REAL-TIME CONSTRAINTS FOR DIGITAL FILTERS TO PARAMETERS ESTIMATION ON MONOVARIABLE LINEAR STOCHASTIC SYSTEMS

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

INSTITUTO POLITECNICO NACIONAL, MEXICO D. F., MEXICO.
CIITEC-IPN [*1,3], CICATA-IPN [*1,2,3], DEMS-IPN [*2]
{jjmedel1, pguevara2}@ipn.mx, dcruzp@yahoo.com3

Abstract The class of digital filters applied into PC and those interacts 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 D. C. motor 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. Digital Filters can't fail, in two senses: response quality and response time, or the processes can be crashed. For these reasons 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).

2 Digital Filters

The filter concept, is commonly used to describe a physic device characteristic, using the information given by the *system* (to see: [13] and [9]); such that, the filter is applied to a set of data generated by a corrupted signal. To speak about the information, it means to consider a lot of states of the system combined with a lot of perturbations (these perturbations have two sources around to the system: a) External and b) internal. The noises or perturbations may arise from a variety of sources). The filter process is based on: a) Monitoring states (issued from the system), and b) Forecast or identify noises (originated from emitter, receptor or environment of system). The filter theory has two basic areas, which describe the principal properties about the essential system characteristics (inside and outside qualities of it) [13], [21]: *Identification* and *estimation*. In both cases, the filters use the information emitting by the system. Error functional $J(k)$, (estimation or identification) must be converge to minimal value in finite time [13], [6]; the convergence criterion is defined by a difference between the filtering signal and original signal and applying on it, the second probability moment in this result [13], [21]. Examples of Digital Filters are: Wiener, Kalman [13] and Medel & Poznyak filter [22].

3 Real-time Systems (RTS)

Many authors define a Real-time Systems (RTS), (for example: Martin in [20], Burns & Wellings in [4], Heitmeyer & Mandrioli in [14], Gray in [11], De la Puente in [8], Stankovic in [26], in the others). And all these authors conclude in the same way: *Real-time System (RTS) is a system that suffice three conditions: a) interaction with physical world, b) correct responses, time constraints from physical world. RTS* must be synchronized with physical world and this velocity is relative, depending of dynamics of real system and the filter algorithm, and the computing system. Then, *RTS* may be fast or slow depending of real system dynamics; this obeys the criterions exposed in [23]. In a PC, the whole of all activities are processed by *Real-time tasks* [18].

4 State of Real-time Digital Filters

Chui & Chen exposing in [7] that Real-time Digital Filters (Kalman for example) can be implanted in *RTS* considering the high velocity computers and facility to express the filter in recursive form. Fredrik Gustafsson describing in [12] that FIR (*Finite Impulse Response*) Digital Filters as "applications to Real-time processing of standard signals". Papoulis & Bertran suggesting in [24] that the implementation of Digital Filter into PC, is through to consider the times measure of all tasks around the implementation of it into PC (considering the measurable: The A/D and D/A converters, processor operations, filter algorithm and precedence constraints). But the concepts about Real-time Digital Filters never were cited: Jane Lui presenting an example of Kalman Filter in [18], but she didn't justify the Real-time characteristics of it, only said that it is a Real-time application. Baras in [3] described the Real-time Signal Processing only if it is expressed in recursive form.

5 Real-time Digital Filter (RTDF)

In base of these 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, could be described as a RTDF, complied the characteristics of real time systems [But97].

Definition 1 (Real-Time Digital Filter). A RTDF is a Digital Filter with time constraints imposed by the dynamical process in the Nyquist sense [23], and:

1. *Receiving* and *giving* input and output responses, respectively, in synchronized form with respect to dynamics of the process. Inputs and outputs RTDF will be expressed in symbolic form as $\{u(k)\}_g$ and $\{y(k)\}_i$, with g, i, $k \in N^+$. [*],
2. *Giving correct responses set with respect to dynamical process:* The quality response is defined in local and global senses considered in [13], and must guarantee stable conditions defined in [2] and [6] and
3. *Express the RTDF in recursive form 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.

5.1 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 [18]). The real-time task characteristics will be expressed in the following:

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 [23]), with respect to real process. In Fig. 1, will be illustrate it.

