Rev. Fac. Ing. Univ. Antioquia N.° 61 pp. 123-131. Diciembre, 2011

Description of execution time dynamics for a set of concurrent real-time tasks

Descripción de la dinámica de los tiempos de ejecución de un conjunto de tareas en tiempo real concurrentes

Pedro Guevara López[1*], *José de Jesús Medel Juárez*[2], *Gustavo Delgado Reyes*[1]

[1]ESIME–IPN. Av. Santa Ana No. 1000. San Francisco Culhuacán. C.P. 04430 México D.F.

[2] CIC, Col. Nueva Industrial Vallejo. C.P. 07738. México D. F.

(Recibido el 1 de mayo de 2011. Aceptado el 17 de noviembre de 2011)

Abstract

A Real-time System (RTS) implemented in a digital computer interacts with the physical world through knowledge variables (sensors, actuators, Analogical - Digital, and Digital-Analogical convertors) processing its requirements through real-time tasks (J_i: $i\in Z_+$). Each J_i is formed by a set of instances {$j_{i,k}$: i, $k\in Z_+$, i is the task index, k is the instance index} with at least three time constraints: arrival, execution and, interval times ($L_{i,k}$, $C_{i,k}$, $D_{i,k}$). The execution time identification $\hat{C}_{i,k}$, is based on Instrumental Variable parameter estimation and the recursive state space ARMA model. The execution times were measured experimentally using a Real-time Operative System QNX 6.5 Neutrino in an Intel Core i7 processor with a speed of 2.66 GHz. This paper presents a state of the art real-time task model, execution time measurements, an Instrumental Variable parameter estimator on recursive state space identification, experimentation and, results.

----- *Keywords:* Tasks, Real-time, estimation, identifier, computing time, instance, error

Resumen

Un Sistema en Tiempo Real (STR) implantado en una computadora digital, interactúa con el mundo físico a través del acondicionamiento de variables (sensores, actuadores convertidores Analógico/Digital A/D y Digital/

* Autor de correspondencia: tel: + 52 + 55 + 562 420 00 ext. 73250, fax: + 52 + 55 + 565 620 58, correo electrónico: pguevara@ipn.mx (P. Guevara)

Analógico D/A) y procesa sus peticiones mediante tareas en tiempo real (J_i: $i \in Z_+$). Cada J_i está formada por un conjunto de instancias {$j_{i,k}$: $i, k \in Z_+$, i es el índice de la tarea, k es el índice de la instancia} con al menos tres restricciones temporales: tiempo de arribo, tiempo de ejecución y plazo ($L_{i,k}$, $C_{i,k}$, $D_{i,k}$). En este sentido, es importante proponer un modelo para reconstruir el comportamiento de los tiempos de ejecución $C_{i,k}$. El modelo propuesto es un ARMA con un estimador de parámetros basado en la variable instrumental. Los tiempos de ejecución fueron medidos experimentalmente en QNX Neutrino 6.5 utilizando un procesador Intel Core i7 de 2.66 GHz. Se presentan los antecedentes teóricos del modelado de tareas en tiempo real, las mediciones de tiempo de ejecución, el modelo de tiempos de ejecución, el estimador de parámetros con variable instrumental, la experimentación y los resultados.

----- *Palabras clave*: Tarea, tiempo real, estimador, identificador, tiempo de ejecución, instancia, error

Introduction

Real-Time Systems (STR) are present in industrialized societies in all electronic devices: televisions, washing machines, microwave ovens, cell phones and digital telephone exchanges; airplanes, trains, automobiles, ensuring the processes considered safety in general operations. In [1], a Real Time System (RTS) actively interacts with the environment with time constraints. The RTS implanted in a Real Time Operating System (RTOS) such as QNX Neutrino is formed by a set of instances {$j_{i,k}$} inside of a real time task J_i, where *i* is the index belonging to the task and, *k* is the instance index. Each instance is started by a service request.

The model considers the dynamic properties to describe execution times $C_{i,k}$ determining the processor usage in each task.

