

Comentario 3: El tiempo de ejecución ($C(k)_i$) puede estar formado por la suma de los tiempos ejecución de una o varias tareas, dependiendo de la complejidad del FDTR y su número de variables.

Comentario 4: Los plazos $D(k)_{i_min}$, $D(k)_{i_max}$ son relativos a $l(k)_i$. Si la respuesta del FDTR es emitida antes de $D(k)_{i_min}$ existe un tiempo prematuro $P(k)_i$ y ésta debe de esperar para ser liberada. Si la respuesta del es emitida después de $D(k)_{i_max}$, entonces no sirve y se presenta un retraso $L(k)_i$.

Comentario 5: Al sumar, el tiempo de llegada $l(k)_i$ a los plazos relativos $D(k)_{i_min}$, $D(k)_{i_max}$, permitirá obtener plazos absolutos ($ld(k)_{i_min}$, $LD(k)_{i_max}$) (Liu (2000)).

Comentario 6: El tiempo de inicio $s(k)_i$ es una variable aleatoria, con función de distribución conocida. Se encuentra en función de: a) La disponibilidad de recursos, b) Los tiempos de latencia, c) Del cambio de contexto, d) Del tamaño de la cola de listos, entre otros. No se conoce la relación entre el conjunto de variables que forman a $s(k)_i$ y no permiten describirlo en forma predecible (ver por ejemplo Gnedenko (1962)).

Comentario 7: La evolución entre intervalos consecutivos está acotada al periodo de muestreo $T(k)_i$. Su magnitud se obtiene a partir del criterio de Nyquist (1928), con $T(0)_i=0$. Las condiciones de T_i son:

- 1) $t(k)_i - t(k+1)_i - t(k)_i = T(k)_i$ y
- 2) $l(k)_i, LD(k)_{i_max} = T(k)_i + g$

De acuerdo al concepto de medida expresado en Ash (1970), donde g , representa un tiempo de holgura infinitamente pequeño (ver: Buttazzo (1997)). Si $LD(k)_{i_max} = 0$ significa que $LD(k)_{i_max} = 0$ y significa que tendremos una tarea nula. El mínimo tiempo en el cual puede ser entregada una tarea es descrito por $LD(k)_{i_min} = T(k)_i - L(k)_i$. Conforme al concepto expresado en la definición 2, inciso f.

En la *Definición 1* se pide la obtención de respuestas correctas, para esto se requiere hacer dos tipos de análisis:

- a) Global: La convergencia a un ϵ a través del funcional de error $\{J(k)\}_i$, conforme a Haykin (1991), involucrando un intervalo de convergencia m , un tiempo de convergencia t_{c_i} y un plazo de convergencia d_i .
- b) Local: Implica la estabilidad del proceso a través de sus parámetros $\{a(k)\}_i$ de acuerdo a Caines (1986) y el de los tiempos de finalizado $f(k)_i$ del FDTR dentro de los plazos absolutos correspondientes $[ld(k)_{i_min}, LD(k)_{i_max}]$, de acuerdo a Buttazzo (1997).

2.2. Análisis del comportamiento global del FDTR

Un FDTR proporciona una respuesta correcta global de acuerdo a un funcional de convergencia $\{J(m)\}$ que tienda a un valor de $\epsilon > 0$, con $m \in \mathcal{Q}$, y $m \in \mathbb{N}^+$. El número m es el contador de intervalos en que la respuesta del FDTR converge.

Además del grado de convergencia del FDTR, es necesario acotar su tiempo de respuesta, para ello se da la siguiente definición:

Definición 3 (Tiempo de convergencia t_{i_c}). Es el tiempo en que el FDTR converge y está expresado por:

$$t_{i_c} := f_i(k=m), \quad (1)$$

donde m es número del intervalo de convergencia del FDTR. La condición que debe cumplir es:

$$d_{i_c_min} \leq t_{i_c} < d_i. \quad (2)$$

Para garantizar una respuesta a tiempo y sincronizada con el sistema físico. El tiempo mínimo de convergencia del filtro es descrito:

$$d_{i_c_min} := \inf_{k=m} [ld(k)_{i_min}, d_i], \quad (3)$$

de acuerdo al concepto de medida expresado en Ash (1970).

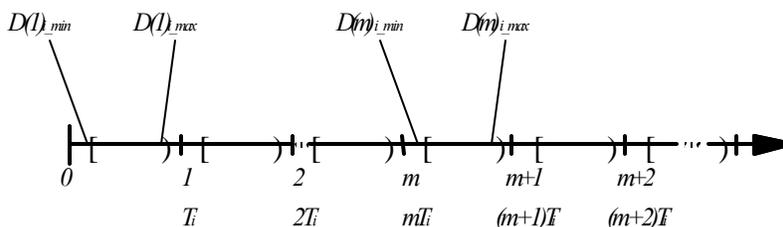
Teorema 1 (Plazo de convergencia d). *El funcional del error de convergencia en el sentido de probabilidad definido en Haykin (1991 y en Caines (1986) para un Filtro Digital, tendrá una cota ínfima $\epsilon > 0$ a la que debe converger el estimador dentro del plazo d_i de acuerdo a Buttazzo (1997).*

Demostración.

Considérese:

- a) *La cota ínfima ϵ es menor que cero. El funcional del error descrito por el segundo momento de probabilidad definido en Haykin (1991) tendría un valor negativo, lo cual sería una contradicción de acuerdo a Ash (1970) ya que comenta que dicha cota es mayor o igual a cero.*
- b) *El tiempo de convergencia del filtro con restricciones de tiempo críticas a la cota ínfima ϵ es mayor al plazo d , entonces la respuesta de este no es correcta en tiempo de acuerdo a Buttazzo (1997) y (Liu & Layland, 1973), ya que un tiempo de finalizado debe estar acotado por un plazo máximo para que el comportamiento sea de Tiempo Real.*

En la figura 2.2 se exponen las restricciones temporales locales y globales.



