Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Computational Geometry & Artificial Vision, Malta, September 15-17, 2005 (pp71-76)

REAL-TIME DIGITAL FILTERS FOR SISO LINEAR STOCHASTIC SYSTEMS *

D. Cruz-Pérez[3], J. J. Medel[1], P. Guevara-López[2]

CIITEC[1,3]-**IPN***, CICATA[1,2,3]-**IPN** , DEMS[2]-**IPN**

Instituto Politécnico Nacional, Mexico City, MEXICO.

Abstract: The class of digital filters applied on PC, those who interact with dynamic processes and emit high quality output responses with time constraints and critical synchrony, will be described as RTDF (Real-time Digital Filters). The main RTDF properties will be described including a basic estimator example. The document is formed by the next sections:

a) *Digital Filter and Real-time Systems basically theories and main results of Real-time Digital Filtering.*
b) *The RTDF theoretical results: considering local and global constraints.*
c) *The RTDF implementation: considering a parameters estimator. In this part develops an extensive analysis of concurrent tasks: precedence constraints and times into a PC are considered.*

Keywords: Digital Filter, Synchronization, Real-time, Task, Interval.

1 Introduction

Digital Filters are applied in industrial processes, control systems and monitoring systems [5], [7]. For example, chemical plants, manufacturing processes, airbags and fuel injection systems, voice analysis, data acquisition, medical applications, telecommunications, missile trajectories, etc. The Digital Filters can't fail, in two senses: response quality and response time. If those did not happen, the processes can be crashed. For these reasons the Digital Filters must be specified and implemented as *Real-time Digital Filters (RTDF)*. A RTDF can be implemented into embedded systems [4] using micro controllers, DSP's, etc. In addition, a RTDF can be implemented in digital computers with Real-time Operating Systems (RTOS).

Then, *RTS* may be fast or slow depending of real system dynamics; this obeys the criterions exposed in [14]. In a PC, the whole of all activities are processed by *Real-time tasks* [11].

2 Real-time Digital Filter (RTDF)

With base on those previous references, the characterization of *Real-time Digital Filters (RTDF)* is absolutely necessary. In this section the basic properties of these systems in real time sense are explained. If the Digital Filter selected has interaction with dynamical process, and has constraints in time, it could be described as a RTDF, fulfilling the characteristics of real time systems [5] (see: *Fig. 1*).

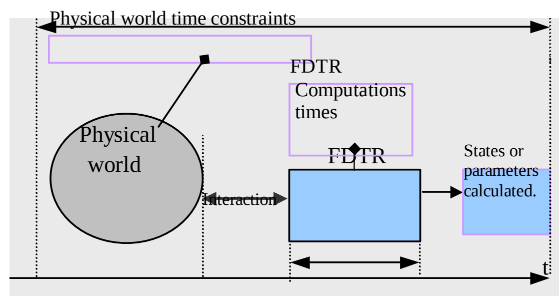

Fig. 1. Real-time Digital Filter.

Definition 1 (Real-Time Digital Filter). A RTDF is a Digital Filter with time constraints imposed by the dynamical process in the Nyquist sense [14], and:

a) *Receiving and giving input and output responses, respectively, in synchronized way with respect to process's dynamics.* Inputs and outputs RTDF will be expressed *symbolically* as $\{u(k)\}_g$ and $\{y(k)\}_i$, with $g, i, k \in \mathbf{N}^+$. *
b) *Giving correct set of responses with respect to dynamical process:* The quality response is defined in local and global senses considered in [8], and must guarantee stable conditions defined in [2] and [6].

* * g, i, *represent a probably number of inputs and outputs respect in a concurrent system that evolution to k intervals.*

Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Computational Geometry & Artificial Vision, Malta, September 15-17, 2005 (pp71-76)

c) *To express the RTDF in recursive way considering the concepts exposed by* Baras in [3] and Chui & Chen in [7], guarantee a minimal use of resources and memory, simulating the dynamics of the real process.