---

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

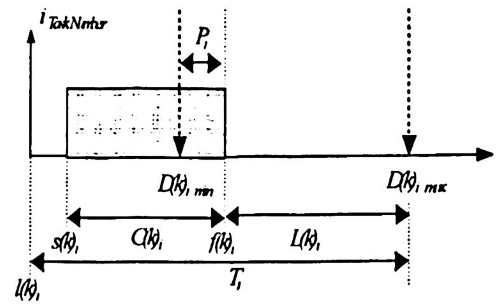

Fig. 1 RTDF as Real-time task

a) Arrival time. ($I(k)_i$) [5]: Is the time when task becomes ready for execution. $I(k)_i > 0$ $\forall k \in N^+$., *b) Computation time* ($C(k)_i$): Is the total time in a *instance* (to see: [5]) of a task obtained by union of times used for atomic *activities*., *c) Deadline minimum relative* ($D(k)_{i\_min}$): Is the minimum time in which a task should be finished[2]. It is a function of $I(k)_i$., d) *Deadline Maximum relative* ($D(k)_{i\_Max}$): Is the maximum time which a task should be accomplished[3], bounded it by the right side with sample time. This deadline is a function too of time $I(k)$, and depend of system., e) *Start time* ($s(k)_i$): Is a time when the task start with execution. This time depend of: Resource availability, latency times, context change, size of ready queue, etc. Start time is considered a stochastic variable [10] with a range named *jitter*, [18], and fulfill the condition $s(k)_i < D(k)_{i\ max}$., f) *Finishing Time* ($f(k)_i$): Is the time when the task execution finished under interval time k. In mathematical sense $f(k)_i \in [Id(k)_{i\_min}, LD(k)_{i\ max})$, with $Id(k)_{i\_min} := I(k)_i + D(k)_{i\ min}$ and $LD(k)_{i\_max} := I(k)_i + D(k)_{i\ max}$[4]., *g) Lateness* ($L(k)_i$): Is the loss time between finishing time and final interval time k. This time is $L(k)_i := |LD(k)_{i\ max} - f(k)_i|$., h) *Premature time* ($P(k)_i$): Is the gain in time generated by finishes task before maximum deadline. This time is defined by $P(k)_i := |Id(k)_{i\ min} - f(k)_i|$, where $f(k)_i \geq ID(k)_{i\ min}$ generate a premature task answer., i) *Sampling Period or Interaction time* ($T(k)_i$): It is obtained by Sampling Criteria described by Nyquist in [23]. 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. $\forall k\ \exists_j\ t(k)_i \ni t(k+1)_i - t(k)_i = T(k)_i$ and, c. $\mu[I(k)_i, LD(k)_{i\ max}) := T(k)_i + \gamma_i$ where μ is a measurable function in the measure theory sense described in [1], and γ_i, represent a tiny time (*jitter*). If $[I(k)_i, LD(k)_{i\ max})$ is empty, i.e., $\mu[I(k)_i, LD(k)_{i\ max}) = 0$ *at the sense described in* [1].

---

[2] If the response is obtained before this lower deadline, it means that the system has a null task In the other hand, if the time is higher that dynamic interval defined by dynamic system period time.

[3] If response is obtained after this deadline, it is bad.

[4] If we sum $I(k)_i$ to $D(k)_{i\ min}$, $D(k)_{i\ max}$ obtain absolute deadlines ($Id(k)_{i\_min}$, $LD(k)_{i\ max}$) [Liu00].

Definition 3 (**RTDF** : *global constraints*). The RTDF whole tasks have a global deadline, considering that the infimum value in agreement criteria described in [13], could be closed with respect to supremum value allowed by dynamical process.

5.1 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 $m \in N^{+}$. 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:

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

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

$$d_{i\_c\_min} \leq t_{i\_c} < d_i. \quad (2)$$

When 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:

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

Theorem 1. *The convergence error in probability sense defined by* J_m *in [13] 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: [13]) i.e., $M\{\Delta_t\Delta_t^T\}\geq\rho_t$, 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 (to 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 $\varepsilon_i\geq 0$.

5.3 RTDF local behavior (for TILS)

All RTDF are stable if the parameters are bounded by the unitary circle for all k ([17], [25] and [6]):

$$\{a_e(k)\}_i \leq 1,\ e=\overline{1,n} \quad (4)$$

The estimated parameters whole $\{a_e(k)\}$, represent the proper values of modeled system [15], and it is stable in discrete sense, when the values have been into a unitary circle [17],

[15], and [6]. Outside of unitary circle the response is unstable, and the filter has a bad construction with respect to [6] and [13].