- El análisis local se realiza por intervalos de plazos relativos por cada instante k .
- El análisis global se realiza para un instante m con múltiplos de T_i .

Fig. 2.2. Restricciones de tiempo locales y globales del EPTR.

2.3. Análisis del comportamiento local del FDTR para sistemas lineales e invariantes en el tiempo (SLIT)

De acuerdo a Kuo (1996), Rudin (1988) y Caines (1986), un FDTR dará una respuesta local estable si al estimar el conjunto de parámetros del SLIT (Sistema Lineal Invariante en el Tiempo) a filtrar, caen todos ellos dentro del círculo unitario para todo intervalo k de acuerdo a la Definición 1. Esto es:

$$\{|a_e(k)\}_i < 1, k_\tau = 1, \dots, n_\tau \quad (4)$$

El conjunto de parámetros estimados $\{a_e(k)\}_i$ representa los valores propios del sistema modelado (Kailath (1980)), tal que para sistemas discretos estables, esos valores deben estar dentro de un círculo de radio unitario (Kuo (1996), Kailath (1980), y Caines (1986)). Fuera de ese círculo la respuesta es no estable, en consecuencia, el filtro no podrá describir alguna característica del sistema a filtrar (Haykin (1991)).

El cumplimiento de las restricciones de tiempo por parte del FDTR está dirigido de acuerdo a las condiciones explicadas en la Definición 2, y se exponen en los siguientes teoremas y corolarios (ver Medel, Guevara y Flores (2003)):

Teorema 2 (Plazo relativo máximo $D(k)_i_{max}$). *Un FDTR debe cumplir con la siguiente condición:*

$$2f_{max}(D(k)_i_{max} - D(k)_i_{min}) < 1, \quad (5)$$

respecto a la frecuencia máxima de respuesta del sistema físico $f_{i_{max}}$.

Demostración.

Del criterio de Nyquist (1928), tenemos la siguiente relación:

$$\frac{1}{2f(k)_{i_max}} \geq \frac{1}{f(k)_{i_muestreo}}, \quad (6)$$

y de acuerdo a la Definición 2 en su inciso i:

$$\frac{1}{2f(k)_{i_max}} \geq T(k)_i. \quad (7)$$

De la Definición 2 inciso i, punto 2, observamos:

$$T(k)_i > D(k)_{i_max} - D(k)_{i_min}. \quad (8)$$

Por transitividad, de las desigualdades (7) y (8), obtenemos la desigualdad (6). δ

2.4. Tiempos de ejecución del FDTR

El algoritmo de un FDTR en éste trabajo está compuesto por un conjunto de sub-tareas concurrentes planificadas a través de restricciones de precedencia que garanticen la sincronía entre el FDTR y el mundo físico así como entre ellas mismas dentro de los intervalos correspondientes. Las tareas concurrentes tienen tiempos de ejecución como (ver Fig. 2.3):

$C(k)_x$: Tiempo de ejecución del algoritmo de la ecuación de estado propuesta.

$C(k)_y$: Tiempo de ejecución del algoritmo de la ecuación de la señal observable.

$C(k)_a$: Tiempo de ejecución del algoritmo del estimador.

$C(k)_j$: Tiempo de ejecución del algoritmo del error de convergencia.

$C(k)_{au}$: Tiempo de ejecución del acondicionamiento de la entrada $u(t)$ del sistema físico.

$C(k)_{ay}$: Tiempo de ejecución del acondicionamiento de la salida $y(t)$ del sistema físico.

$C(k)_{aye}$: Tiempo de ejecución del acondicionamiento de la salida estimada $\hat{y}(t)$ del sistema físico.

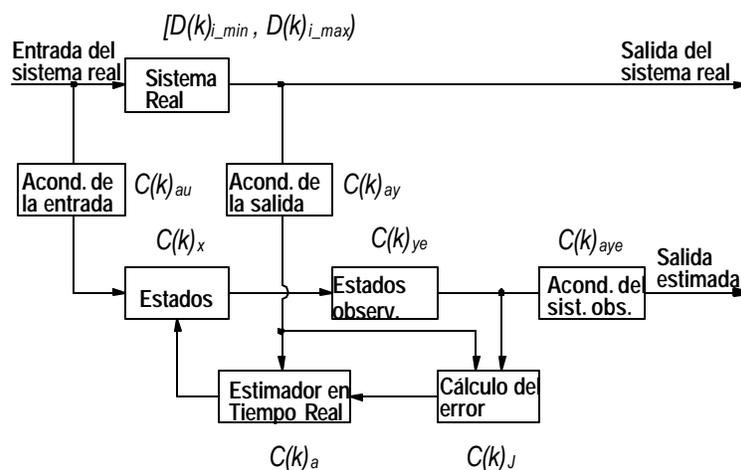


Fig. 2.3. Diagrama de implementación a bloques de un FDTR. utilizado como estimador.

Como se considera que es un *sistema concurrente*, todos estos tiempos de ejecución deben ser sumados para obtener un tiempo de ejecución $C(k)_i$. Tal que:

$$C(k)_i = C(k)_x + C(k)_y + C(k)_a + C(k)_J + C(k)_{au} + C(k)_{ay} + C(k)_{aye} \quad (9)$$

Teorema 3 (Tiempo de ejecución y plazo). El tiempo de ejecución $C(k)_i$ de un FDTR debe ser menor al plazo relativo máximo $D(k)_{i_max}$. es decir (ver Medel, Guevara y Flores (2003)):

$$C(k)_i < D(k)_{i_max}. \quad (10)$$

Demostración.

a) Considere que el caso de la Figura 2.4.