2.1 Implementation constraints of RTDF in a digital computer

When a RTDF is implemented in a computer with one processor, its different components use *concurrent Real-time tasks* (to see: [5] and [11]). The real-time task characteristics will be expressed as follows:

Definition 2 (RTDF : *local constraints*). The RTDF tasks have a lot of local constraints, imposed these by a lot of dynamical characteristics (in Nyquist sense [14]), with respect to real process.

The Sampling Period or Interaction time ($T(k)_i$): It is obtained by Sampling Criteria described by Nyquist in [14]. The RTDF evolution is bounded by Sample Period. The main characteristics of this are:

a. $T(k)_i := 1/f(k)_{i\_sampling}$. with $T(0)i = T$,
b. $\forall k\ \exists_i\ t(k)_i \ni t(k+1)_i - t(k)_i = T(k)_i$ and
c. $\mu[\ l(k)_i\ ,\ LD(k)_{i\_max}) := T(k)_i + \gamma_i$ where μ is a measurable function in the measure theory sense described in [1], and γ_i, represent a tiny time (named *jitter*). If $[\ l(k)_I\ , LD(k)_{i\_max})$ is empty, i.e., $\mu[l(k)_i\ , LD(k)_{i\_max})=0$ *at the sense described in* [1].

Definition 3 (RTDF : *global constraints***).** The RTDF has a global deadline, considering that the infimum value in agreement criteria described in [8], could be closed with respect to supremum value allowed by dynamical process.

2.2 FDTR: Global performance

In this section will be described the RTDF global properties in agreement to convergence functional $\{J(m)\}$ when the infimum value tend to ε with $m>0$, and $m \in \mathbf{N}^+$. The number m represent the interval when the RTDF converge, and ($m_i \uparrow m$, with $i = \overline{1,n}$).

Definition 4 (*Convergence: time* t_{i_c}). The time at which the RTDF converges, is expressed:

$$t_{i\_c} := f_i(k = m), \tag{1}$$

Where m is the RTDF convergence interval and t_{i_c} has the condition:

$$d_{i\_\bar{c}\min} \le t_{i\_c} < d_i. \tag{2}$$

Where d_i is a convergence deadline and and $d_{i\_c\_min}$ is a minimal convergence deadline imposed by physical world. Guarantying a response on time and synchronized with physical world. The shortest minimal convergence time is defined:

$$d_{i\_c\_min} := D(k)_{i\_min}, \tag{3}$$

Theorem 1. *The convergence error in probability sense defined by* J_m *in* [8] *and* [6], *has a value* ε_i *semi-positive defined, with respect to convergence time described in symbolic form by* t_{i_c}.

Proof. *Suppose that* ε_i *is lower than zero* ($\varepsilon_i<0$). *The convergence error defined in probability sense is described by the second moment (see for example:* [8]*) i.e.,* $M\{\Delta_t\Delta_t^T\}\ge\rho_t$, *with* ρ_t *positive semi defined, M represent the mathematical expectation operator and* Δ_t *is defined as difference between filtered value and real value (see:* [1]*). Now, considering that the* $\lim(M\{\Delta_t\Delta_t^T\}_{t\to d_i})\to\varepsilon_i$, *because the superior limit of* ρ_t *when* $t\to t_{i\_c}$, *is bounded by* ε_i, *and the inferior limit of* ρ_t *when* $t\to t_{i\_c}$, *is bounded by zero. Then* $\varepsilon_i\ge 0$.

2.3 RTDF local behavior (for TILS)

All RTDF are stable if the parameters are bounded by the unitary circle for all k ([10], [15] and [6]):

$$\left|\{a_e(k)\}\right|_i \le 1,\ \ e = \overline{1,n} \tag{4}$$

The estimated parameters $\{a_e(k)\}_i$ represent the proper values of modeled system [9], and it is stable in discrete sense, when the values have been into a unitary circle [10], [9], and [6]. Outside of unitary circle the response is unstable, and the filter has a bad construction with respect to [6] and [8].

