

Kalman Filter Applied to a Digital Computing Process to Find Its Latency

Paulo David Battaglin*, Gilmar Barreto

Department of Semiconductors, Instruments and Photonics, DSIF, School of Electrical and Computing Engineering, FEEC, State University of Campinas, UNICAMP, Campinas, SP, Brazil

Abstract Instantaneous observability is a method used here to watch a system whose internal states vary very fast. It is a system property which allows us to estimate the system internal states. A discrete computing process which is time-varying can be represented by a mathematical model. This model has linear, discrete, stochastic and time-varying equations which contain matrices and vectors whose elements are deterministic functions of discrete time k . This computing process is performed during a period of time called latency. The process is performed in four steps in sequence, the system states: *new process*, *ready-queue*, *CPU running*, *process end*. The system property instantaneous observability depends on the pair of matrices $\{A(k), C(k)\}$ and regards the possibility to estimate the system internal states when the system state equations are known. The problem is the system states are inside and they are not always accessible directly. In this paper we propose a method to determine: the instantaneous observability matrices at discrete time k , the system state estimation and the system latency when the system output measurements are known. We will show when instantaneous observability property comes true, the system instantaneous internal states and latency can be estimated. This is an advantage compared to usual observability method based on static scenarios. The potential application of the results is a prediction of data traffic-jam on a computer process. In a broader perspective, the instantaneous observability method can be applied on identification of pathology, weather forecast, navigation, tracking, stock market and many other areas.

Keywords Digital computing process, Instantaneous observability, Discrete time-varying system, Latency, Kalman filtering

1. Introduction

An end-user participates in a teleconference on a personal computer, which is connected through the Internet to another end-user located at a remote place. This teleconference triggers the creation and execution of many computer processes inside this end-user's personal computer. We will look at a simple computing process which can be represented by a mathematical model that is a system of linear, discrete, stochastic and time-varying equations.

Our objectives are: evaluate the end-user satisfaction who is running an application on a personal computer, so that the end-user has a perception that everything is going on like a real time; determinate the *latency*: when end-user receives fast communications on a personal computer's monitor screen through sound, image and other messages within the

smallest possible time which is the *latency*; apply the instantaneous observability property on the system of linear, discrete, stochastic and time-varying equations; and apply the Kalman filtering on this system.

The current literature issued [1-3] present a ready queue processing time using lottery analysis, an application where the processing time of jobs in ready queue is predicted, and have given their contributions to the field.

The approach we will present here is a method to evaluate computing process latency and the latency of each inner state of it through the Kalman filtering in the time domain. The scope of this article is an extent of perception about what is going on inside a computing process as far as time domain is concerned. Each inner state process was estimated using a state space methodology through a Kalman filter which we have built. We have not seen a work like this and for this reason we think it has significance in relation to the other researches. The notation to be used within Kalman filter in this paper is summarized in Table 1.

* Corresponding author:

paulodav@dsif.fee.unicamp.br (Paulo David Battaglin)

Published online at <http://journal.sapub.org/control>

Copyright © 2014 Scientific & Academic Publishing. All Rights Reserved

Table 1. Notations

$E(k)$	connecting matrix between system noise and system state vector
$r_s(k)$	System noise vector
$F(k)$	matrix that connects measurement noise vector and measurement vector
$r_m(k)$	measurement noise vector
$K_r(k)$	noise covariance matrix
$K_{r_s}(k)$	System noise covariance matrix
$K_{r_m}(k)$	Measurement noise covariance matrix
k_τ	discrete time when event $\hat{x}(k_\tau)$ occurs
$\hat{x}^-(0)$	estimated state vector initial condition at discrete time $k = 0$
$\tilde{x}^-(0)$	initial condition of estimated state vector error at discrete time $k = 0$
$K_x^-(0)$	Initial condition of covariance matrix of state vector at discrete time $k = 0$
$\hat{x}^-(k)$	State estimation vector “a priori” at discrete time k
$K_x^-(k)$	State estimation covariance matrix “a priori” at discrete time k
$G_k(k)$	Kalman filter gain matrix “a priori” at discrete time k
$\hat{x}^-(k+1)$	state estimation vector “a priori” at discrete time $k+1$
$K_x^-(k+1)$	State estimation covariance matrix “a priori” at discrete time $k+1$
$y(k)$	measurement vector obtained at the system output at discrete time k
$In(k)$	innovation vector calculated from $y(k)$ at discrete time k
$G_k(k k)$	Kalman filter gain matrix which was filtered at discrete time k
$\hat{x}^+(k k)$	state estimation vector which was filtered at discrete time k
$K_x^+(k k)$	state estimation covariance matrix which was filtered at discrete time k

2. The Digital Computing Process

Process is a software running program on a digital computer, which can be associated within the operating system internal activities and can be related to end-user programs. A process execution is performed sequentially over time. The process under investigation is performed in a sequence of four steps: new process, ready queue, CPU

running, end of process. *New Process* represents the time of a new process creation T_N and the connection to the next step. *Ready queue* represents the time each process remains in a line till it is dispatched to a CPU for execution. This time is represented by a stochastic variable T_i associated to dispatching priorities and initial waiting time on a CPU. *CPU Running* represents the process running time on a central processing unit, T_{CPU} . *End of Process* represents the time of departure of a process that was executed and its connection to the output and the end-user, T_s . Each of the four steps can be represented by a mathematical model. The block diagram that represents the process under investigation is shown in Figure 1 below.

$u(k)$ and $y(k)$ are the input vector and the output vector respectively as well as the matrix $D(k)$ connects the input to the output for control purposes.

The output $y(k)$ and input $u(k)$ ratio is a function of discrete time k which can be obtained through a discrete time convolution of the four steps in sequence. The z -transform was applied to this function of discrete time k that relates output and input of process as well as we have got a discrete transfer function that depends on z variable as per equation (1):

$$\frac{Y(z)}{U(z)} = \left[\frac{z^{-(T_N + T_s + T_i)}}{(1-z^{-1})^4} - \frac{z^{-(T_N + T_s + T_i)}}{(1-z^{-1})^3(1-e^{-350T}z^{-1})} \right] \quad (1)$$

The values of T_N , T_{CPU} and T_s are set up. Then equation (1) can be decomposed by a method of partial fractions,