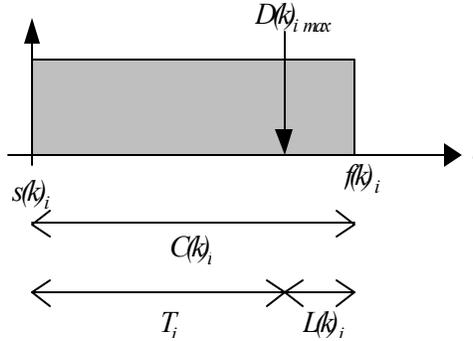


Fig. 2.4. Tarea de Tiempo Real con restricciones de tiempo extremas.

Se tienen las restricciones de tiempo más extremas:

$$s(k)_i = l(k)_i, \quad (11)$$

$$D(k)_{i_max} = 0, \text{ (ya que es relativo a } l(k)_i), \quad (12)$$

$$C(k)_i > D(k)_{i_max}, \quad (13)$$

$$L(k)_i > 0, \text{ y} \quad (14)$$

$$D(k)_{i_max} = T(k)_i. \quad (15)$$

entonces:

$$C(k)_i = D(k)_{i_max} + L(k)_i. \quad (16)$$

Si se considera que el tiempo de retraso $L(k)_i$ es un submúltiplo del plazo, se tiene:

$$L(k)_i = \frac{D(k)_{i_max}}{r} \quad (17)$$

Donde $r \in \mathbf{N}^+$, $r > 1$.

El tiempo de ejecución es expresado:

$$C(k)_i = (1 + \hat{a}) D(k)_{i_max}, \quad (18)$$

Con $\hat{a} := 1/r. \quad (19)$

Obteniendo el mínimo con respecto al plazo relativo. Se tiene que $r=-1$, lo que significa que no se cumplió la hipótesis de este inciso de la demostración.

b) De acuerdo a la Definición 2, se tiene:

$$C(k)_i = f(k)_i - s(k)_i, \quad (20)$$

$$y f(k)_i \in [l(k)_i + D(k)_{i_min}, l(k)_i + D(k)_{i_max}). \quad (21)$$

Así como también, al considerar el inciso a) de esta prueba:

$$f(k)_i - s(k)_i < D(k)_{i_max}. \quad (22)$$

Y el peor caso cuando $f(k)_i \in l(k)_i + D(k)_{i_max}$, al sustituir en la expresión anterior, tenemos:

$$l(k)_i - s(k)_i < 0 \quad (23)$$

Como la afirmación es cierta, el teorema se cumple. \square

Corolario 1: Por el uso de recursos del procesador se tiene que de acuerdo a (Liu & Layland, 1973), un FDTR Crítico (considerado este como un sistema de Tiempo Real), es aquel donde todas las tareas se terminan antes de su plazo $D(k)_{i_max}$.

Por contradicción, probemos este corolario. Supóngase que:

$$Si tal que $C(k)_i > D(k)_{i_max}$, \quad (24)$$

resultado contradictorio al expuesto por (Liu & Layland, 1973), para el uso del procesador:

$$\frac{C(k)_i}{D(k)_{i_max}} \leq 1, \quad (25)$$

y la hipótesis de partida de la prueba de este corolario es una contradicción, ya que $C(k)_i < D(k)_{i_max}$ de acuerdo al Teorema 3. \square

Teorema 4 (Sincronización). El conjunto de respuestas de un FDTR para un sistema dado, debe estar sincronizada con el mundo físico a fin de evitar demoras o adelantos acumulativos.

Demostración.

- El conjunto de respuestas $\{f(k)_i\}$ debe encontrarse dentro del conjunto de intervalos $\{[l_d(h)_{i_min}, LD(h)_{i_max}]\}$ para todo i , $h, k \in \mathbb{N}^+$, tal que cada $f(k)_i$, cumplirá que: 1) Es mayor al plazo absoluto mínimo $\{l_d(h)_{i_min}\}$ y, 2) Es menor o igual el plazo relativo máximo $LD(h)_{i_max}$. Esto es cierto si y solo si $h \leq k$, y esto significa que el DFTR esta sincronizado con el mundo físico.
- En el caso de que $h > k$, entonces las respuestas y los plazos del FDTR no tendrían una relación biyectiva de acuerdo a la Definición 1, obteniendo retrasos acumulativos que ocasionarían tiempos de ejecución no sincronizados con sus plazos.
- Si $h < k$ entonces las respuestas y los plazos del FDTR no tendrían una relación biyectiva de acuerdo a la Definición 1, obteniendo retrasos acumulativos que ocasionarían tiempos de ejecución no sincronizados con sus plazos \square .

En forma ilustrativa se observa la Fig. 2.5:

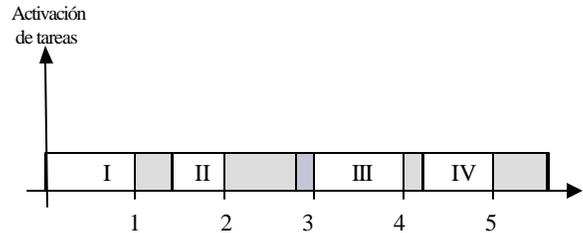


Fig. 2.5. Tarea de Tiempo Real no sincronizada con su intervalo (las franjas indican los retrasos de las tareas). En este caso la instancia IV debe terminar antes del plazo 4 y termina en el 6. La correspondencia debe ser tarea IV – plazo 4 y no se cumple, por lo tanto el sistema está no sincronizado y hay un retraso acumulativo.