Theorem 2 (Relative maximal deadline $D(k)_{i\_max}$). *A RTDF fulfill the follows condition:*

$$2 f_{\max}\left(D(k)_{i\_\max} - D(k)_{i\_\min}\right) < 1, \qquad 5)$$

f_{i_max} *is the dynamical process maximal frequency.*

Proof. *Considering the Nyquist criterion* [14], *the follows condition is true:*

$$\left(2f(k)_{i\_max}\right)^{-1} \geq \left(f(k)_{i\_muestreo}\right)^{-1}, \quad (6)$$

$$\left(2f(k)_{i\_max}\right)^{-1} \geq T(k)_i, \quad (7)$$

And also, considering Definition 2 section i, subsections a, b and c; the relative deadline differences is lower than sample time, i.e.:

$$T(k)_i > D(k)_{i\_max} - D(k)_{i\_min}. \quad (8)$$

Using transitivity in inequalities (7) *and* (8), *finally is clear how to obtain the inequality (*6).

2.4 RTDF Computational times and Deadlines

The RTDF diagram implementation is exposed in *Fig. 2*. In concurrent sense, the total computational time is defined by the sum of all tasks computing times.

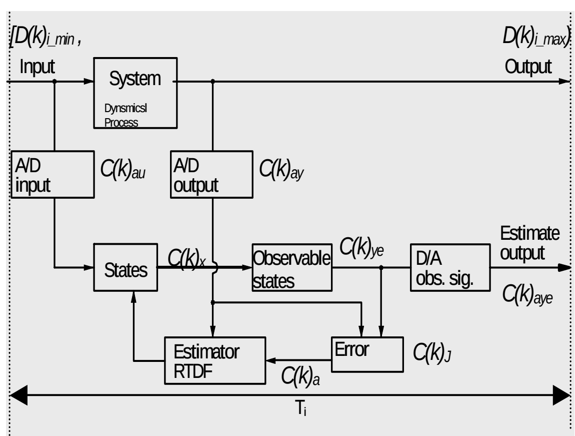

Fig. 2. RTDF: Diagram Implementation.

Where: $C(k)_x$: Computation time of state equation algorithm. $C(k)_y$: Computation time of observable signal equation algorithm. $C(k)_a$: Computation time of estimator equation algorithm. $C(k)_J$: Computation time of convergence error equation algorithm. $C(k)_{au}$: Computation time of A/D conversion of input *u(t)*. $C(k)_{ay}$: Computation time of A/D conversion of output *y(t)*. $C(k)_{aye}$: Computation time of D/A conversion of estimated output.

Generally in a RTDF that is implemented in a digital computer with one processor, all tasks around the filter will be scheduled in concurrent form; then the total time of RTDF will be conformed by the sum of computational times of total of all tasks:

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

Theorem 3 (*Computation time & deadline*). In a *hard RTDF, the computational time* $C(k)_i$ *will be bounded by deadline* $D(k)_{i\_max}$.

$$C(k)_i < D(k)_{i\_max} \quad (10)$$

Proof. *a) Suppose the case illustrated in the Fig. 3.*

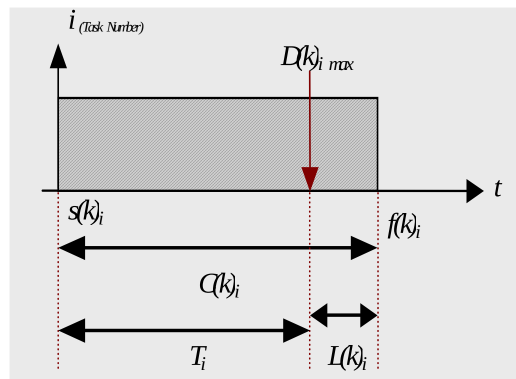

Fig. 3. Real-time task constraints.

