Revista Iberoamericana para la Investigación y el Desarrollo Educativo
ISSN 2007 - 7467

https://doi.org/10.23913/ride.v13i26.1510

*Artículos científicos*

# Determinación Experimental del Tiempo de Muestreo en Hardware con Microcontroladores y su Relación con las Matemáticas en t, s y z

## *Experimental Determination of Sampling Time in Hardware with Microcontrollers and Its Relationship to Mathematics in t, s and z*

## *Determinação Experimental do Tempo de Amostragem em Hardware com Microcontroladores e sua Relação com a Matemática em t, s e z*

**José Rivera-Mejia**
Tecnológico Nacional de México, Instituto Tecnológico de Chihuahua, México
jose.rm@chihuahua.tecnm.mx
https://orcid.org/0000-0001-8574-9376

**Vicente Medina-Rodríguez**
Tecnológico Nacional de México, Instituto Tecnológico de Chihuahua, México
M21060377@chihuahua.tecnm.mx
https://orcid.org/0009-0004-1522-555X

**Jesús Ezequiel Molinar-Solís**
Tecnológico Nacional de México, Instituto Tecnológico de Cd. Guzmán, México
jesus.ms@cdguzma.tecnm.mx
https://orcid.org/0000-0002-4708-7102

## Resumen

El diseño de sistemas de monitoreo y/o control se basa en modelos matemáticos en el tiempo t, que implican ecuaciones diferenciales y transformadas de Laplace para su solución. Al asociar estos modelos con sistemas discretos y al usar microcontroladores o computadoras para digitalizar estos sistemas, se hace necesario el uso de la Transformada-z. Esta última requiere que se conozca el tiempo de muestreo utilizado, para que todos los modelos concuerden en sus resultados.

El presente trabajo muestra un método práctico para determinar el tiempo de muestreo. Se utilizó el kit de desarrollo ESP-32-S2 y una red con una resistencia y un condensador. Mediante un programa en lenguaje C/C++ y un osciloscopio, se midió el tiempo de muestreo. Se realizaron experimentos para diferentes condiciones de operación del sistema de adquisición de datos, registrando el voltaje del condensador en respuesta a una alimentación escalón unitario. Los resultados reales se compararon gráficamente con los resultados en t, s y z, y se calculó el error entre la respuesta en t y los resultados prácticos.

Esto demostró que el método propuesto genera una buena medida del tiempo de muestreo y que los resultados de los modelos en todos los planos concuerdan con los obtenidos en la práctica. Con esto, esperamos motivar a estudiantes y profesionales a aplicar las matemáticas en el diseño de sistemas digitales, fomentando el uso de herramientas de simulación que permitan el desarrollo de sistemas más rápidos y eficientes. Además, buscamos fortalecer trabajos relacionados con sistemas embebidos, sensores inteligentes, la Internet de las cosas y la industria 4.0.

**Palabras Clave:** Tiempo de muestreo, Sistemas digitales, IoT, sensores inteligentes.

## Abstract

The design of monitoring and/or control systems is based on mathematical models at time *t*, which involve differential equations and Laplace transforms for their solution. Associating these models with discrete systems and the use of microcontrollers or computers to digitize these systems involves the use of the z-transform, which requires knowledge of the sampling time used for all models to coincide in their results.

The present work shows a practical method for determining the sampling time. The ESP-32-S2 development kit and a network with a resistor and a capacitor were used. Using a C/C++ language program and oscilloscope, the sampling time was measured. Experiments were performed for different operating conditions of the data acquisition system, recording the capacitor voltage in response to one unit step supply. The true results were compared graphically with the results in *t,s* and *z* and the error between the response in *t* and the practical results also was calculated. Demonstrating that the proposed method generates a good measure of the sampling time and that the model results in all planes coincide with those obtained in practice. Hoping to motivate students and professionals to apply mathematics in the design of digital systems. Encouraging the use of simulation tools that allow the development of faster and more efficient systems. Strengthening work related to embedded systems, intelligent sensors, internet of things and industry 4.0.

**Keywords:** Sample time, Digital Systems, IoT, Intelligent sensors.

## Resumo

O projeto de sistemas de monitoramento e/ou controle é baseado em modelos matemáticos no tempo t, que envolvem equações diferenciais e transformadas de Laplace para sua solução. Ao associar esses modelos a sistemas discretos e ao usar microcontroladores ou computadores para digitalizar esses sistemas, o uso da Z-Transform torna-se necessário. Este último exige que o tempo de amostragem utilizado seja conhecido, para que todos os modelos concordem em seus resultados.