3. Implementación de un Filtro Digital en Tiempo Real

En el proceso de implementación de un FDTR se presenta un algoritmo de estimación de parámetros para un motor de C.C. utilizando el método de mínimos cuadrados sobre un conjunto de equipo y materiales de hardware y software. El experimento se realizó en el Laboratorio de Electrónica del Centro de Investigación en Computación.

La implementación del FDTR es independiente de la plataforma de Tiempo Real que se utilice ya que las restricciones temporales del FDTR las dicta el mundo real y la plataforma a usarse debe cumplirlas (ver las secciones 1.1, 1.2 y 1.3, así como las definiciones 1 y 2). En este trabajo se eligió QNX4.24 (plataforma disponible en la red en forma gratuita para estudiantes e investigadores)

Una vez implementado el FDTR propuesto se realizaron pruebas de estabilidad, convergencia y cumplimiento de las restricciones de tiempo y sincronía necesarias para comprobar sus características de Tiempo Real.

3.1. Herramientas de implementación

Para la implementación del FDTR se utilizaron los siguientes equipos y herramientas:

a) Hardware

- Motor de C.C. de 20 V, 1 A, 1800 rpm, dos polos, campo permanente con tacómetro analógico integrado.
- Acoplamiento de potencia A/D, D/A de 20 V, 5 A.
- PC Pentium III 400 MHz, 64 MB RAM,
- Tarjeta de adquisición de datos Advantech PCL 818L,

b) Software de implementación

- Sistema Operativo de Tiempo Real QNX[®] 4.24,
- Ambiente gráfico MicroPhoton[®],
- Watcom[®] C.

3.2. Características del Filtro Digital

El modelo y filtro propuestos consideran un sistema con las siguientes características (Medel (2002), (Vicuña & Guevara, 2000)):

- El sistema (modelo del motor) es del tipo UEUS (una entrada una salida, el subíndice $i=1$), lineal, estacionario e invariante en el tiempo, caracterizado por un modelo Autoregresivo de acuerdo a Haykin (1991) y Caines (1986),
- Entrada escalón de 20 volts al motor.
- Algoritmo Mínimos Cuadrados (ver: Haykin (1991)),
- Los ruidos $v(k)$ y $w(k)$ están correlacionados con la señal observable $y(k)$ pero no entre ellos.

Las ecuaciones del proceso dinámico y del filtro utilizadas para el experimento se expresan a continuación:

$$x(k+1) = ax(k) + v(k) \quad (26)$$

$$y(k) = x(k) + w(k) \quad (27)$$

donde:

$x(k+1)$ es el estado del sistema,

$y(k)$ es la señal observable (velocidad angular),

$v(k)$ y $w(k)$ son las perturbaciones internas y externas respectivamente.

a es el parámetro a estimar.

El estimador está expresado por:

$$a(k) := P(k)B(k)^{-1}, \quad (28)$$

definido por:

$$P(k) := \hat{\sigma}^2(j)y(j-1), \quad (29)$$

$$B(k) := \hat{\sigma}^2(j-1)y(j-1), \quad (30)$$

en forma recursiva:

$$P(k) = P(k-1) + y(k)y(k-1), \quad (31)$$

$$B(k) = (B^{-1}(k-1) + y(k-1)y(k-1))^{-1}. \quad (32)$$

Esta forma de representación del estimador de parámetros cumple con la *Definición 1*.

El error de estimación es definido por Haykin (1991):

$$\mathbf{D}(k) := a(k) - a, \quad (33)$$

así como el funcional del error de acuerdo a Haykin (1991) y Caines (1986) :

$$J(k) = E(\mathbf{D}(k)\mathbf{D}^T(k)) \quad (34)$$

Para el modelo del motor se consideraron como resultados de la validación del algoritmo de estimación los siguientes datos utilizados en Medel (2002):

$$a = 0.3, \quad \sigma^2(v(k)) = 0.95, \quad \sigma^2(w(k)) = 0.99.$$

3.3. Implementación experimental del FDTR

La decisión de implementar este FDTR a través de sub-tareas concurrentes se basa en que idealmente cada actividad del sistema real se realiza en paralelo a ellas en (Burns & Wellings, 1997), se comenta que en un sistema físico, las actividades se realizan es paralelo) y en la *Definición 1* se describe como unas de las características de un FDTR es que este sincronizado con el mundo físico (en este caso con el motor de c.c.). Para ello se requiere que las tareas sean planificables y que la suma de sus tiempos de ejecución (y el tiempo de finalizado $f(k)$) cumpla: $C(k)_i < D(k)_i_{max}$.

QNX[®] soporta POSIX de Tiempo Real (1003.1b, 1003.1c) para la creación de tareas se utilizó la tarea "O". En la Fig. 3.1 se presenta la secuencia de creación de las tareas concurrentes utilizadas por el estimador de parámetros (FDTR); una vez que todas las tareas son creadas (nueve tareas), la tarea "O" muere y su identificador es borrado del sistema.

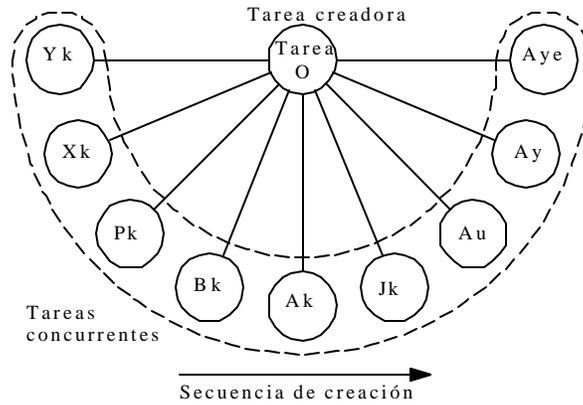


Fig. 3.1. Creación de las tareas utilizadas para el FDRT.