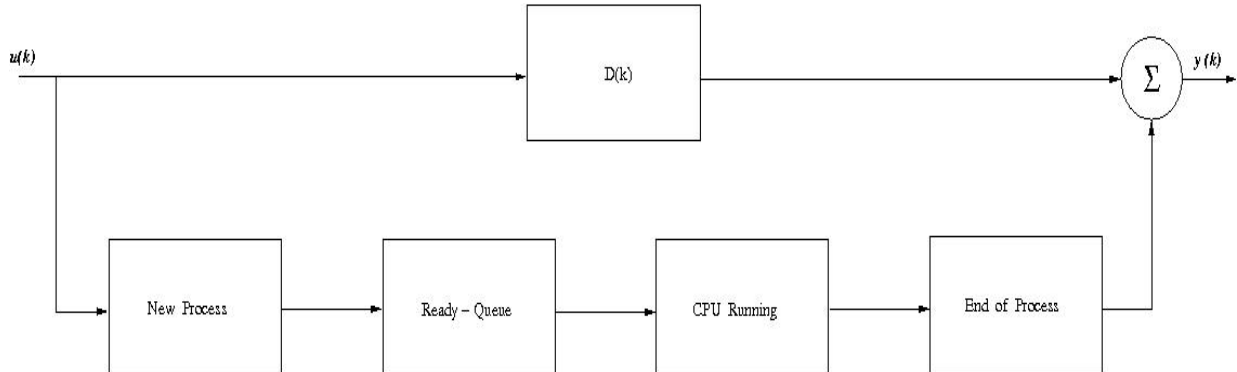
$$\begin{aligned} \frac{Y(z)}{U(z)} &= \frac{z^{-(0,2+T_i)}(1-e^{-350T})z^4}{(z-1)^4(z-e^{-350T})} \\ &\equiv \frac{a_1}{(z-e^{-350T})} + \frac{a_2}{(z-1)} + \frac{a_3}{(z-1)^2} + \frac{a_4}{(z-1)^3} + \frac{a_5}{(z-1)^4} \end{aligned}$$

a_1, a_2, a_3, a_4, a_5 are z -functions to be determined. The partial fractions obtained will allow us to build a block diagram as we can see in Figure 2 below.

Based on the block diagram of figure 2 we can write the state equation (2) and measuring equation (3) with state variables in discrete time domain for a digital computing process as follow [4]:

$$x(k+1) = A(k)x(k) + B(k)u(k) \quad (2)$$

$$y(k) = C(k)x(k) + D(k)u(k) \quad (3)$$

**Figure 1.** The block diagram illustration of a digital computing process with four steps in sequence

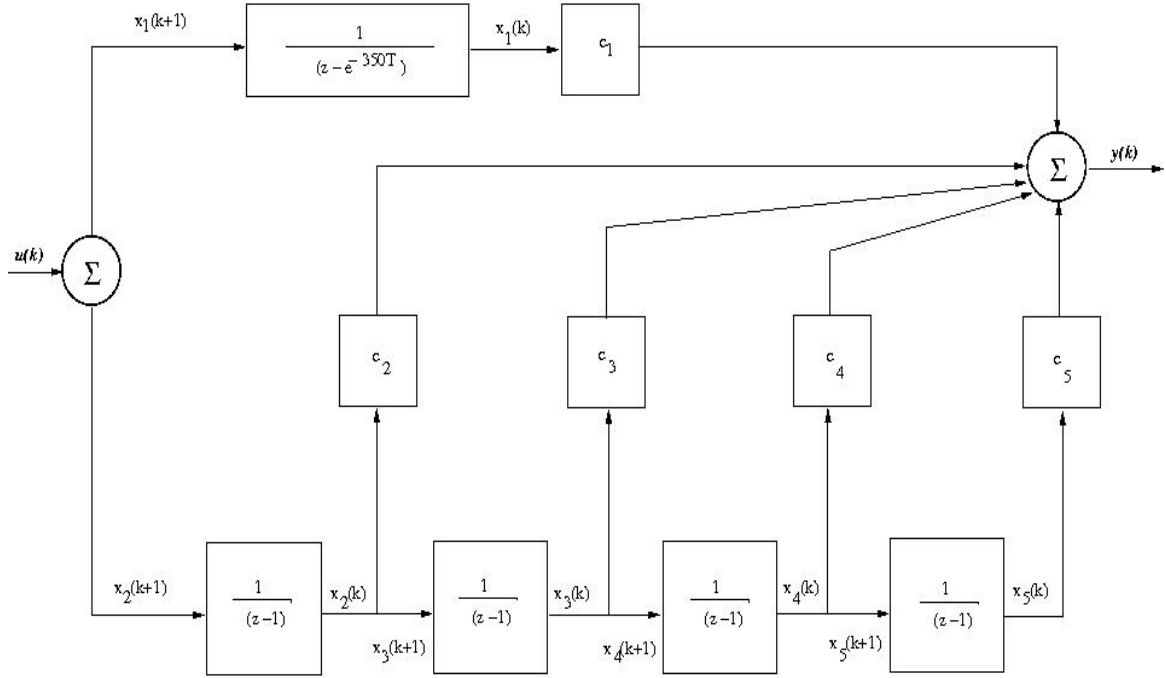


Figure 2. State variables block diagram of the digital process transfer function

The vectors $\mathbf{u}(k) \in \mathbb{R}^p$, $\mathbf{x}(k) \in \mathbb{R}^n$, $\mathbf{y}(k) \in \mathbb{R}^q$ are respectively the system input, the system state and the system output. The matrices $\mathbf{A}(k) \in \mathbb{R}^{n \times n}$, $\mathbf{B}(k) \in \mathbb{R}^{n \times p}$, $\mathbf{C}(k) \in \mathbb{R}^{q \times n}$ e $\mathbf{D}(k) \in \mathbb{R}^{n \times p}$ are respectively the dynamic system matrix, input coupling matrix, measurement sensitivity matrix and input-output coupling matrix which can be seen in reference [9].

$$\mathbf{A}(k) = \begin{bmatrix} e^{-350kT} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \text{ so that } \mathbf{A}(k) \in \mathbb{R}^{5 \times 5}$$

$$\mathbf{B}(k) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ so that } \mathbf{B}(k) \in \mathbb{R}^{5 \times 1}$$

$$\mathbf{C}(k) = (1 - e^{-350T}) \cdot \begin{bmatrix} \Delta[k - (T_N + T_s + T_i)] \\ 4\Delta[k - (T_N + T_s + T_i)] \\ 6\Delta[k - (T_N + T_s + T_i)] \\ 4\Delta[k - (T_N + T_s + T_i)] \\ \Delta[k - (T_N + T_s + T_i)] \end{bmatrix}^T$$

so that $\mathbf{C}(k) \in \mathbb{R}^{1 \times 5}$

$$\Delta[k - (T_N + T_s + T_i)] = \begin{cases} 1, & \text{for } k = (T_N + T_s + T_i) \\ 0, & \text{for other } k \end{cases}$$

Δ is Kronecker's delta function for discrete signals.

$\mathbf{D}(k) = 1$, so that $\mathbf{D}(k) \in \mathbb{R}$

Note that $\mathbf{A}(k)$ and $\mathbf{C}(k)$ are time-varying matrices and $\mathbf{C}(k)$ has a stochastic variable T_i . The CPU running has two states: waiting for a process and running a process. These estates added to the other three states make $\mathbf{A}(k) \in \mathbb{R}^{5 \times 5}$. The system input vector $\mathbf{u}(k)$ is an impulse train with constant

amplitude of 0.07 second over each discrete time interval as well as the solution of Eq. (2) and (3) is the state vector $\mathbf{x}(k)$. The system latency is the summation of vector $\mathbf{x}(k)$ elements.

3. Instantaneous Observability Method

The determination of the instantaneous observability for a linear, discrete and time-varying system depends on time interval over which the instantaneous observability matrix is determined. The system is represented by equations (2) and (3) shown before.

The algebraic point of view of instantaneous observability condition it may be presented such as: a linear, discrete, time-varying system is observable, if and only if, its instantaneous observability matrix $\mathbf{O}(k)$ has a rank equals to n at each instant of time k , so that n is system state vector dimension [5].