Modeling execution time of real-time tasks

This work was developed in QNX Neutrino 6.5, modeling real-time tasks, considering time constraints. The STR implanted a QNX Neutrino operating system, composed of a set of concurrent tasks. The turnout has the potential parallelism in the execution tasks when their number exceeds the number of processors available by computer. The RTOS QNX Neutrino is complete and robust according to [2], and is adaptable to the resource constraints of embedded RTS.

The paper described in [3] summarizes what some authors consider the specific execution times for the worst case (WCET) and not as alternatives to probabilistic behavior, describing the execution times and duration periods of a real-time task. In [1], real-time tasks are classified according to time constraints, considering the arrival time, execution and deadline times. These are classified as periodic, sporadic and non periodic tasks according to time conditions. In [4], the models were limited considering fixed execution times, but the variations were different between the identified and the results of the computer system. In [5], an identification of the worst case execution time (WCET) considered the variations due to external factors, such as, pipeline, route search, loop number, and mutual exclusion. In [6], a WCET probability distribution function tool analyzed this in a real-time program, depending on factors such as, memory cache, forecasts, pipelines and other interactions. In [7], a stochastic ARMA model represented the behavior of arrival times, estimating the parameter using a digital filter.

Real-time tasks time constraints

In [1], a real-time task J_i it is described as an executable job entity characterized by arrival times and restriction times associated with an instance. Where i is the task index, and k represents the instance index related to a time interval according to Nyquist [8]. The real-time task is defined by the following quintuple ($L_{i,k}$, $S_{i,k}$, $C_{i,k}$, $F_{i,k}$, $D_{i,k}$), $i,k \in Z_+$ where the temporal conditions are arrival $L_{i,k}$, start S_k and, execution time $C_{i,k}$ in which the system finishes its operations in completion time $F_{i,k}$ and bounded by maximum deadline $D_{i,k}$. Figure 1 shows a block diagram with temporal elements that make up an instance of a real-time task.

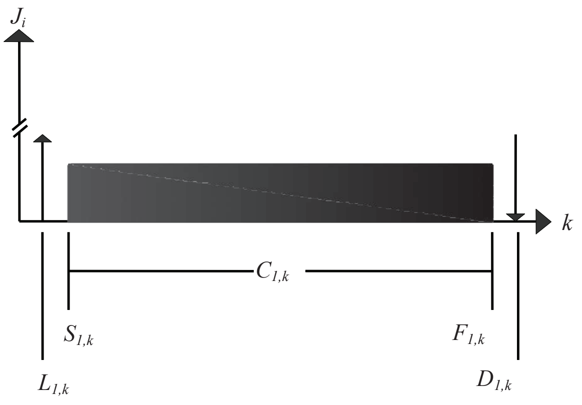


Figure 1 Time constraints for an instance of real-time task J_1 with five temporary restrictions

Execution time model for real-time task

Execution time C_k is the time in which the instance with index k of a real-time task J_i computing its operations, disregarding the preemption task, this is shown in figure 2.

The execution time vector $\mathbf{C}_k$ of a set of concurrent real-time tasks J for an k interval is defined in (1).

$$\mathbf{C}_k = \begin{bmatrix} C_{1,k} & C_{2,k} & \ldots & C_{n,k} \end{bmatrix}^T \tag{1}$$

In [9], the states contain disturbance errors read by the receiver (oscilloscopes, data acquisition cards, etc.), influenced by the environment surrounding sender and receiver. In [5, 6] consider variations of the execution times due to the search for the best execution path and other internal interactions. In [10], the disturbances are due to events outside the digital system, and were modeled based on [1] with an ARMA described in (2) and (3).