O presente trabalho mostra um método prático para determinar o tempo de amostragem. Foi utilizado o kit de desenvolvimento ESP-32-S2 e uma rede com um resistor e um capacitor. Usando um programa em linguagem C/C++ e um osciloscópio, o tempo de amostragem foi medido. Experimentos foram realizados para diferentes condições de operação do sistema de aquisição de dados, registrando a tensão do capacitor em resposta a uma alimentação em degrau unitário. Os resultados reais foram comparados graficamente com os resultados em t, s e z, e o erro entre a resposta t e os resultados práticos foi calculado.

Isso demonstrou que o método proposto gera uma boa medida do tempo de amostragem e que os resultados dos modelos em todos os planos concordam com os obtidos na prática. Com isso, esperamos motivar estudantes e profissionais a aplicar a matemática no projeto de sistemas digitais, promovendo o uso de ferramentas de simulação que permitam o desenvolvimento de sistemas mais rápidos e eficientes. Além disso, buscamos fortalecer trabalhos relacionados a sistemas embarcados, sensores inteligentes, internet das coisas e indústria 4.0.

**Palavras-chave:** Tempo de amostragem, Sistemas digitais, IoT, sensores inteligentes.

Revista Iberoamericana para la Investigación y el Desarrollo Educativo
ISSN 2007 - 7467

**Fecha Recepción:** Noviembre 2022 **Fecha Aceptación:** Junio 2023

# Introduction

The present work was inspired by the students of the Electronic Engineering career of the Technological Institute of Chihuahua of the Tecnológico Nacional de México (IT de Chihuahua del TecNM). Throughout their career, these students take five math subjects and, in recent years, there has been a considerable gap in applying mathematics when developing a monitoring or control project. They are so focused on running a program on hardware that they forget to apply mathematics, a very useful tool to which they have dedicated many hours of their lives. Therefore, this work focuses on consolidating in a single document the mathematical tools for system analysis, presenting a method to experimentally determine the sampling time, and building confidence in the use of these tools, showing their usefulness in system design. monitoring and/or control.

It is important to remember that the mathematical model used in many areas of engineering analysis is the linear ordinary differential equation at time t, which will be discussed later. To facilitate its analysis, it is recommended to use the Laplace transform s (F(s)) and, for the analysis of discrete-time systems, the z transform (F(z)) is used (Dobelin, 1985; Ogata, 1995). . The models in time and the models in Laplace are directly related, and there are tables that facilitate the use of these theories. However, to relate these models to discrete-time models, one depends on the value of the sampling time, that is, a system has a unique model at t and s, but z will have a different equation if the sampling time is different.

In the available literature, reviewing different application areas of discrete systems, it can be seen how different authors agree that sampling time is one of the most important variables that directly influences the results obtained. For example, for the study of classical control theory, the identification of systems is essential to determine the characteristics of the plant to be controlled and to be able to design the most appropriate control, as shown by different works for the identification of continuous time systems from of sampled data (Tuma y Jura, 2017; Figwer, 2018; Ghoul et al., 2016).

There are also works that demonstrate how the computational power of the hardware used and the sampling time are important factors that limit the development of real-time applications (Dobrogowski and Kasznia, 2005). Along the same lines, there are analyzes on the differences in algorithm calculations in a measurement system with personal computers at different operating frequencies of the processors (Dobrogowski and Kasznia, 2007).

Methods have been proposed to improve multichannel communication systems, reducing the error in measurement time with a personal computer (Dobrogowski and Kasznia, 2013). Emphasis has been placed on the importance of sampling frequency when developing measurement systems with multiple measurement devices, and on the care that must be taken due to measurement errors and interpretation of results (Vollmer et al., 2019 ). Algorithms have been developed to use different sampling times and correct the main problems in modern, dynamic and non-linear industrial processes, thus improving the quality of the acquired data (Yuan et al., 2021). This includes the detection and correction of sampling time errors (Chakravarthi and Mohan, 2017; Leuciuc, 2017; Neitola, 2020; Qi and Chen, 2018; Quian et al., 2019).

Work has been done on the development of FPGA-based acquisition systems to obtain greater speed in the sampling time (Krishnamoorthy et al., 2018) and the impact of the number of samples and the sampling time on the calculation has been evaluated. of variables, such as the estimation of wind energy density (Gross et al., 2020) and the analysis of the effects of sampling time in a network of an integral proportional control and a second order plant (Santhosh and Noronha, 2018). .