The matrices $\mathbf{A}(k)$ and $\mathbf{C}(k)$ are time-varying and discrete in time k , for $k = 0, 1, 2, \dots, (n-1)$. Then the pair of n -dimension matrices $\{\mathbf{A}(k), \mathbf{C}(k)\}$ is observable at instant k_0 if there exists a finite time $k > k_0$, so that,

$$\text{rank} [\mathbf{O}(k)] = \text{rank} \begin{bmatrix} \mathbf{O}_0(k) \\ \mathbf{O}_1(k) \\ \mathbf{O}_2(k) \\ \vdots \\ \mathbf{O}_{n-1}(k) \end{bmatrix} = n \quad (4)$$

$$\mathbf{O}_{m+1}(k) = \mathbf{O}_m(k)\mathbf{A}(k) + \mathbf{O}_m(k) - \mathbf{O}_m(k-1) \quad (5)$$

$$\mathbf{O}_0(k) = \mathbf{C}(k) \quad (6)$$

[6] m is the number of row-block matrices so that we have $m = 0, 1, 2, \dots, n-1$.

The equation (4) is the general shape of the instantaneous observability matrix concerning a linear, discrete and time-varying system. The equation (5) is the generating

function of each instantaneous observability matrix block $\mathbf{O}_{m+1}(k)$. The equation (6) is the initial block $\mathbf{O}_0(k)$ of the instantaneous observability matrix. m is the index of block generating function for instantaneous observability matrix, which runs from value zero up to the system order minus one. The instantaneous observability matrix generation is focused on the hypothesis that the linear, discrete and time-varying system is instantaneous observable at the instant k , and there exist times $k = k_1, k_2, \dots, k_{n-1}$, so that $k > k_0$ to have valid the equation (4).

From equations (5) and (6) we obtain the instantaneous observability matrix blocks to build equation (4) as follow:

$$\mathbf{O}_0(k) = \mathbf{C}(k) \quad (6)$$

$$\mathbf{O}_1(k) = [\mathbf{C}(k)\mathbf{A}(k) + \mathbf{C}(k) - \mathbf{C}(k-1)] \quad (7)$$

$$\mathbf{O}_2(k) = \mathbf{C}(k)\mathbf{A}^2(k) + \mathbf{C}(k) \cdot [2\mathbf{A}(k) + \mathbf{I}] - \mathbf{C}(k-1) \cdot [\mathbf{A}(k) + \mathbf{A}(k-1) + 2\mathbf{I}] + \mathbf{C}(k-2) \quad (8)$$

⋮

$$\mathbf{O}_{n-1}(k) = [\mathbf{O}_{n-2}(k)\mathbf{A}(k) + \mathbf{O}_{n-2}(k) - \mathbf{O}_{n-2}(k-1)] \quad (9)$$

The rank of each stacked matrix in equation (4) is defined as the number of linearly independent column vectors. This is equals to the number of linearly independent row vectors too. Therefore, each matrix block in equation (4) has the dimension $q \times n$, then equation (4) rank is

$$\text{rank} \leq \min(nq, n)$$

The global dimension of these stacked matrices is $n \cdot q \times n$, n is the number of linearly independent columns and it is also the smallest dimension, and then equation (4) rank is n . Then we conclude the above equation (4) belongs to $\mathbb{R}^{n \cdot q \times n}$ [7].

Consequently, we can build an instantaneous observability matrix set over time shown by equation (10),

$$\mathbf{O}(k) = \{\mathbf{O}(k_1), \mathbf{O}(k_2), \mathbf{O}(k_3), \dots, \mathbf{O}(k_\infty)\} \quad (10)$$

This set we call *Set of Instantaneous Observability Matrices of a Linear, Discrete and Time-Varying System*. Instantaneous Observability takes into account the possibility to estimate the system instantaneous state vector $\mathbf{x}(k)$ based on output measurements, assuming that the state equations (2) and (3) are known.

The rank condition for an instantaneous observability matrix of a linear, discrete and time-varying system can be presented in a manner equivalent to the condition of being a positive definite matrix. The product of a transposed instantaneous observability matrix by its instantaneous observability matrix is shown in equation (11) as follows

$$[\mathbf{O}(k)]^T \cdot \mathbf{O}(k) = \mathbf{G}_O(k) \quad (11)$$

and $\mathbf{G}_O(k) \in \mathbb{R}^{n \times n}$. The result of this product is a square matrix called *Instantaneous Observability Gramian Matrix* $\mathbf{G}_O(k)$ of a linear, discrete and time-varying system. The elements of this matrix are n -dimension vectors result of inner products which vary in time. When the determinant of $\mathbf{G}_O(k)$ is different of zero, it means this matrix is nonsingular and it is invertible. Hence, there exists a matrix $\mathbf{G}_O(k)^{-1}$ so that

$$\mathbf{G}_O(k)^{-1} \cdot \mathbf{G}_O(k) = \mathbf{I}$$

The column vectors of an instantaneous observability Gramian matrix for a given system are linearly independent. Consequently these vectors are mutually orthogonal and they build a base of dimension \mathbb{R}^n related to Euclidean inner product. This Gramian is a qualitative algebraic characterization of the solution uniqueness. The output vector $\mathbf{y}(k)$ has elements which are available and system state equation solution is possible and unique, and the system state solution is the state vector $\mathbf{x}(k)$ [8].

Determinant of $\mathbf{O}(k)$ was computed based on equation (10) and Figure 3 has the results over discrete time k .