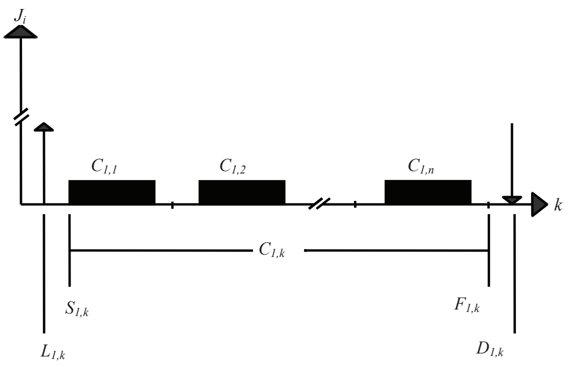


Figure 2 Executions time $C_{i,k}$ for n instances in a real-time task

$$X_{k+1} = AX_k + V_k \tag{2}$$

$$C_k = X_k + W_k + H_k \tag{3}$$

Where C_k is a execution time of set of concurrent real-time task in the *k-th* instance, A_k is the system parameter, V_k is a computer internal noise, W_k is a computer external noise, H_k is a execution time reference and X_k is an internal execution time model state. Recursive execution time from a concurrent real-time task is described in Theorem 1 and (4). ***Theorem 1***. *The concurrent real-time task described by (2) and (3), has a recursive execution time form (4).*

$$C_k = AC_{k-1} + \widetilde{W}_k \tag{4}$$

Where $\widetilde{W}_k$ represent the generalized noises. ***Proof.*** *Assume that X_{k+1} described in (2) is delayed in time and yields (5).*

$$X_k = AX_{k-1} + V_{k-1} \tag{5}$$

From (3), the internal state of the run-time function C_k is achieved with a delay (6).

$$X_{k-1} = C_{k-1} - W_{k-1} - H_{k-1} \quad (6)$$

With (6) in (5), it is possible to obtain (7).

$$X_k = AC_{k-1} - AW_{k-1} - AH_{k-1} + V_{k-1} \quad (7)$$

With (7) in (3), it is possible to obtain (8).

$$C_k = AC_{k-1} - AW_{k-1} - AH_{k-1} + V_{k-1} + W_k + H_k \quad (8)$$

In (8), it is describe

$$\widetilde{W}_k = AW_{k-1} - AH_{k-1} + V_{k-1} + W_k + H_k,$$

as generalized noises.■

The *recursive execution time form (4)* has an explicit matrix parameter *A* with respect to concurrent real-time task response times. The generalized noises $\widetilde{W}_k$ would only be described affecting the response, through internal gains excitation.

Instrumental variable execution times parameter estimation

In [9], the black box system has an estimation process, based on matrix digital filter description. In [11], three parameter estimator comparisons were presented: Recursive Least Squares (MSM), Recursive Gradient (MGR) and, Instrumental Variable Methods (MVI).

Theorem 2. A *concurrent real-time task parameters execution time model according to (6) and using an instrumental variable is presented in (9).*

$$\hat{A}_n = P_k B_k^+ \quad (9)$$

Where the estimator (9) is represented considering the second probability stage

$$P_k = E\left\{ C_k Z_k^T \Big|_{\Im C_k} \right\} \quad \text{and} \quad B_k = E\left\{ C_{k-1} Z_k^T \Big|_{\Im C_k} \right\},$$

respectively.

Proof. *According to (4) using the second probability moment, (10) is obtained.*

$$E\left\{ C_k Z_k^T \Big|_{\Im C_k} \right\} = AE\left\{ C_{k-1} Z_k^T \Big|_{\Im C_k} \right\}^+ E\left\{ \widetilde{W}_k Z_k^T \Big|_{\Im C_k} \right\} \quad (10)$$

The properties described in (11) and (12) satisfy (10)

$$\begin{gathered} E\{W_{k+1} Z_k^T \big|_{\Im C_k}\} = 0, \; E\{W_k Z_k^T \big|_{\Im C_k}\} = 0 \\ E\{V_k Z_k^T \big|_{\Im C_k}\} = 0, \; E\{V_k W_k^T \big|_{\Im C_k}\} = 0 \end{gathered} \quad (11)$$

$$E\{W_k W_k^T \big|_{\Im C_k}\} = \sigma^2{}_{w_k}, \quad E\{V_k V_k^T \big|_{\Im C_k}\} = \sigma^2{}_{v_k} \quad (12)$$