Then, the following temporal constraints:

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

$$D(k)_{i\_min} = 0 \quad (12)$$

$$C(k)_i > D(k)_{i\ max} \quad (13)$$

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

$$D(k)_{i\_max} \leq T(k)_i\ . \quad (15)$$

and based in Fig. 3, the task time is described as:

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

Now, if $L(k)_i$ *is expressed as submultiples of* $D(k)_{i\_max}$*,:*

$$L(k)_i = D(k)_{i\_max}\, r^{-1}\ . \quad (17)$$

Where $r \in \mathbf{N}^+$, $r > 1$.Then, the equality (16) is:

$$C(k)_i = (1+\gamma)D(k)_{i\_max}. \quad (18)$$

With $\gamma := r^{-1}$ (19)

The minimal value respect to relative deadline ($D(k)_{I\_max}$), is r=-1, then this result implies that the starting hypothesis isn't satisfied.

b) In accordance with Definition 2:

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

and $f(k)_i \in [l(k)_i + D(k)_{i\_min}, l(k)_i + D(k)_{i\_max})$. (21)

Using section a, in this Proof.

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

Considering the worst case: $f(k)_i \rightarrow l(k)_i + D(k)_{i\_max}$, *and substituting in (22) we obtain:*

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

such that (23) is true, then the theorem is true. ;

Corollary 1: In agreement to [12] about processor assignment, a hard RTDF is formed by tasks, that are finished before theirs local deadlines $\{D(k)^j_{i\_max}\}$, and global deadline $D(k)_{i\_max}$, *where* $D(k)^j_{i\_max} \uparrow D(k)_{i\_max}$, *with* $j=1,\ldots,n$.

By contradiction, suppose that the global deadline obey the follows condition:

$$\exists\, i \text{ such as } C(k)_i > D(k)_{i\_max}\,, \quad (24)$$

The processor resources assignment in accordance with [12] are less than one, then:

$$C_{(k)i}\left(D(k)_{i\_max}\right)^{-1} \le 1, \quad (25)$$

The result expressed by (25), represents a basic contradiction because $C(k)_i < D(k)_{i:max}$ in agreement to Theorem 3, then this corollary is true. ;

Theorem 4 (Synchronization). *Whole responses of the RTDF would be synchronized with respect to temporal properties, all with respect to dynamical process, avoiding accumulative delay.*

Proof. *The finished times set* $\{f(k)_i\}$ *would be bounded by a intervals set* $\{[ld(h)_{i\_min}, LD(h)_{i\_max})\}$ *for all i, h, and* $k \in \mathbf{N}^+$, *such as* $\forall\, f(k)_I$:

$$f(k)_i \ge ld(h)_{i\_min} \text{ and, } f(k)_i < LD(h)_{i\_max}.$$

If and only if $h \equiv k$, *and this equality implies that RTDF would be synchronized with dynamical properties of physical process.*

By the other hand if:

a) *If h > k then, the RTDF generates in it accumulative delays and avoid synchronization.*

b) *If h<k then the RTDF generates accumulative time slices and avoid synchronization.*

In the last two points, the tasks of RTDF don't have synchronization with physical process.

3. RTDF Implementation example in a digital computer

We will implement a RTDF using this as parameter estimator applied to generic ARMA model.

To implement a RTDF into a PC, we used the following tool set:

- *QNX® 4.24 Real-time Operating system*
- *MicroPhoton Development Kit®,*

3.1 RTDF Characteristics

The model characteristics and RTDF are:

- The model is SISO ($i=1$), linear, stationary and time invariant defined by ARMA model in agreement to [8], [6],
- Least Squares Algorithm [8],
- Noises $v(k)$ and $w(k)$ are correlated with observable signal described by $y(k)$ but they aren't correlated.

The basic representation of the ARMA model is described as:

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

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

where: $x(k+1)$ is the system state, $y(k)$ is the observable signal, $v(k)$ and $w(k)$ are internal noise and external noise respectively, a is the parameter to estimate.