Para la planificación de las tareas se usaron mensajes bloqueantes; ésta técnica evitó el uso de algoritmos como el RM (ver: Liu & Layland, 1973) o el EDF (ver: Buttazzo (1997)) y además impuso a las sub-tareas las restricciones de precedencia necesarias donde cada flecha representa la comunicación de dos sub-tareas mediante la forma *send()* – *receive()* – *reply()* (ver :Krtén (1998)). Este resultado se muestra en la Figura 3.2.

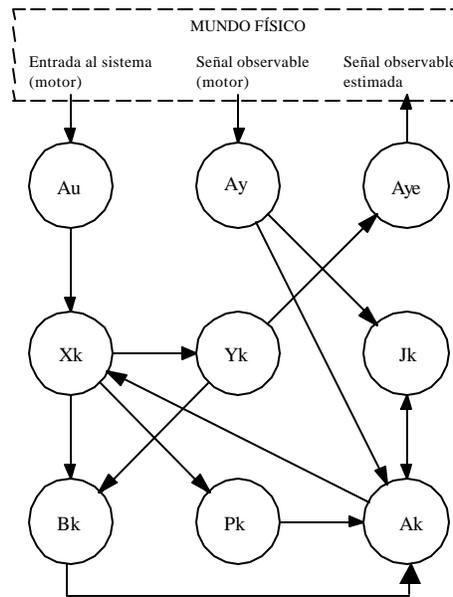


Fig. 3.2. Grafo de comunicación de tareas.

Como resultado de la Figura 3.2. se presenta la siguiente matriz de precedencia de tareas:

	Yx	Xk	Pk	Bk	Ak	Jk	Au	Ay	Aye
Yk		∈	∈	→	∪	∪	∈	∈	→
Xk	→		→	→	∪	∪	∈	∈	∪
Pk	∪	∪		∪	→	∪	∈	∈	∪
Ak	∪	→	∪	∪		→	∈	∈	∪
Bk	∪	∪	∪		→	∪	∈	∈	∪
Jk	∪	∪	∪	∪	∪		∈	∈	∪
Au	∪	→	∪	∪	∪	∪		∈	∪
Ay	∪	∪	∪	∪	→	→	∈		∪
Aye	∈	∈	∈	∈	∈	∈	∈	∈	

Donde de acuerdo a la notación expuesta en Buttazzo (1997) y Liu (2000) se tiene:

- Ⓜ La tarea A_i es inmediatamente predecesora de A_1 .
- Ⓔ La tarea A_i es predecesora no inmediata de A_1 .
- Ⓘ La tarea A_i no es predecesora de la tarea A_1 .

Con la restricción de l^i .

La implementación experimental del FDTR se describe en la Figura 2.3. Para esto se consideraron las siguientes restricciones:

- Los tiempos de ejecución ponderados ($C_p \cdot k$) del conjunto de tareas se obtuvieron del primer momento de probabilidad del conjunto de tiempos de ejecución $\{C(k)\}$,
- El plazo máximo $D(k)_{max}$, del sistema es igual al periodo $T(k)$.
- El tiempo de inicio ponderado ($s_p \cdot l(k) \cdot k$) se obtuvo a través del primer momento de probabilidad del conjunto de tiempos de inicio $\{s(k)\}$ tal que $s_p = l(k) + 0.0015$ ms .
- El periodo de muestreo $T(k)$ para el motor de C.C a 1500 rpm es de 20 ms.
- El plazo mínimo $D(k)_{min} = 2.5$ ms. Experimentalmente se comprobó que una respuesta en un tiempo menor afectaba al motor en forma no deseada
- El plazo de convergencia del motor (en base a su curva de arranque) es: $d = 3$ s.

Para tal fin se midieron los tiempos de ejecución de cada sub-tarea $C(k)$:

Xk: Ecuación de estados del sistema.	$C_{Xk} = 0.237$ ms,
Yk: Ecuación de estados observables.	$C_{Yk} = 0.289$ ms,
Bk: Varianza de la señal observable.	$C_{Bk} = 0.258$ ms,
Pk: Ecuación de Ricatti.	$C_{Pk} = 0.249$ ms,
Ak: Estimador de parámetros.	$C_{Ak} = 3.252$ ms,
Jk: Funcional de error.	$C_{Jk} = 0.245$ ms,
Au: Acondicionamiento: entrada.	$C_{Au} = 0.310$ ms,
Ay: Acondicionamiento: señal observable.	$C_{Ay} = 0.302$ ms,
Aye: Acondicionamiento: señal observable estimada.	$C_{Aye} = 0.314$ ms,
O: Tarea creadora.	$C_O = 0.261$ ms.

El valor de $C(k)$ se obtuvo de acuerdo a (9), esto es:

$$C(k) = C_{Xk} + C_{Yk} + C_{Bk} + C_{Pk} + C_{Ak} + C_{Jk} + C_{Au} + C_{Ay} + C_{Aye},$$

De acuerdo a la *Definición 2*, inciso *f*, el tiempo de finalizado total $f_T(k)$, es igual a la suma del tiempo de ejecución $C(k)$, más la suma de sus cambios de contexto descritos por $(s(k)-l(k))$, para cada intervalo k . En este trabajo, se consideró al promedio de la suma de los cambios de contexto, para todas las evoluciones k :

$$f_T(k) = C(k) + 9 (s(k)-l(k)), \quad (35)$$

La precedencia de ejecución de las tareas del FDTR se muestra en la *Figura 3.3*.