Where Z_k is the instrumental variable according to [11], described in (13).

$$Z_k := f(C_{k-i}), \; i = \overline{0,m}, \;\; \ni C_k \perp W_k, \; C_k \perp V_k \quad (13)$$

From expression (10), pseudo-inverse of B_k allows describing the internal gain (9). The recursive form of P_k *and* B_k is shown in (14) and (15), respectively.

$$P_k = P_{k-1} + Y_k Z_k^T \quad (14)$$

$$B_k = B_{k-1} + Y_{k-1} Z_k^T \,. \;\blacksquare \quad (15)$$

Theorem 3*. The estimator described in (9) uses the instrumental variable and converges at almost all points, shown in (16).*

$$\hat{A}_k \underset{k\to\infty}{\rightarrow} A \quad (16)$$

Proof. *According to the execution times of a concurrent real-time recursive form task described in (4), the identifier* $\hat{C}_k$ *is formed in (17)*

$$\hat{A}_k C_{k-1} + \Xi_k \quad (17)$$

This leads to the conclusion that the identification error is described in (18).

$$\Delta(C_k) = \hat{C}_k - C_k \tag{18}$$

Considered (4) and (17) in (18), the difference resulted in (19).

$$\Delta(C_k) = (\hat{A}_k - A)C_{k-1} + (\Xi_k - \widetilde{W}_k) \tag{19}$$

The stochastic gradient of (19) with respect to execution time, obtains (16). ■

The functional error G_k described in (20) was obtained considering the second probability moment, in recursive form.

$$G_k = \tfrac{1}{k}\left((k-1)G_{k-1} + \Delta(C_k)\right) \tag{20}$$

Results and discussion

In [12], different methods of measuring the performance of the instances execution times of a real-time task are presented. It used a clock () function in library time.h in the POSIX systems. This method has the advantage of more information details configured as a *clock resolution* in milli-seconds of the program code shown in figure 3. In this investigation, execution times were measured experimentally in the QNX® Neutrino RTOS 6.5 system, using an Intel® Core i7 2.66 GHz. The experiment under consideration is the simulation of a DC motor described in (21) and presented in more detail in [13].

$$\begin{bmatrix} \frac{di_a(t)}{dt} \\ \frac{dw_r(t)}{dt} \end{bmatrix} = -\begin{bmatrix} \frac{R_a}{L_a} & \frac{k_a i_f}{L_a} \\ \frac{k_a i_f}{J} & \frac{b}{J} \end{bmatrix} \begin{bmatrix} i_a(t) \\ w_r(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & \frac{1}{J} \end{bmatrix} \begin{bmatrix} V_a(t) \\ T_D(t) \end{bmatrix} \tag{21}$$

Where $i_a(A)$ is the armature current, $w_r\,(rad/sec)$ angular velocity, $R_a\,(\Omega)$ armature resistance, The $L_a(H)$ armature inductance, J (kgm^2) torque, ε the difference conditions, $i_f(A)$ is the stator current, k_a engine construction, b constant friction viscosity, $V_a\,(V)$ armature voltage, $T_D\,(Nm)$ motor load.

The state space (27) was discreetly computed considered finite differences, becoming in (22).

$$\begin{bmatrix} i_a(\tau) \\ w_r(\tau) \end{bmatrix} = \begin{bmatrix} 1-\varepsilon\frac{R_a}{L_a} & -\varepsilon\frac{k_a i_f}{L_a} \\ -\varepsilon\frac{k_a i_f}{J} & 1-\varepsilon\frac{b}{J} \end{bmatrix}^{+} \begin{bmatrix} i_a(\tau-1) \\ w_r(\tau-1) \end{bmatrix} + \begin{bmatrix} \varepsilon\frac{1}{L_a} & 0 \\ 0 & \varepsilon\frac{1}{J} \end{bmatrix} \begin{bmatrix} V_a(\tau) \\ T_D(\tau) \end{bmatrix} \tag{22}$$