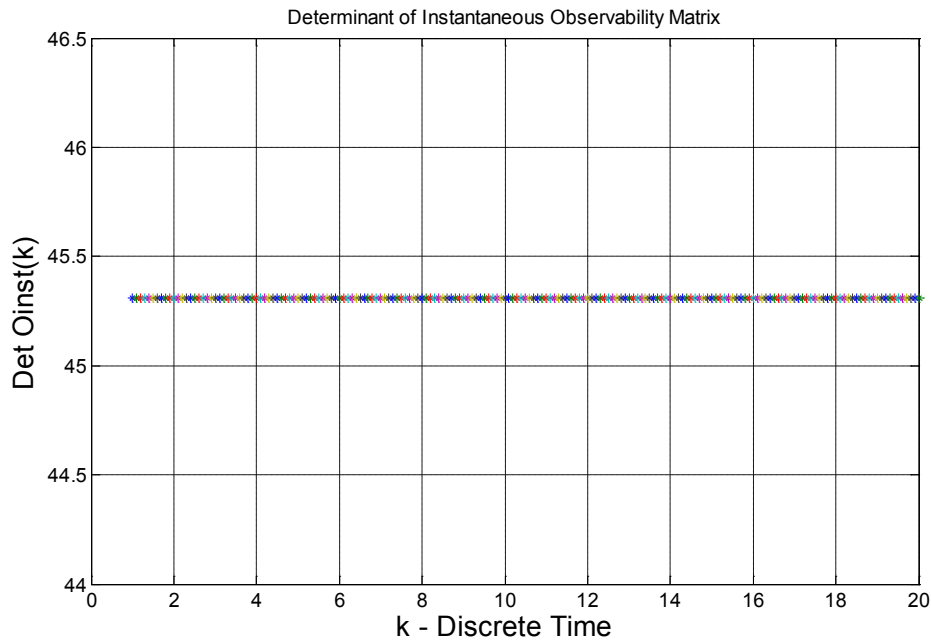


Figure 3. Determinant of Instantaneous Observability Matrix

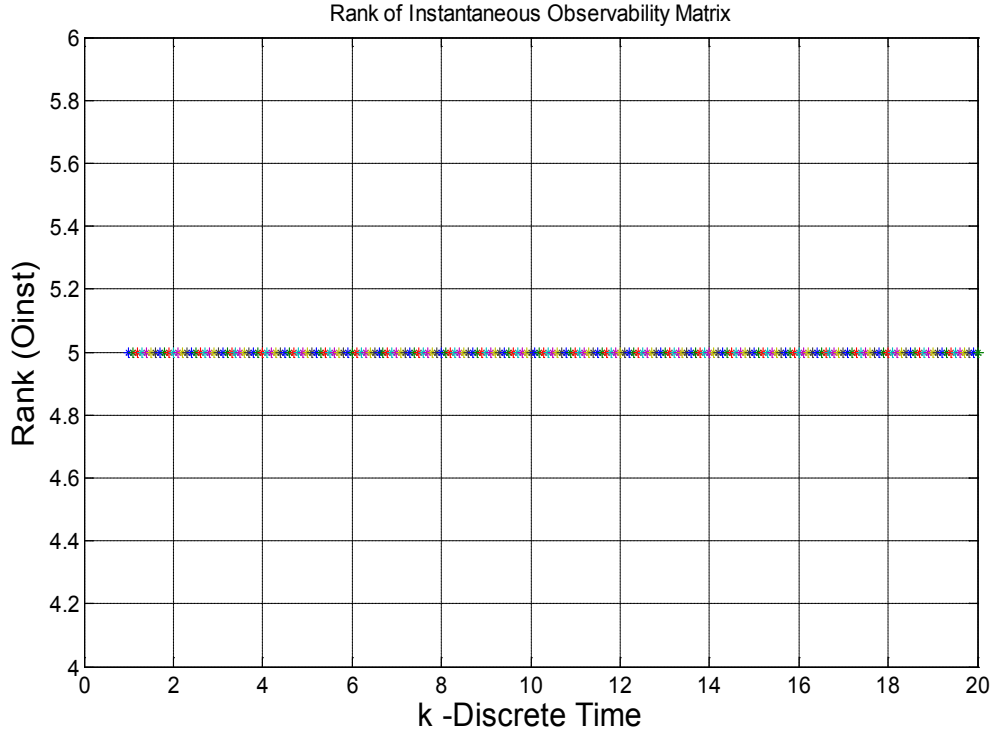


Figure 4. Rank of Instantaneous Observability Matrix

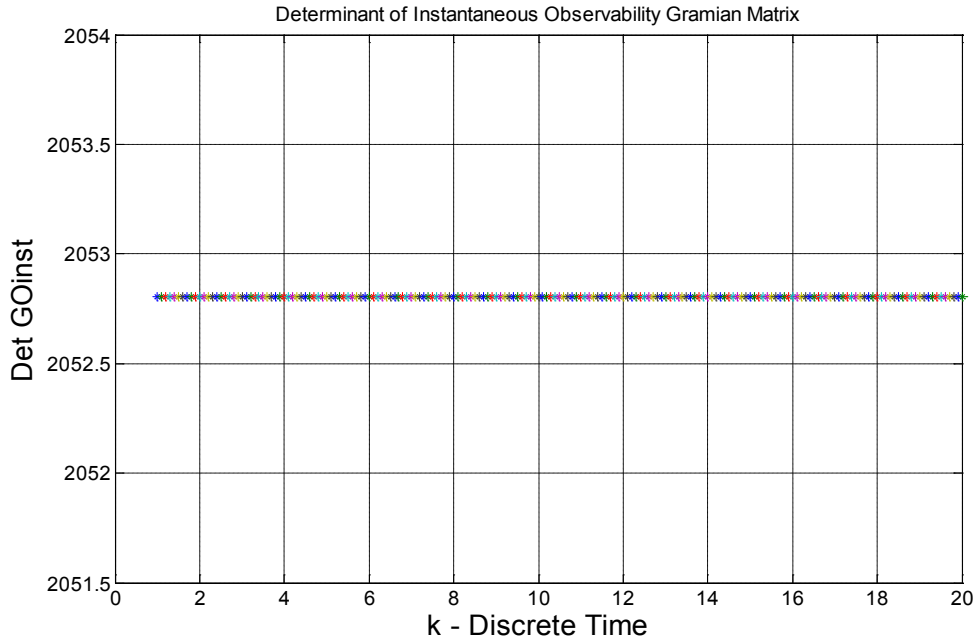


Figure 5. Determinant of Instantaneous Observability Gramian Matrix

From Figure 3 below we note the determinant is positive, constant and equals to 45.3079 over discrete time k , concerning the system the equations (2) and (3). It means all column vectors of the instantaneous observability matrix $\mathcal{O}(k)$ are linearly independents.

The rank of instantaneous observability matrix was computed and Figure 4 below has the result over discrete time k .

From Figure 4 we note the rank of instantaneous

observability matrix is equals to five. It shows the system has order five and it is instantaneously full observable.

From Figure 5 below we note the instantaneous observability gramian matrix determinant is positive, constant and equals to 2052.83 over discrete time k . It shows all column vectors of the instantaneous observability gramian matrix $G_o(k)$ are linearly independents and there exists a unique solution for equations (2) and (3) which is the state vector $\mathbf{x}(k)$ and its covariance matrix $\mathbf{K}_X(k)$. Then we

have seen the system is instantaneously observable and it means the system internal states can be estimated. The Kalman filter will make this estimation.

4. Kalman Filter Applied to a Digital Computing Process

In preparation for applying the Kalman filter to the system equations that represents the digital computing process defined in section 2, we will arrange the Kalman filter equations with the appropriate notation, so as to facilitate a visualization for filtering algorithm. We will use the concept "a priori" which refers to values obtained before the measurement, as well as the concept "a posteriori" that refers to that after obtaining the values measured at the system output.

The vectors and matrices with values "a priori" will be represented by a superscript minus "-" as $\hat{\mathbf{x}}^-(k)$, $\hat{\mathbf{x}}^-(k+1)$, $\mathbf{K}_x^-(k)$ e $\mathbf{K}_x^-(k+1)$. The vectors and matrices with values "a posteriori" will be represented by a superscript plus "+" as $\hat{\mathbf{x}}^+(k|k)$ and $\mathbf{K}_x^+(k|k)$.

Filtering is the processing of $\mathbf{y}(k)$ measurement obtained at the system output to improve or refine the estimates of the state $\hat{\mathbf{x}}^-(k)$ and $\mathbf{K}_x^-(k)$ at time k . After this, we will obtain the filtered estimates $\hat{\mathbf{x}}^+(k|k)$ and $\mathbf{K}_x^+(k|k)$ which will propagate. Similarly to the measurement $\mathbf{y}(k+1)$, to enhance or refine the estimates of the state $\hat{\mathbf{x}}^-(k+1)$, $\mathbf{K}_x^-(k+1)$ at time $k+1$. After this, we obtain the filtered estimates $\hat{\mathbf{x}}^+(k+1|k+1)$, $\mathbf{K}_x^+(k+1|k+1)$ which will propagate.