Estimator is expressed in the next form:

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

Where, in recursive form:

$$P(k) = P(k-1) + y(k)\, y(k-1), \quad (29)$$

$$B(k) = B(k-1) + y(k-1)\, y(k-1). \quad (30)$$

The error is defined by [8] as:

$$\Delta(k) := |a(k) - a|, \quad (31)$$

The functional of error defined in [8] and [6] is expressed as:

$$J(k)=E\,(\Delta(k)\,\Delta^{T}(k)) \qquad (32)$$

In the ARMA model, the following data will be considering for its implementation [13]: a= 0.35, $\sigma(v(k))$= 0.95, $\sigma(w(k))$= 0.99.

3.2 RTDF implementation

The scheme of tasks could be seen in the *Fig. 4* where each arrow signifies a message.

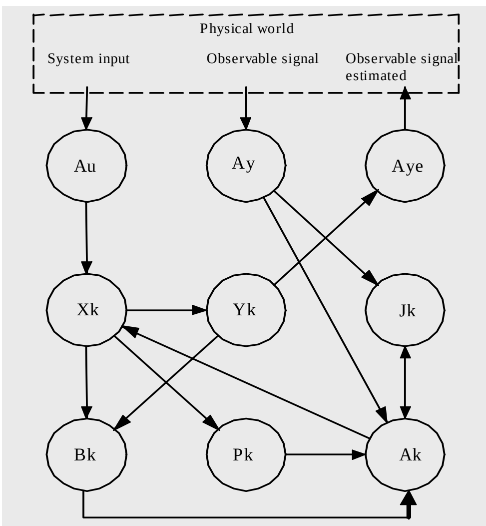

Fig. 4. To communicate and schedule RTDF tasks use the concepts of Message passing.

Experimental implementation of FDTR is made by concurrent tasks:

- Maximal relative deadline $D(k)_{max}$, is equal to period $T(k)$.
- Bounded starting time (s_p-$l(k)$ $\forall\, k$) was obtained by first probability moment, such as s_p = $l(k)+0.0015$ *ms* for each task.
- Sampling period $T(k)$ is 20 ms.
- Minimal relative deadline is $D(k)_{min}$= 2.5 ms. In experiment sense.
- Convergence deadline in agreement to stationary state is: d= 3 s.

The RTDF require creating next task: Xk: System states algorithm, C_{Xk}= 0.237 ms, Yk: Observable signal algorithm, C_{Yk}= 0.289 ms, Bk: Observable signal variance, C_{Bk}= 0.258 ms, Pk: Ricatti equation algorithm, C_{Pk}= 0.249 ms, Ak: Parameter estimator algorithm, C_{Ak}= 3.252 ms, Jk: functional of Error, C_{Jk}= 0.245 ms, Au: A/D conversion of input, C_{Au}= 0.310 ms, Ay: A/D conversion of observable signal, C_{Ay}= 0.302 ms, Aye: D/A conversion of estimate signal, C_{Aye}= 0.314 ms, O: Parent task, C_O = 0.261 ms. $C(k)$ value is in agreement to (9), is: 5.7305 ms.

In an equivalent diagram in time we obtain: $T(k)$= 20 ms, $l(k)$ = $T(k\text{-}1)k$ 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. The experimental convergence time is: m= 113 intervals, t_c=2.24 s, d= 3 s, t_c<d is complied. The estimate value is 0.41 (see: *Fig. 5*, *Fig. 6 and* [13]).

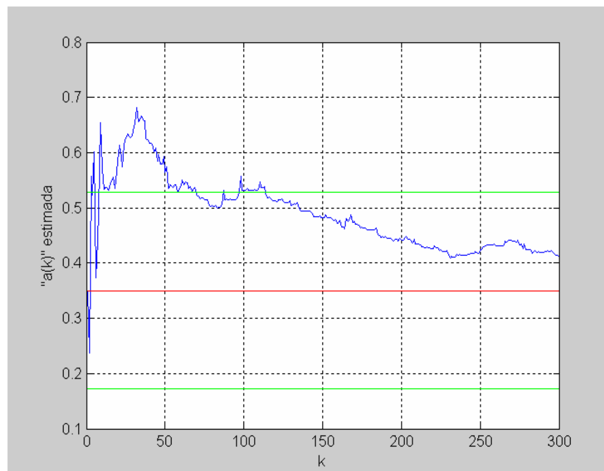

Fig. 5. RTDF: parameter estimator.