(22) is transformed symbolically in state space shown in (23).

$$S_k = \widetilde{M}S_{k-1} + \widetilde{B}\tilde{N}_{k-1} \tag{23}$$

Where the matrix parameters $\widetilde{M}$ and $\widetilde{B}$ are shown in (24).

$$\widetilde{M} := \begin{bmatrix} 1-\varepsilon\frac{R_a}{L_a} & -\varepsilon\frac{k_a i_f}{L_a} \\ -\varepsilon\frac{k_a i_f}{J} & 1-\varepsilon\frac{b}{J} \end{bmatrix} \text{ and } \widetilde{B} := \begin{bmatrix} \varepsilon\frac{1}{L_a} & 0 \\ 0 & \varepsilon\frac{1}{J} \end{bmatrix} \tag{24}$$

The simulation considered the variables involved in (4) and (5), which were programmed using a proportional controller, establishing a measurement governed by the International Units System shown in table 1.

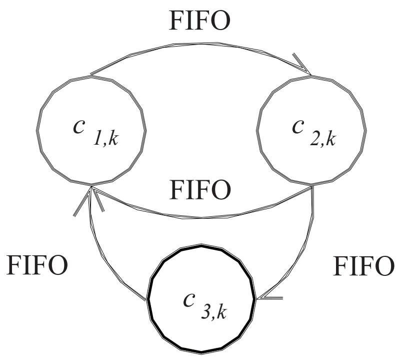


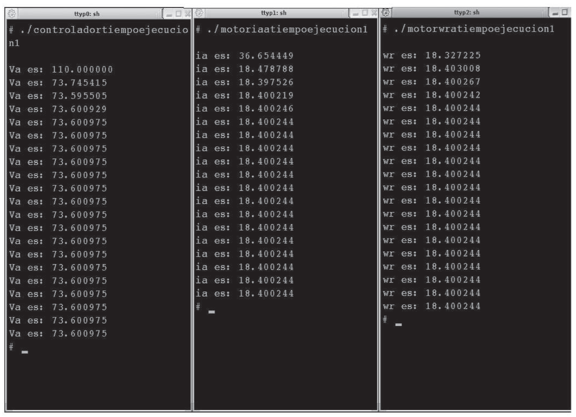
Figure 3 FIFO communication mechanism concurrent real-time tasks diagram

Table 1 Variable definitions

Variables	*Proposed value*
J	1 kgm^2
B	1 Nm(s/rad)
k_a	0,05
i_f	20 A
L_a	0,001 H
R_a	3 Ω
V_{ref}	110 V
T_D	0 – 25 Nm
K	1,36 – 1,7

Three tasks were scheduled in a vector: $\mathbf{C}_k = [C_{1,k}, C_{2,k}\ C_{3,k}]^T$ for DC motor dynamics simulation, according to equations (9), (14), (15) and, (22), where the matrix parameter was obtained in (25), and applied in (4) describing the identification $_k$. The communications among tasks are performed by FIFO pipelines as shown in figure 3.

Each task is executed in a different QNX terminal, shown in figure 4. The simulation results considering the torque load due: $T_D = 0$ *Nm*, a reference voltage = 110V and a gain in the feedback, k = 1.36, presenting the first 20 instances.



```
# ./controladortiempoejecucio
n1

Va es: 110.000000
Va es: 73.745415
Va es: 73.595505
Va es: 73.600929
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
Va es: 73.600975
#
```

```
# ./motoriaatiempoejecucion1

ia es: 36.654449
ia es: 18.478788
ia es: 18.397526
ia es: 18.400219
ia es: 18.400246
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
ia es: 18.400244
#
```

```
# ./motorwratiempoejecucion1

wr es: 18.327225
wr es: 18.403008
wr es: 18.400267
wr es: 18.400242
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
wr es: 18.400244
#
```

Figure 4 DC motor simulation described for three concurrent tasks operating in different terminals

This work measures the execution time tasks of $C_{1,k}$, $C_{2,k}$, $C_{3,k}$, (see figure 5). A proposed reconstruction of the dynamics through (4) and parameter estimation of execution times according to (9) obtained the results shown in figure 6.