In data science, there are works that highlight the importance of sampling time and information processing speed (Bei-Bei and Xue-bo, 2015). The importance of the usefulness of the data is highlighted insofar as it can be quickly analyzed to reveal valuable information. It shows that with high-quality data, we can increase revenue, reduce costs, and reduce risk. It is concluded that the data sample acquisition technique is very important and has an impact on the quality of these data. (Liu et al., 2019).

As can be observed in the consulted literature, the sampling time or its equivalent, the sampling frequency, is the fundamental variable of study and analysis to obtain reliable results in many applications such as data acquisition, system control, data science. , among other. The main problem is the determination of the sampling time (T), since this depends on the characteristics of the hardware to be used, the speed of the central processing unit (CPU), the speed of the input and output peripherals (I/ S) and the size of the program to use. By obtaining the sampling time, we can now make use of the z transform, the most widely used tool in analysis to describe systems in discrete time (Horváth, 2020; Ogata, 1995). That is why the present work focuses on concentrating in a single document the classical tools for the analysis of systems in the domains of time (t), frequency (s) and discrete frequency (z). In addition, a practical method is presented to determine the sampling time (T) in systems with low computational power microcontrollers, with the aim of developing monitoring and/or control applications. With this work, the aim is to facilitate the use of mathematics in the t, s and z domains, and encourage its use to simplify the design of monitoring and/or control systems, using simulators and obtaining optimal results.

# Method

Mathematical models of systems in continuous time and in the Laplace domain were used. As mentioned above, the widely used mathematical model in various areas of engineering analysis is the ordinary linear differential equation with constant coefficients. (Dobelin, 1985):

$$a_n \frac{d^n q_0}{dt^n} + a_{n-1} \frac{d^{n-1} q_0}{dt^{n-1}} + \cdots + a_1 \frac{dq_0}{dt} + a_0 q_o = b_m \frac{d^m q_i}{dt^m} + b_{m-1} \frac{d^{m-1} q_i}{dt^{m-1}} + \cdots + b_1 \frac{dq_i}{dt} + b_0 q_i \qquad (1)$$

Where:

$q_o$ → It is the output variable or physical system response.
$q_i$ → It is the input or excitation variable of physical system
$t$ → It is the time
$a_{n,} b_m$ → Physical parameters of system.

Defining the differential operator $D = \frac{d}{dt}$ Equation (1) can be rewritten as (Dobelin 2004):

$$(a_n D^n + a_{n-1} D^{n-1} + \cdots + a_1 D + a_0) q_o = (b_m D^m + b_{m-1} D^{m-1} + \cdots b_1 D + b_0) q_i \qquad (2)$$

The use of this model is widespread because it is directly available analytical solutions regardless of the n order of the equation, allowing the treatment of linear systems of arbitrary complexity.

In the analysis, design and applications of measurement and/or control systems, the concept of operational transfer function is very useful. The operational transfer function relates the output $q_0$ to the input $q_i$ and is defined as (Dobelin 2004):

$$\frac{q_o}{q_i} = \frac{b_m D^m + b_{m-1} D^{m-1} + \cdots b_1 D + b_0}{a_n D^n + a_{n-1} D^{n-1} + \cdots + a_1 D + a_0} \qquad (3)$$

From Equation (1), the most common dynamic system models can be defined: those of zero order, first order and second order (Dobelin, 1985). The Laplace Transform is commonly used to study linear systems. The Laplace transfer function is defined as the ratio between the Laplace transform of the output and the Laplace transform of the input, with initial conditions equal to zero. Therefore, Equation (3) can be rewritten as shown below (Dobelin, 2004):

$$\frac{q_o(s)}{q_i(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \qquad (4)$$

The model of a dynamic system of zero order is defined by the equation:

$$a_0 q_o = b_0 q_i \tag{5}$$

Equation (5) in standardized form is written as $q_o = K q_i$ where K is the static sensitivity of the system or steady state gain defined by $K = \frac{b_0}{a_0}$.

The transfer function in Laplace is:

$$\frac{Q_o}{Q_i}(s) = K \tag{6}$$

The basic model of a first order dynamical system is defined by the equation:

$$a_1 \frac{dq_o}{dt} + a_0 q_0 = b_0 q_i \tag{7}$$

And its standardized form is:

$$\tau \frac{dq_o}{dt} + q_0 = K q_i \tag{8}$$

As $\tau = \frac{a_1}{a_0}$ is the time constant and $K = \frac{b_0}{a_0}$.is the steady state gain.