The matrix equations (12), (13) and (14) below represent the mathematical model of the linear, discrete, stochastic and time-varying system; the system Gaussian white noise component $\mathbf{r}_s(k)$ was added on equation (12), the measurement Gaussian white noise component $\mathbf{r}_m(k)$ was added on equation (13) and the noise covariance matrix $\mathbf{K}_r(k)$ is shown by equation (14). These noises are not correlated. Matrix equations (15) to (24) represent the Kalman filter and its initial conditions concerning the values "a priori" and "a posteriori" for each discrete time interval k .

- *Mathematical model of the linear, discrete, stochastic and time-varying system*

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{E}(k)\mathbf{r}_s(k) \quad (12)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) + \mathbf{F}(k)\mathbf{r}_m(k) \quad (13)$$

$$\mathbf{K}_r(k) = \begin{bmatrix} \mathbf{K}_{r_s}(k) & 0 \\ 0 & \mathbf{K}_{r_m}(k) \end{bmatrix} \cdot \Delta(k - k_\tau) \quad (14)$$

- *Kalman filter initialization with initial conditions*

$$\hat{\mathbf{x}}^-(0) = \mathbf{E}[\mathbf{x}^-(0)] = \mathbf{m}_x^-(0)$$

$$\tilde{\mathbf{x}}^-(0) = \hat{\mathbf{x}}^-(0) - \mathbf{x}^-(0)$$

$$\mathbf{K}_x^-(0) = \mathbf{E}[\tilde{\mathbf{x}}^-(0)(\tilde{\mathbf{x}}^-(0))^T] \quad (15)$$

- *Kalman filter Gain before measurement*

$$\mathbf{G}_k(k) = [\mathbf{A}(k)\mathbf{K}_x^-(k)\mathbf{C}^T(k)][\mathbf{C}(k)\mathbf{K}_x^-(k)\mathbf{C}^T(k) + \mathbf{K}_{R_s}(k)]^{-1} \quad (16)$$

- *Processing before the measurement-propagation*

$$\hat{\mathbf{x}}^-(k+1) = \mathbf{A}(k)\hat{\mathbf{x}}^-(k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{E}(k)\mathbf{r}_s(k) \quad (17)$$

$$\begin{aligned} \mathbf{K}_x^-(k+1) &= \mathbf{A}(k)\mathbf{K}_x^-(k)\mathbf{A}^T(k) - \mathbf{G}_k(k)\mathbf{C}(k)\mathbf{K}_x^-(k)\mathbf{A}^T(k) \\ &\quad + \mathbf{G}_k(k)\mathbf{C}(k)\mathbf{K}_x^-(k)\mathbf{C}^T(k)\mathbf{G}_k^T(k) \\ &\quad - \mathbf{A}(k)\mathbf{K}_x^-(k)\mathbf{C}^T(k)\mathbf{G}_k^T(k) + \mathbf{K}_{R_s}(k) \end{aligned} \quad (18)$$

- *Getting the measurement $\mathbf{y}(k)$ at the system output and calculate the innovation*

$$\mathbf{In}(k) \quad (19)$$

- *Kalman filter Gain after measurement: filtered Gain*

$$\mathbf{G}_k(k|k) = \mathbf{K}_x^-(k)\mathbf{C}^T(k)[\mathbf{C}(k)\mathbf{K}_x^-(k)\mathbf{C}^T(k) + \mathbf{K}_{R_m}(k)]^{-1} \quad (20)$$

- *State update to the measurement: filtered state*

$$\begin{aligned} \hat{\mathbf{x}}^+(k|k) &= \hat{\mathbf{x}}^-(k) + \mathbf{G}_k(k|k)[\mathbf{y}(k) \\ &\quad - \mathbf{C}(k)\hat{\mathbf{x}}^-(k) - \mathbf{D}(k)\mathbf{u}(k)] \end{aligned} \quad (21)$$

$$\mathbf{K}_x^+(k|k) = \mathbf{K}_x^-(k)$$

$$- \mathbf{K}_x^-(k)\mathbf{C}^T(k)[\mathbf{C}(k)\mathbf{K}_x^-(k)\mathbf{C}^T(k) + \mathbf{K}_{R_m}(k)]^{-1}\mathbf{C}(k)\mathbf{K}_x^-(k) \quad (22)$$

- *Propagation of state until the following interval k*

$$\hat{\mathbf{x}}^-(k+1) = \mathbf{A}(k)\hat{\mathbf{x}}^+(k|k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{G}_k(k|k)\mathbf{In}(k) \quad (23)$$

$$\begin{aligned} \mathbf{K}_x^-(k+1) &= \mathbf{A}(k)\mathbf{K}_x^-(k|k)\mathbf{A}^T(k) - \\ &\quad - \mathbf{G}_k(k|k)\mathbf{C}(k)\mathbf{K}_x^-(k|k)\mathbf{A}^T(k) + \\ &\quad + \mathbf{G}_k(k|k)\mathbf{C}(k)\mathbf{K}_x^-(k|k)\mathbf{C}^T(k)\mathbf{G}_k^T(k|k) + \\ &\quad + \mathbf{G}_k(k|k)\mathbf{K}_{R_m}(k)\mathbf{G}_k^T(k|k) + \\ &\quad - \mathbf{A}(k)\mathbf{K}_x^-(k|k)\mathbf{C}^T(k)\mathbf{G}_k^T(k|k) + \mathbf{K}_{R_s}(k) \end{aligned} \quad (24)$$

Test: Was the last measure processed?

- *If not, go to the equation (19)*
- *If so, finish processing the Kalman filter algorithm for digital computing process [9].*

5. The Latency of a Digital Computing Process

When an end-user submits an application and sometime later receives fast communications on a personal computer's monitor screen through sound, image and other messages within the smallest possible time, this time is the *latency*. The utility of latency is to measure the satisfaction of this end-user who is using an application on a computer, so that the end-user has the perception that everything is happening in real time; when the personal computer receives rapid communication through sound, image and other messages like "a bar of sound intensity" with the lowest latency.

In order to determine the latency of our digital computer process, we have set up the initial conditions to run the algorithm called Kalman filter; system noise and measurement noise are generated as Gaussian white noises, the Kalman filter initial conditions as per equation (15) are given, measurements at system output are available, matrices $\mathbf{A}(k)$, $\mathbf{B}(k)$, $\mathbf{C}(k)$, $\mathbf{D}(k)$, $\mathbf{E}(k)$, $\mathbf{F}(k)$ are known. The system input vector will be $\mathbf{u}(k)$ which is a train of pulses of two bytes.

The four following figures we will see bellow illustrate the

“*a posteriori*” behavior for filter gain, state vector, error of estimation covariance matrix and latency given by the Kalman filter, after the end-user has submitted the application twenty times. The figures have dashed lines which represent discrete values. The numbers to be presented have four decimals.

Next we will see the Figure 6 with the five elements of the Filter Gain Matrix “*a posteriori*” obtained from the matrix equation (20). In these elements the errors on the Gain “*a priori*” were filtered and the filtering is based on the measurements read at the system output, under the condition of minimum error variance.

This figure shows the five Kalman Filter Gain Matrix’s elements which have the following values: $gkf11(k) = 0,0050$; $gkf21(k) = 0,0208$; $gkf31(k) = 0,0712$; $gkf41(k) = 0,1102$; $gkf51(k) = 0,0433$.