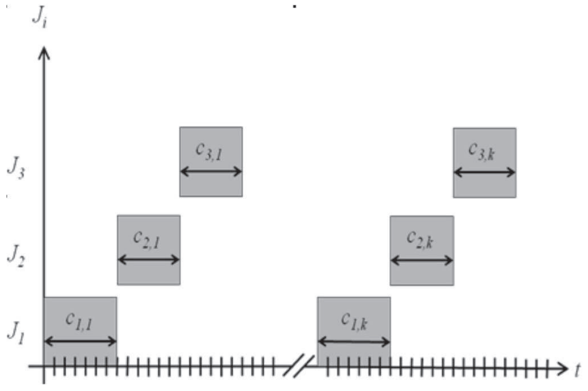


Figure 5 Time diagram of real-time concurrent tasks for DC motor

Figure 7 shows three execution times measured of the algorithm programmed into the real-time QNX operating system, using the code segment shown in figure 6, through 500 instances. The three execution time shown in this figure correspond to each processes that were implemented using the Real-time Operative System QNX 6.5 Neutrino.

```
#include<stdio.h>
#include<time.h>
#include<sched.h>
#include<fcntl.h>
void main(void)
{
double inicio, fin,
resultado;
      int k;
      for (k=0; k<500; k++)
      {
      inicio=(double)clock();
      /*  Algorithm   */
      fin=(double)clock();
      resultado=(fin-
      inicio)*1000/
      (double)CLOCKS_PER_SEC;
      }
      }
```

Figure 6 Code used for measuring the execution time task

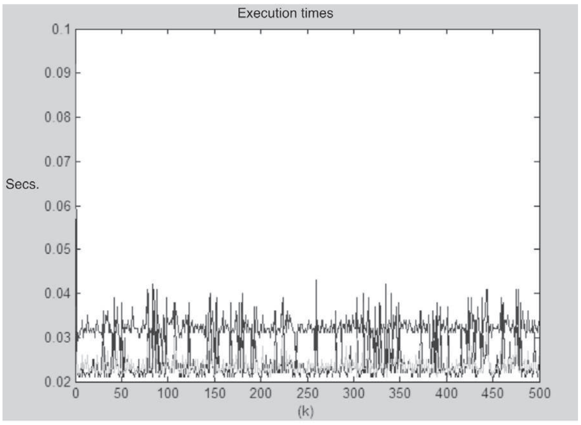


Figure 7 Execution times measuring $C_{1,k}$, $C_{2,k}$, $C_{3,k}$ using *clock() function*

The matrix filter estimation result (25) was based on an instrumental variable applied in (9).

$$\hat{A}_n = \begin{bmatrix} 0.003315 & 0 & 0 \\ 0 & 0.01086 & 0 \\ 0 & 0 & -0.03723 \end{bmatrix} \quad (25)$$

The matrix parameter evolution is shown in figure 8, in addition this matrix presents the three estimated parameters obtained thru instrumental variable technique.

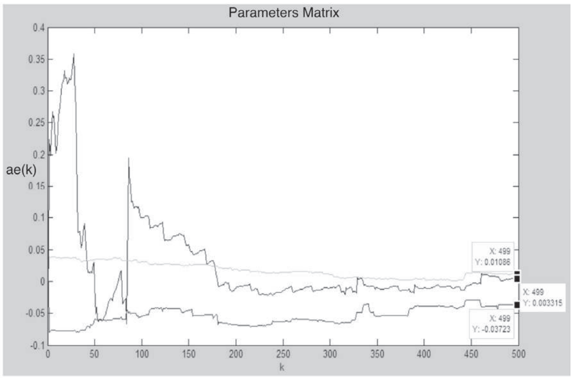


Figure 8 Matrix parameters estimation obtained in 500 instances

The matrix parameters (25) applying in (4) instead of A, transform to the recursive execution time model (4) in an identifier with non-correlated disturbances (26), and the resulting matrix can be seen in figure 9, in which it is propose the recursive way to model the execution times of the real time task presented in this paper.