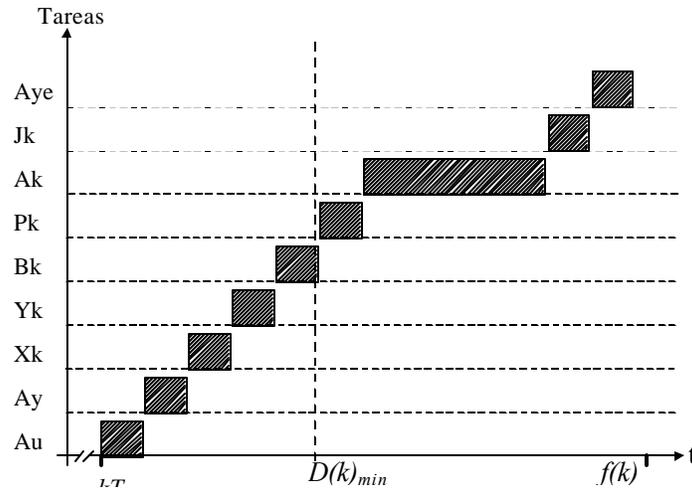


Fig. 3.3. Diagrama de tiempos de procesos del FDTR.

Si se realiza un diagrama equivalente suponiendo que el FDTR está compuesto por una sola sub-tarea se obtiene la Figura 3.4., donde:

$T(k) = 20$ ms, $l(k) = T(k-1)k$ ms, $s(k) = l(k) + 0.0135$ ms, $C(k) = 5.7305$ ms, $D(k)_{min} = 2.5$ ms, $D(k)_{max} = 20$ ms, $f(k) = 5.7305$ ms, $L(k) = 14.269$ ms, $P(k) = 3.2305$ ms.

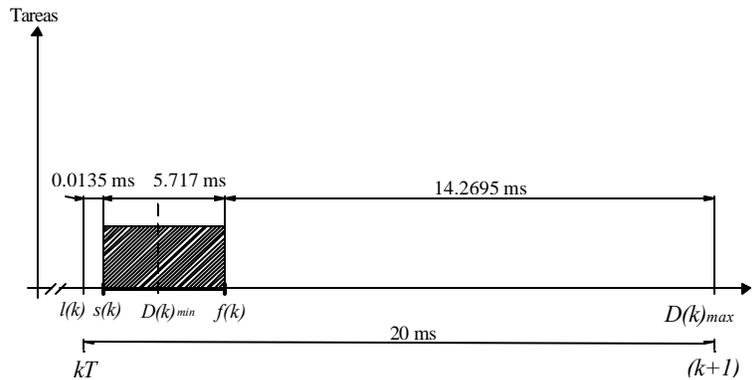


Fig. 3.4. Diagrama de tiempos equivalente a las tareas concurrentes del FDTR.

Del experimento se obtuvieron las Figura 3.5 y Fig 3.6 con los siguientes resultados:

$$m = 113 \text{ intervalos, } t_c = 2.24 \text{ s, } d = 3 \text{ s,}$$

Con $t_c < d$ se muestra que el plazo de convergencia se cumplió.

Los ruidos se generaron a través de las funciones *random* de Watcom® C en QNX® con un rango de variación de ± 0.008 con relación a los ruidos reales. El error de estimación calculado fue de 0.32 unidades por arriba del parámetro del modelo, siendo el valor estimado de 0.62 de acuerdo al experimento mostrado por Medel (2002) (ver la Fig. 3.5).

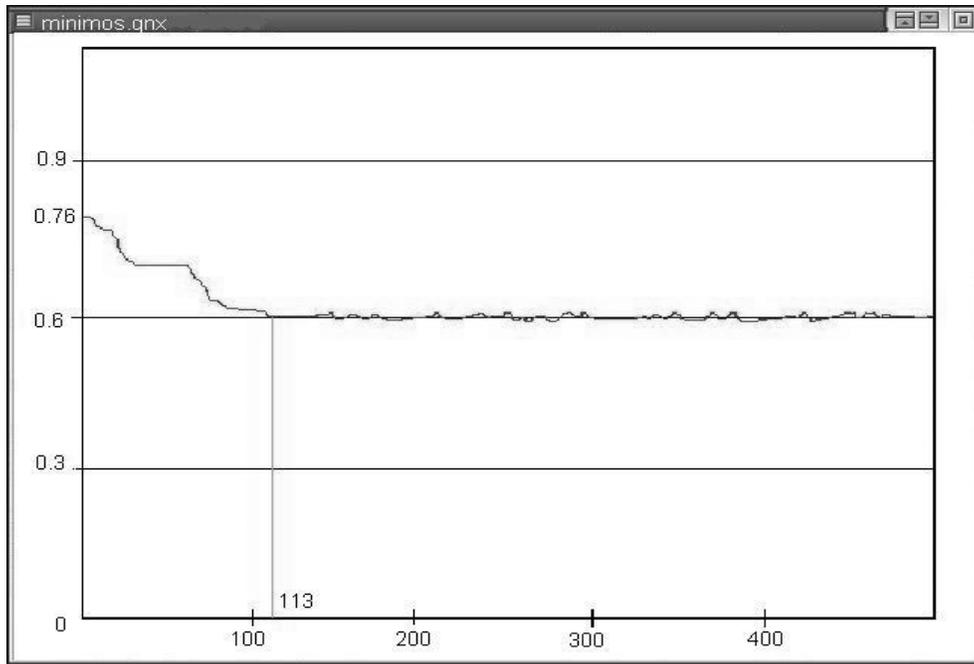


Fig. 3.5. Gráfica obtenida de la estimación del parámetro a a través del FDTR.