These filtered values mean the error covariance matrix has its elements minimized. In other words, the system state estimations have variances which were minimized by the Kalman filtering. The Gain matrix with filtered values can be written as,

$$\mathbf{G}_k(k|k) = \begin{bmatrix} gkf11(k) \\ gkf21(k) \\ gkf31(k) \\ gkf41(k) \\ gkf51(k) \end{bmatrix}$$

Next we see Figure 7 with the estimated state vector’s five elements “*a posteriori*” which were obtained from the matrix equation (21). In these elements the estimation errors were filtered based on the measurements obtained at the system output. In this figure the estimated state vector elements have the following values: $xestMf11(k) = 0.0247$ second; $xestMf21(k) = 0.0517$ second; $xestMf31(k) = 0.1168$ second; $xestMf41(k) = 0.1361$ second; $xestMf51(k) = 0.0894$ second. The gain matrix is influenced by innovations too and so this matrix and vector of innovations act on the values “*a priori*” of the state vector, turning them into “*a posteriori*” or filtered values.

The estimated state vector with filtered five elements can be written as,

$$\hat{\mathbf{x}}^+(k|k) = \begin{bmatrix} xestMf11(k) \\ xestMf21(k) \\ xestMf31(k) \\ xestMf41(k) \\ xestMf51(k) \end{bmatrix}$$

The vertical scales of Fig. 7, 8 and 9 are not the same, in order to facilitate viewing and comparing the amplitudes of mean vector elements, the covariance matrix elements and latency.