$$\begin{bmatrix} \hat{C}_{1.k} \\ \hat{C}_{2,k} \\ \hat{C}_{3.k} \end{bmatrix} = \begin{bmatrix} 0.003315 & 0 & 0 \\ 0 & -0.03723 & 0 \\ 0 & 0 & 0.01086 \end{bmatrix} \begin{bmatrix} \hat{C}_{1.k-1} \\ \hat{C}_{2,k-1} \\ \hat{C}_{3.k-1} \end{bmatrix} + \begin{bmatrix} \widetilde{W}_{1.k-1} \\ \widetilde{W}_{2,k-1} \\ \widetilde{W}_{3.k-1} \end{bmatrix} \quad (26)$$

The execution times and its identifications results, in 500 instances are shown in figure 10. This figure in fact is a comparison between the execution times obtained using the code shown in figure 6, and the execution times obtained thru instrumental variable technique.

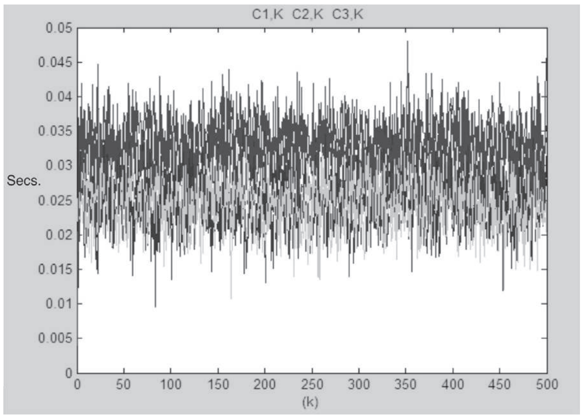


Figure 9 Identification evolution execution times with respect to $C_{1,k}$, $C_{2,k}$, $C_{3,k}$, in 500 instances

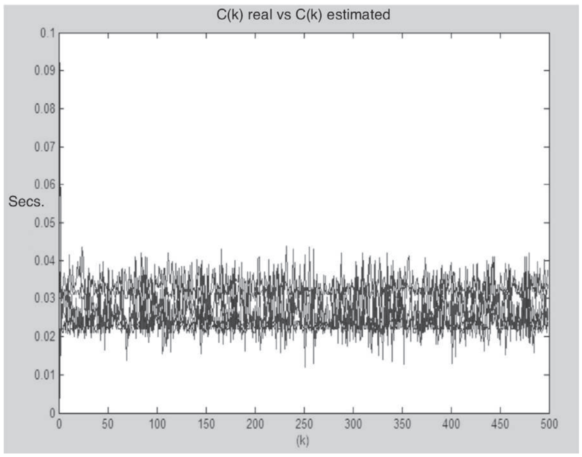


Figure 10 Identification and evolution execution times obtained in 500 instances

Conclusions

Real-time tasks are made up of instances. They have the problem of how to describe their dynamic execution times considering a clock () function. An illustrative simulation model of a DC motor using finite difference described in (22) was proposed, and demonstrated that the system had three concurrent tasks as shown in figure 4. The reconstruction of the execution times was very successful, based on the functional error convergence levels. Execution time of a real-time task in which the internal parameter *A* with respect to (2) is reflected in (3) and explicitly seen in (4) allows considering the estimation process, based on the reconstruction ARMA model achieved good execution times. Additionally this included the functional error concept described by (20). This concept is a comparison between the execution times shown in figure 9 and measured execution times shown in figure 7, this determines that the smaller the functional error, the greater the accuracy of the reconstruction achieved through a proper process of matrix system estimation. Therefore one of the important contributions of this paper is the successful reconstruction of the three scheduled execution time tasks, and the functional error vector obtained is very small so conclusively the smaller is the functional error vector, the better reconstruction it is obtained, as it is demonstrated in figure 11.