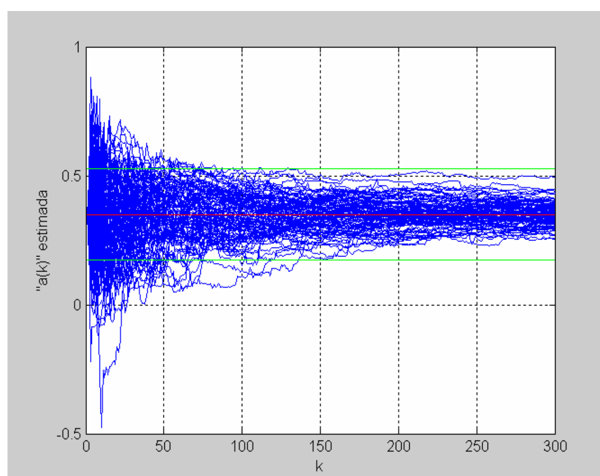

Fig. 6. RTDF: 100 experiments to parameter estimator.

4. Conclusions

Based on comments of different authors cited before, a RTDF obey several conditions: Interaction with physical world, good responses, time constraints and filter ability expressed in recursive form, and if we need that a RTDF be critical, then it has comply all time constraints to all cases and in each interval. Other important characteristic of RTDF is the ability to synchronizing it with dynamical process. A RTDF hasn't delay or pass the time constraints dictated by physical world, in another way a physical world is modify in not desired form. We observe that not all Digital Filters have characteristics of RTDF It is necessary to analyze many conditions: Physical world's dynamic, digital filter's structure, digital computer and operating system.

5 Bibliography

[1] **Ash R.** "*Real Analysis and Probability*", Ed. Academic Press, USA, 1970.

[2] Banach S., "The theory of Linear Operations", North-Holland Mathematical Library, 1991.

[3] **Baras J.** "*Symbolic and numeric Real-time signal processing*", Technical Report University of Maryland, USA 1999.

[4] **Burns A., Wellings A.** "*Real-time systems and programming languages*". University of York, Addison Wesley, 1997.

[5] **Buttazzo G.** "*Hard real-time computing systems*". Scuola Superiore S. Anna, Kluwer Academic Publishers, 1997.

[6] **Caines P**. "*Linear Stochastic Systems*". Ed. Wiley, Canada, 1986.

[7] **Chui C., Chen G**. "*Kalman Filtering with Real-time Applications*". Ed. Springer, USA 1999.

[8] **Haykin S**. "*Adaptive filter theory*". Prentice Hall information and system sciences series, USA, 1991.

[9] **Kailath** T. "Linear Systems", Prentice Hall, USA 1980.

[10] **Kuo B. C.** "*Sistemas de control automático*". Prentice Hall 7ª edición, México, 1996.

[11] **Liu J**. "*Real-time Systems*" Ed. Prentice Hall USA 2000.

[12] **Liu C., Layland J.** "*Scheduling algorithms for multiprogramming in hard-real-time environment*". Journal of the ACM, Vol. 20, No. 4, USA 1982.

[13] **Medel J. J.** "Análisis de dos métodos de estimación para sistemas lineales estacionarios e invariantes en el tiempo" Computación y sistemas volumen 6 número 1, México 2002.

[14] **Nyquist H.** *Certain Topics in Telegraph Transmission Theory*. AIEE Transactions, USA 1928.

[15] **Rudín W.** (1988). *Análisis real y complejo.* Universidad de Wisconsin, McGraw Hill.