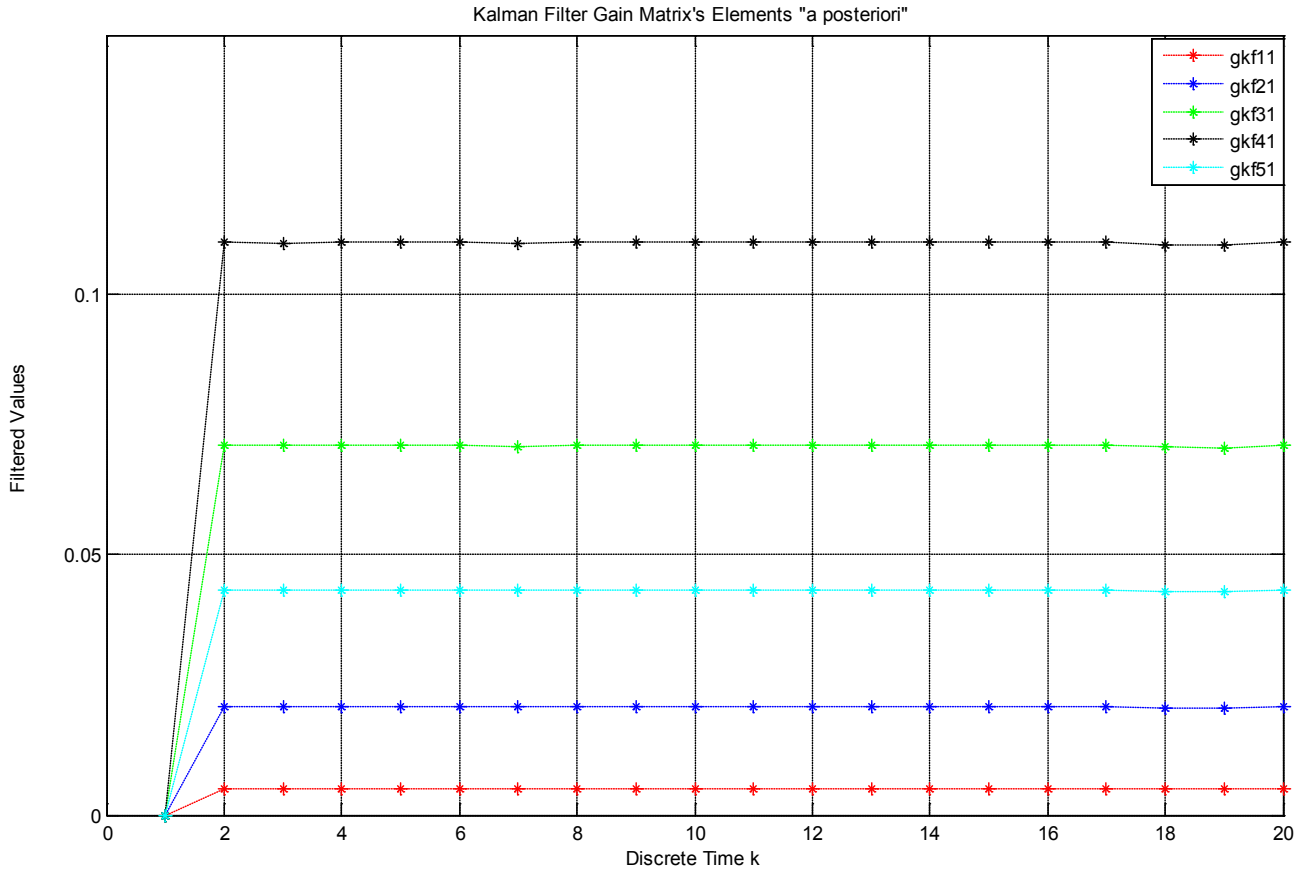


Figure 6. Kalman Filter Gain Matrix’s five elements which were filtered

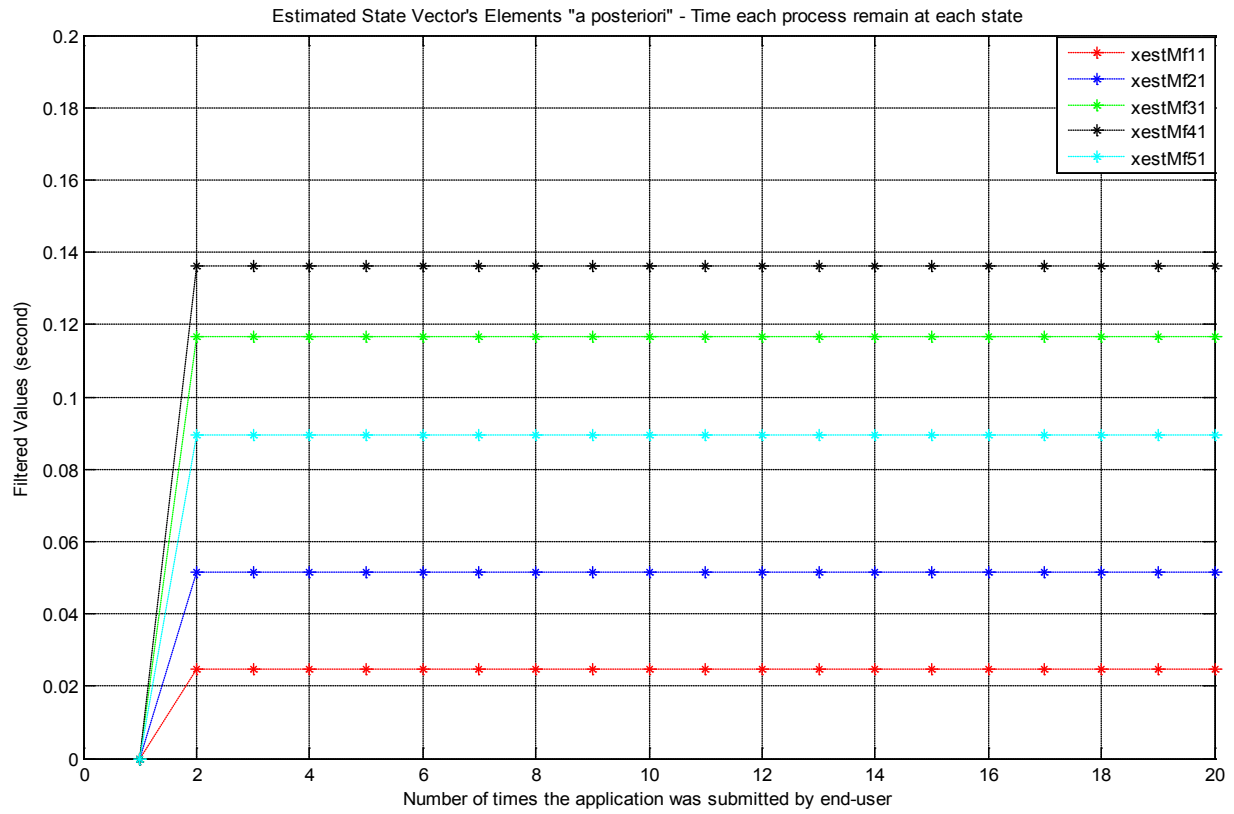


Figure 7. Five elements of the estimated State Vector which were filtered

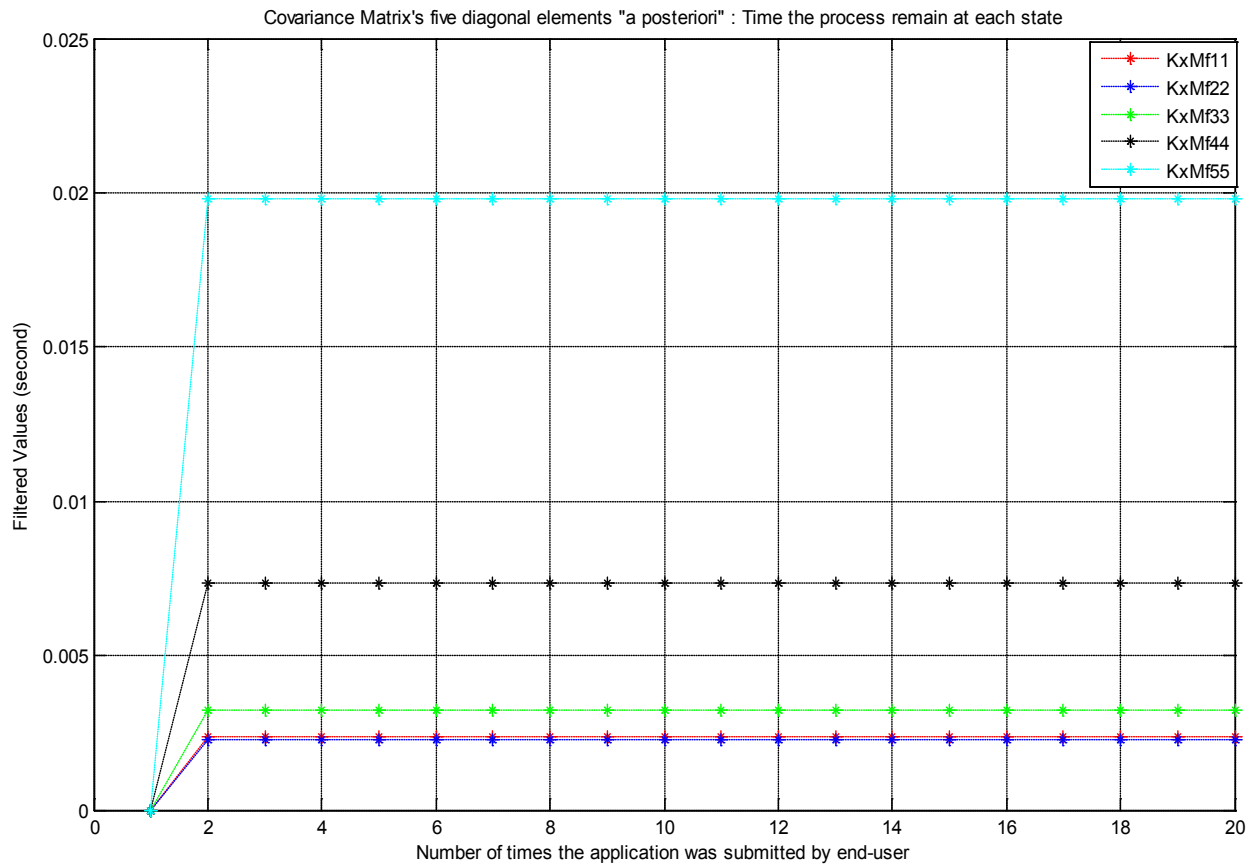


Figure 8. Covariance Matrix's five elements which were filtered

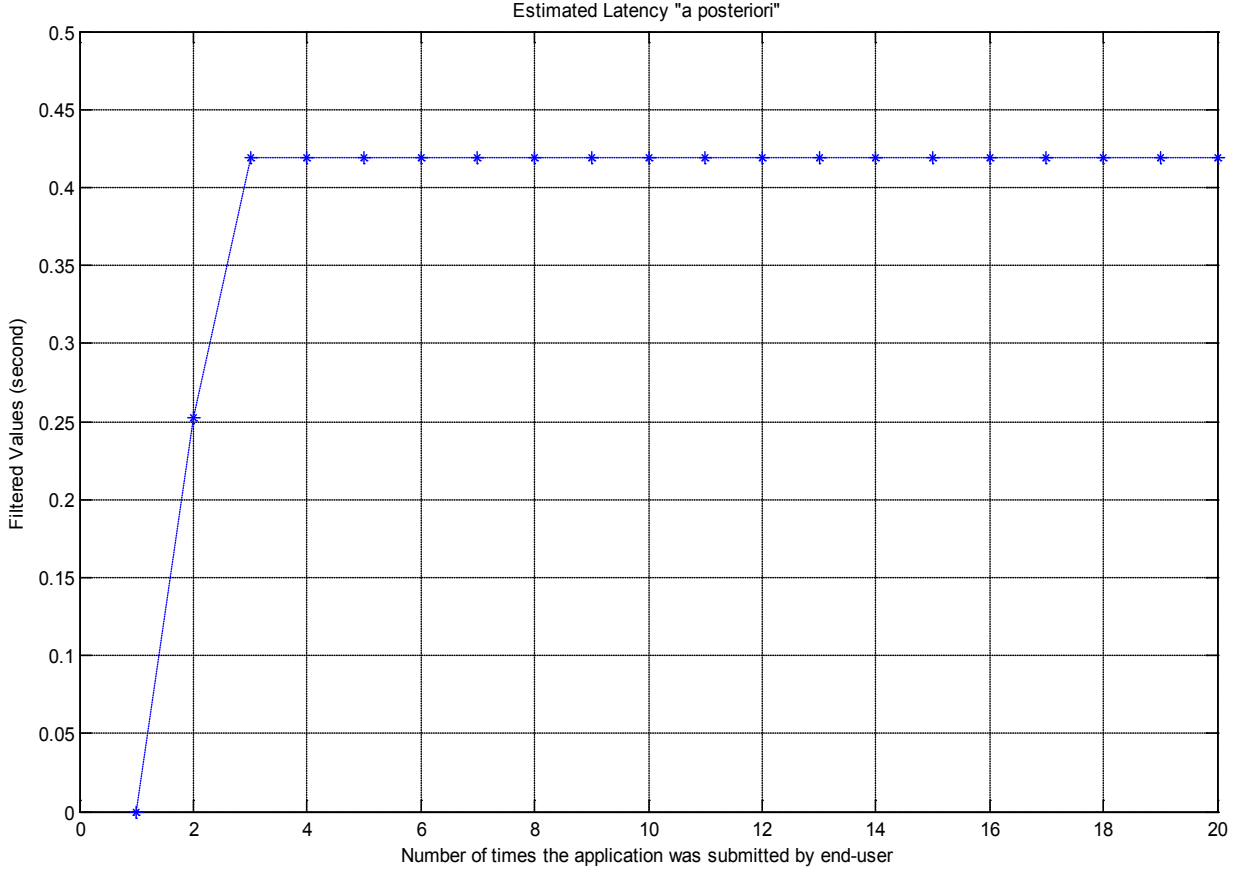


Figure 9. Latency of a digital computing process under investigation

Next we see Fig. 8 with the five elements of the covariance matrix "*a posteriori*", which were obtained from the matrix equation (22). In these elements the errors of state estimation were filtered based on the measurements obtained at the system output. The elements of the covariance matrix have the following values: $K_{xMf11}(k) = 0.0024$ second; $K_{xMf22}(k) = 0.0023$ second; $K_{xMf33}(k) = 0.0032$ second; $K_{xMf44}(k) = 0.0074$ second; $K_{xMf55}(k) = 0.0198$ second. Moreover, as the covariance matrix is time-varying, the time average of the trace of covariance matrix "*a priori*" equals to 0.0513 second, while the time average of the trace of the covariance matrix "*a posteriori*" equals to 0.0351 second. In other words, the elements of the covariance matrix converge over the interval

["*a priori*" \rightarrow "*a posteriori*"]. This makes sense because the more measurements are obtained at the system output over the discrete time k , the better the system state estimation.

The covariance matrix $K_x^+(k|k)$ with five filtered diagonal elements can be written as

$$K_x^+ = \begin{bmatrix} K_{xMf11}(k) & 0 & 0 & 0 & 0 \\ 0 & K_{xMf22}(k) & 0 & 0 & 0 \\ 0 & 0 & K_{xMf33}(k) & 0 & 0 \\ 0 & 0 & 0 & K_{xMf44}(k) & 0 \\ 0 & 0 & 0 & 0 & K_{xMf55}(k) \end{bmatrix}$$

The Figure 9 above depicts the latency which is the sum of estimated and filtered five elements of the state vector $x(k)$.

The sum is based on *sums of averages theorem* [10]. Then this digital computer process latency is 0.4186 second. It means the end-user submits an application and 0.4186 second later receives the result on the personal computer's monitor screen. If the same application is submitted twenty times, then each of the twenty latencies will be 0.4186 second. The application was submitted twenty times to check the mathematical model and the repeatability of the results.

6. Conclusions

We have seen the system is instantaneously observable at output and it means: the measurements could be obtained at the system output instantaneously and the system internal states could be estimated by Kalman filter instantaneously. The input vector $u(k)$ was given and the measurements at system output could be determined.

The Kalman filter equations were built according to state space approach as well as it was applied to the system of linear, discrete, stochastic and time-varying equations which represents the digital computing process. The filter Gain matrix behavior, the estimation of the state vector and the error of estimation covariance matrix along discrete time k were determined through Kalman's algorithm. We have concluded it is possible to apply the Kalman filter to a digital computing process and estimate its latency on an instantaneous basis.

The latency was determined by the sum of the five elements of the estimated and filtered state vector, based on the *sums of averages theorem*.