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

$$2f_{max}(D(k)_{i\_max}-D(k)_{i\_min}) < 1, \tag{5}$$

f_{i_max} is the dynamical process maximal frequency.

Proof. *Considering the Nyquist criterion [23], the follows condition is true:*

$$f(k)_{i\_muestreo} \geq 2f(k)_{i\_max}, \tag{6}$$

$$2T(k)f(k)_{i\_max} \leq 1, \tag{7}$$

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

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

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

5.4 RTDF Computational times and Deadlines

In concurrent sense, the total computational time is defined by the sum of all computing times of tasks.

Generally a RTDF that is implanted in a digital computer with a one processor, the whole of the tasks around of filter will be scheduled in concurrent form:

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

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.

To implement a RTDF into PC, required a tool set: *a)D.C. Motor, 20 V, 1 A, 1800 rpm, two poles, permanent field., b) Power unit A/D, D/A, 5 V input, 20 V output, 0.05 A input, 5 A output.*, c) *PC Pentium III 400 MHz, 64 MB RAM., d) ADC card PCL 818L, e) QNX® 4.24 Real-time Operating system., f) MicroPhoton Development Kit®.*

The RTDF ini agreement to (9), 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: Error functional, 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 form in time, we obtain: *T(k)*= 20 ms, *I(k)* = *T(k-1)k* ms, *s(k)*= *I(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 convergence time proved experimentally is: *m*= 113 intervals, t_c=2.24 s, *d*= 3 s, $t_c<d$ is complied. The time value is 0.41 (To see: [21]).

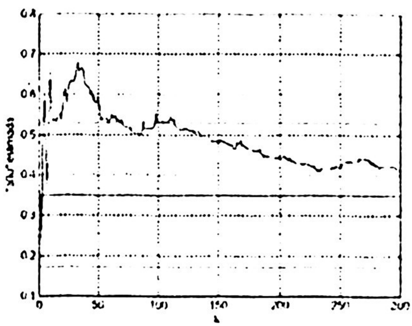

Fig. 2 RTDF: parameter estimator.

6 Conclusions

With base 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 to comply all time constraints in 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, is necessary to analyze many conditions: Physical world dynamic, digital filter structure, digital computer, operating system, in the others.

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] De la Puente J. A. "Introducción a los Sistemas en Tiempo Real" Universidad Politécnica de Madrid, 2000.
[9] Grewal M. and Andrews A. Kalman filtering, theory and practice. Prentice Hall: Information and System Sciences Series. USA 1993.
[10] nedenko, B. V., the Theory of Probability, Ed. Chelsea Publising Company, USA, 1962.
[11] Gray D. "Introduction to the formal design of Real-Time Systems", Ed. Springer, UK1999.
[12] Gustafsson F. "Adaptive filtering and change detection". Ed. John Wiley & Sons, LTD, Linkoping University Sweden 2000.
[13] Haykin S. "Adaptive filter theory". Prentice Hall: Information and System Sciences Series, USA, 1991.
[14] Heitmeyer C., Mandrioli D. "Formal Methods for Real-Time Computing" Ed. Wiley, England 1996.
[15] Kailath T. "Linear Systems", Prentice Hall, USA 1980.
[16] Krten R. Getting started with QNX 4, PARSE Software Devices, and Canada 1998.
[17] Kuo B. C. "Sistemas de control automático". Prentice Hall 7ª edición, México, 1996.
[18] Liu J. "Real-time Systems" Ed. Prentice Hall USA 2000.
[19] Liu C., Layland J. "Scheduling algorithms for multiprogramming in hard-real-time environment". ACM, Vol. 20, No. 4, USA 1982.
[20] Martín J. "Design Real-time computer systems", Ed. Diana México 1980.
[21] 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.
[22] Medel J. J ., & Poznyak A. S., Adaptive Tracking for DC- derivate motor Based on Matriz Forgetting, C y S, pp. 201-217, 2001.
[23] Nyquist H. Certain Topics in Telegraph Transmission Theory. AIEE Transactions, USA 1928.
[24] Papoulis A., Bertrán M. "Sistemas y circuitos", Ed. Marcombo, 1ª edición, España 1989.
[25] Rudín W. (1988). Análisis real y complejo. Universidad de Wisconsin, McGraw Hill.
[26] Stankovic J. "Real-time Computing". Technical Report, University of Massachusetts, 1992