El resultado del estimador de la ecuación 28, fue sustituido en el modelo descrito por las ecuaciones 31 y 32, y se obtuvo un gráfico donde se observan la evolución del proceso dinámico a través de sus señales observables real y estimada. Ambas señales en línea, ejecutándose en el mismo intervalo k . (Ver Figura 3.6.).

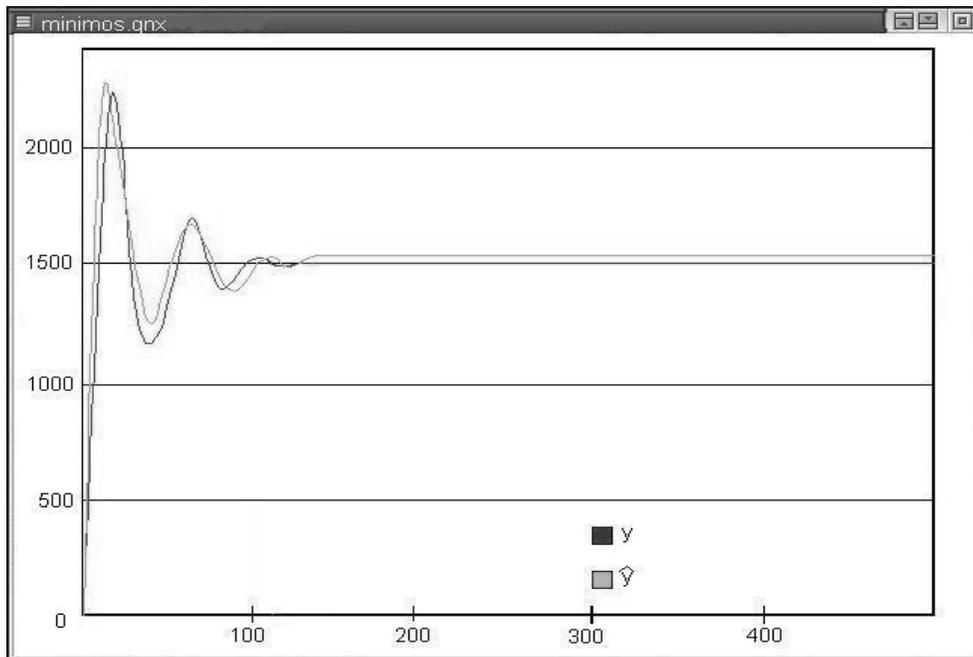


Fig. 3.6. Señales observable real y observable estimada.

De la implementación de este Filtro Digital se puede decir que cumple con las definiciones y restricciones de un FDTR: Interactúa con un proceso físico y dinámico (motor), ofrece respuestas correctas tanto en estabilidad de parámetros estimados como en convergencia, cumple los plazos establecidos para el proceso dinámico y está expresado en forma recursiva. Además, para todo caso sus tiempos de finalizado $f(k)$ (probabilísticamente) siempre estuvieron dentro de los intervalos $[ld(k)_{i_min}, LD(k)_{i_max}]$ para toda k , y se concluye que se logró implementar un FDTR.

4. Conclusiones

Los Filtros Digitales tienen una amplia gama de aplicaciones en diversos sectores; muchas veces tienen interacción directa con el mundo real a través de procesos dinámicos siendo necesario que se sincronicen con estos en calidad de respuesta en forma local y global.

Muchos autores definen Filtro Digital en Tiempo Real, pero dejan algunos huecos entre las teorías: Filtrado y Tiempo Real. En este trabajo se da un conjunto de condiciones a cumplir por parte de los FDTR, se exponen algunas definiciones, teoremas corolarios y comentarios; pero en general se expone que los FDTR deben proporcionar respuestas correctas, cumplir todas las restricciones de tiempo (plazos) y sincronía para no modificar indeseadamente las condiciones del proceso físico con el que interactúan. No siempre es necesario contar con la computadora más rápida, sino con aquella que cumpla todas las restricciones impuestas.

En la sección 3 de este artículo se presentó la implementación de un FDTR y se demostró la posibilidad de que un estimador como el de mínimos cuadrados puede cumplir con características de Tiempo Real si se consideran las condiciones necesarias y se cuenta con el equipo adecuado para lograr tal propósito.