The end-user expectation about latency will determine the level of satisfaction. If the expectation is not set up by an end-user, then any result can guarantee a satisfaction. If the end-user expectation is less than one second, then the end-user's satisfaction will be assured here. If the end-user expectation is latency less than 0.4186 second, then the end-user will not be happy. Anyway, the end-user satisfaction level depends on the expectation.

The potential application of the results is a prediction of data traffic-jam on computer processes. In a broader perspective, the instantaneous observability method within Kalman filter can be applied on other digital computer processes and other variables than latency can be observed. It is also possible to apply this method to a computer with several CPUs in parallel and an I/O processor, to a several computers like this one connected to a local area network and it connected to a major server, to several major servers connected to a mainframe supercomputer type. Beyond that it can be applied on identification of a pathology, weather forecast, navigation and tracking on ground, water, air and space areas; and stock market prediction.

ACKNOWLEDGEMENTS

We thanks CNPq – National Council of Scientific and Technological Development – in Brazil for financial support through the doctoral scholarship granted, which made this research possible at DSIF/FEEC/UNICAMP.

REFERENCES

- [1] Shukla, Diwakar; Jain, Anjali; Choudhary, Amita. *Prediction of ready queue processing time in a multiprocessor environment using Lottery Scheduling*, Journal of Applied Computer Science & Mathematics, Number 11, Volume 5, Suceava, Romania, 2011.
- [2] Shukla, Diwakar; Jain, Anjali; Choudhary, Amita. *Estimation of ready queue processing time under SL scheduling scheme in multiprocessor environment*, International Journal of Computer Science and Security, Number 4, Volume 1, pp.74-81, 2010.
- [3] Shukla, Diwakar; Jain, Anjali; Choudhary, Amita. *Estimation of ready queue processing time under Group Lottery Scheduling (GLS) in multiprocessor environment*, International Journal of Computer and Applications, Number 14, Volume 8, pp.39-45. 2010.
- [4] Cadzow, James A.; Martens, Hinrich R., *Discrete-time and computer control systems*, Prentice Hall, New Jersey, USA, 2011.
- [5] Kalman, Rudolf Emil; *Lectures on Controllability and Observability* - Reprint, Centro Internazionale Matematico Estivo - Firenze - Italy, Springer-Verlag, Berlin Heidelberg, Germany, 2010.
- [6] Chen, Chi-Tsong; *Linear System Theory and Design*, 3rd. Edition, Oxford University Press, New York, NY, USA, 2009.
- [7] Battaglin, Paulo David; Barreto, Gilmar; *Observability of State Space Model in Linear Discrete Time-Varying Systems*, Congresso Latino Americano de Control Automatico – IFAC, Lima, Peru, 2012.
- [8] Fairman, Frederick Walker; *Linear Control Theory –The State Space Approach*, John Wiley & Sons, New York, USA, 2000.
- [9] Battaglin, Paulo David, *Kalman Filtering applied to a Digital Computing Process with a Time-Varying State Space Approach* – PhD thesis, DSIF/FEEC, University of Campinas, SP, Brazil, 2014.
- [10] Lemons, Don S. (2000). *An Introduction to stochastic process in Physics*, The John Hopkins University Press, Baltimore, MD, USA, 2002.