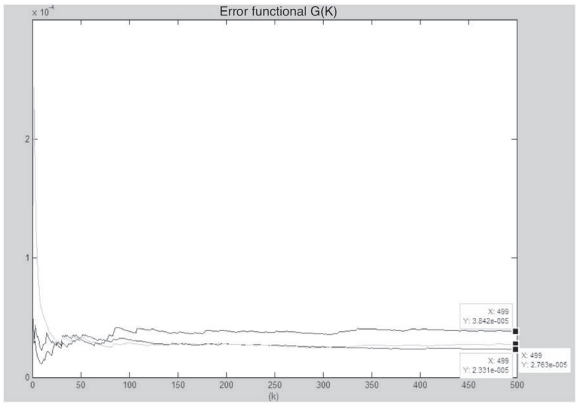


Figure 11 The functional error execution times curves G_k=[$G_{1,k}$, $G_{2,k}$, $G_{3,k}$]

References

1. J. J Medel, P. Guevara, D. Cruz. *Temas Selectos de Sistemas en Tiempo Real*. Ed. Politécnico. México. 2007. pp. 59-70.

2. R. Krten. *QNX Neutrino RTOS Getting Started with QNX Neutrino: A Guide for Realtime Programmers*. QNX Software Systems International Corporation. Ontario. 2008. pp. 23-24.

3. C. Liu, J. Layland. "Scheduling algorithms for multiprogramming in a Hard Real Time System Environment." *Journal of the ACM*. Vol. 20. 1973. pp. 46-61.

4. S. Manolache, P. Eles, Z. Peng. "Schedulability Analysis of Applications with Stochastic Task Execution Times." *ACM Transactions on Embedded Computing Systems*. Vol. 3. 2004. pp. 706-735.

5. F. Stappert, P. Altenbernd. "Complete Worst-Case Execution Time Analysis of Straight-line Hard Real-Time Programs." *Journal of Systems Architecture*. Vol. 46. 2000. pp. 339-355.

6. G. Bernat, A. Colin, S. M. Petters. *PWCET: A tool for probabilistic Worst-Case Execution Time Analysis of Real-Time Systems.* Technical Report YCS-2003-353, Department of Computer Science. University of York. UK. 2003. pp. 2-4.

7. J. J. Medel, P. Guevara, D. Cruz. *Matricial estimation for start times with stochastic behaivor by periodic real time tasks in a concurrent system*. MMACTE'05 Proceedings of the 7th WSEAS International Conference on Mathematical Methods and Computational Techniques In Electrical Engineering. UK. 2005. pp. 254-255.

8. H. Nyquist. "Certain Topics in Telegraph Transmission Theory". *Proceedings of The IEEE.* Vol. 90. February 2002. pp. 285-286

9. J. J. Medel. "Análisis de Dos Métodos de Estimación para Sistemas Lineales Estacionarios e Invariables en el Tiempo con Perturbaciones Correlacionales con el Estado Observable del Tipo: Una Entrada por Salida." *Computación y Sistemas*. Vol. 5. 2002. pp. 216.

10. P. Guevara, G. Delgado, B. del Muro. *Modelado y Validación de un Vector de Tiempos de Ejecución para Tareas en Tiempo Real Concurrentes que Simulan un Motor de Corriente Continua*. VIII Congreso Internacional sobre Innovación y Desarrollo Tecnológico. México. 2010. pp. 3-5.

11. T. Söderström, P. Stoica. "On some system identification techniques for adaptive filtering." *IEEE Trans. Circuits and Systems.* Vol. CS-35. 1988. pp. 457-461.

12. D. B. Stewart. *Measuring Execution Time and Real-Time Performance*. Embedded Systems Conference. Boston. 2006. pp. 2-7.

13. G. Delgado, P. Guevara, J. S. Falcón. *Simulación Concurrente en Tiempo Real de un Motor de Corriente Continua Sobre la Plataforma QNX*. XIV Congreso Latinoamericano de Control Automático, XIX Congreso de la Asociación Chilena de Control Automático ACCA. Chile. 2010. pp. 2-6.