Referencias

- [1] Ash R. "Real Analysis and Probability", Ed. Academic Press, USA, 1970.
- [2] Baras J. "Symbolic and numeric Real-time signal processing", Technical Report University of Maryland, USA 1999, pp 226.
- [3] Burns A., Wellings A. "Real-time systems and programming languages". University of York, Addison Wesley, 1997.
- [4] Buttazzo G. "Hard real-time computing systems". Scuola Superiore S. Anna, Kluwer Academic Publishers, 1997.
- [5] Caines P. "Linear Stochastic Systems". Ed. Wiley, Canada, 1986.
- [6] Chui C., Chen G. "Kalman Filtering with Real-time Applications". Ed. Springer, USA 1999.
- [7] Cheng D. "Analysis of linear systems" Ed. Addison-Wesley USA 1961.
- [8] De la Puente J. A. "Introducción a los Sistemas en Tiempo Real" Universidad Politécnica de Madrid, 2000.
- [9] Fan, H., y Nayeri, M., "Stability of some system identification techniques for underparametrized IIR adaptive filters", IEEE Transaction on Circuits and Systems, v2., 1989, pp 1748 - 1751.
- [10] Grewal M. and Andrews A. "Kalman filtering, theory and practice". Prentice Hall information and system sciences series. USA 1993.
- [11] Gnedenko, B. V., "The Theory of Probability", Ed. Chelsea Publishing Company, USA, 1962.
- [12] Gray D. "Introduction to the formal design of Real-Time Systems", Ed. Springer, UK 1999.
- [13] Guevara P., Medel J. J.. "Introducción a los sistemas en tiempo real". Publicado por la editorial Politécnico, octubre de 2003, tiraje de 1000 ejemplares. Registro de Derechos de Autor 03-2003-012912240400-01.
- [14] Gustafsson F. "Adaptive filtering and change detection". Ed. John Wiley & Sons, LTD, Linkoping University Sweden 2000.
- [15] Haykin S. "Adaptive filter theory". Prentice Hall information and system sciences series, USA, 1991.
- [16] Heitmeyer C., Mandrioli D. "Formal Methods for Real-Time Computing" Ed. Wiley, England 1996.
- [17] Jordan Chares, "Calculus of Finite Differences" Chelsea Publishing Company, New York.
- [18] Kailath T. "Linear Systems", Prentice Hall, USA 1980.
- [19] Krten R. "Getting started with QNX 4", PARSE Software Devices, Canada 1998.
- [20] Kuo B. C. "Sistemas de control automático". Prentice Hall 7ª edición, México, 1996.
- [21] Ifeachor E., Jervis B. "Digital Signal Processing". Addison-Wesley. USA 1993.
- [22] Liu J. "Real-time Systems" Ed. Prentice Hall USA 2000.
- [23] Liu C., Layland J. "Scheduling algorithms for multiprogramming in hard-real-time environment". Journal of the ACM, Vol. 20, No. 4, USA 1982.
- [24] Martín J. "Design Real-time computer systems", Ed. Diana México 1980.
- [25] Medel J. J. "Análisis de dos métodos de estimación para sistemas lineales estacionarios e invariantes en el tiempo con perturbaciones correlacionadas con el estado observable del tipo: Una entrada una salida" Computación y sistemas volumen 6 número 1, México 2002
- [26] Medel J. J. , y Poznyak A. S., "Adaptive Tracking for DC- derivate motor Based on Matriz Forgetting", Computación y Sistemas, pp. 201-217, 2001.

- [27] **Medel J. J., Guevara P. y Flores A.** "Constraints for Real-time Digital Filters". Automatic Control and Computer Sciences AVT No. 5 pages 63-69 ISSN 0132-4160, September 2003, (ISI), Rusia (2003)
- [28] **Nyquist H.** "Certain Topics in Telegraph Transmission Theory". AIEE Transactions, USA 1928.
- [29] **Papoulis A., Bertrán M.** "*Sistemas y circuitos*", Ed. Marcombo, 1ª edición, España 1989.
- [30] **Ramamritham K., Stankovic J.** "Scheduling algorithms and operating systems support for real-time systems". Proceedings of the IEEE, USA 1994.
- [31] **Rudín W.** "*Análisis real y complejo*". Universidad de Wisconsin, McGraw Hill.
- [32] **Söderström, T., and Stoica, P.,** "*Systems Identification*", Prentice Hall International, Hemel, Hempstead, Hertfordshire, England, 1988a.
- [33] **Söderström, T., y Stoica, P.,** "*On Some System Identification Techniques for Adaptive Filtering*", IEEE Transaction of Circuits and Systems, vol.35, Iss. 4, 1988b, pp. 457 - 461.
- [34] **Stankovic J.** "*Real-time Computing*". Technical Report, University of Massachusetts, 1992
- [35] **QNX Real-time Operating Systems**, <http://www.qnx.com> Canada 2001.
- [36] **Vicuña R., Guevara P.** "Elaboración de Administradores de Recursos en QNX para aplicaciones en Sistemas en Tiempo Real", Informe Técnico serie azul CIC-IPN México 2001.
- [37] **Wei, R and Kumar, P. R.,** "Stochastic adaptive prediction and model reference control", IEEE Transaction on Automatic Control, v. 39, Iss. 10, 1994, pp. 2047 - 2060.
- [38] **Young, S.,** "Real Time Languages Desing and Developments", Ellis Horwood, USA, 1982.



J. J. Medel. *Obtuvo los grados de Doctor en Ciencias (1998) y Maestro en Ciencias (1996) en Control Automático en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional. Actualmente es Profesor-Investigador del Centro de Investigación en Computación del Instituto Politécnico Nacional y miembro del Sistema Nacional de Investigadores nivel I, con varias publicaciones nacionales e internacionales en congresos, revistas de divulgación, revistas científicas con arbitraje internacional, trabajos publicados por editoriales de prestigio. Sus áreas de investigación son: Filtrado Digital, Teoría de Control y Sistemas en Tiempo Real.*



Pedro Guevara López. *Obtuvo el grado de Maestro en Ciencias de la Computación (1999) en el Centro de Investigación en Computación de donde actualmente es candidato a Doctor en Ciencias de la Computación; tiene el título de Ingeniero Electricista (1997) en la Escuela Superior de Ingeniería Mecánica y Eléctrica, ambos del Instituto Politécnico Nacional. Cuenta con varias publicaciones nacionales e internacionales en congresos, revistas y editoriales de prestigio. Sus áreas de investigación son: Filtrado Digital en Tiempo Real y Sistemas en Tiempo Real.*



Alberto Flores Rueda. *Tiene el grado de Maestro en Ciencias en Ingeniería de Cómputo con Especialidad en Sistemas Digitales (2000) obtenido en el Centro de Investigación en Computación y el título de Ingeniero Industrial (1998) de la Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas, ambos del Instituto Politécnico Nacional. Sus áreas de interés son: Procesamiento Digital de Señales y Dispositivos Programables.*