And its transfer function in Laplace is:

$$\frac{Q_o}{Q_i}(s) = \frac{K}{\tau s+1} \tag{9}$$

The basic model of a second order dynamical system is defined by the equation:

$$a_2 \frac{d^2 q_o}{dt^2} + a_1 \frac{dq_o}{dt} + a_0 q_o = b_o q_i \tag{10}$$

The standard model is of the form:

$$\frac{1}{w_n^2}\frac{d^2 q_o}{dt^2} + \frac{2\varsigma}{w_n}\frac{dq_o}{dt} + q_o = K q_i \tag{11}$$

As $w_n = \sqrt{\frac{a_0}{a_2}}$ v, $\varsigma = \frac{a_1}{2}\sqrt{a_2 a_0}$ is the damping ratio y $K = \frac{b_0}{a_0}$. is the steady state gain.

## Discrete-time mathematical models of systems, *z*

The z-transform (z) is an important tool in the analysis and design of discrete-time systems. It plays a role like that played by the Laplace transform in continuous time problems.

The one-sided z-transform of a function of time x(t), or of the sequence of values x(kT), where k≥0 and T is the sampling time, is defined by the following equation (Ogata, 1995):

$$X(z) = Z[x(t)] = Z[x(kT)] = \sum_{k=0}^{\infty} x(kT) z^{-1} \tag{12}$$

The one-sided-transform is used in all applications associated with discrete-time systems since the sampling time is not defined for negative values.

## Pulse sampling

A dummy sampler commonly called a pulse sampler is considered. The output of this sampler is considered as a train of impulses that starts at t=0, with the sampling period equal to T and the magnitude of each impulse equal to the sampled value of the signal in continuous time at the corresponding sampling instant. Assume that x(t)=0 for t<0, then the pulse-sampled output is a sequence of pulses, with the magnitude of each pulse equal to the value of x(t) at the corresponding time instant. If we use the notation x* (t) to represent the pulse-sampled output, this signal can be represented as (Ogata, 1995):

$$x^*(t) = x(0)\delta(t) + x(T)\delta(t-T) + \cdots + x(kT)\delta(t-kT) \quad (13)$$

o

$$x^*(t) = \sum_{k=0}^{\infty} x(kT)\delta(t-kT)$$

The Laplace transform of (13) is:

$$X^*(s) = L[x^*(t)] = x(0)L[\delta(t)] + x(T)L[\delta(t-T)] + x(2T)L[\delta(t-2T)] + \cdots$$

$$X^*(s) = x(0) + x(T)e^{-Ts} + x(2T)e^{-2Ts}$$

$$X^*(s) = \sum_{k=0}^{\infty} x(kT)e^{-kTs} (14)$$

If we define $e^{-Ts} = z$ ó $s = \frac{1}{T}\ln z$ $s = \frac{1}{T} lnz$, so:

$$X^*(s)|_{s=\left(\frac{1}{T}\right)\ln z} = \sum_{k=0}^{\infty} x(kT)z^{-1} \ (15)$$

As can be seen, the second member of (12) and (15) are exactly the same, so we can express:

$$X^*(s)|_{s=\left(\frac{1}{T}\right)\ln z} = X^*\left(\frac{1}{T} lnz\right) = X(z) = \sum_{k=0}^{\infty} x(kT)z^{-1} \ (16)$$

In the available literature it is easy to find tables of functions over time and their corresponding Laplace-s or z-transforms (Dobelin, 1985; Ogata, 1995). For example, the function over time $f(t) = e^{-at}$ its Laplace transform is $F(s) = \frac{1}{s+a}$, its equivalent in pulse sampling $f(kT) = e^{-akT}$ and its $z$ transform of the form: $F(z) = \frac{1}{1-e^{-aT}z^{-1}}$.

As can be seen, any function in time has a single equivalent function in Laplace but it will have an infinite number of equivalent functions in impulse sampling and in the Transform-***z***, which will depend on the sampling time used. That is why, in order to use these tools in the design and analysis of systems, it is necessary to correctly define the sampling time T.

# Analysis of the response of a first-order real system to an input of a step function

For this exercise we will use as a model a basic circuit formed by a constant voltage source V, a resistor R and a capacitor C (RC network). The voltage across the capacitor is defined by $v_c(t)$. The switch SW is open at t<0 and at t≥0 it closes, as shown in Figure 1. Figure 1a represents the circuit when t<0, Figure 1b shows the circuit for t≥0, at the moment of closing the switch we apply a voltage to the RC circuit of the form Vu(t) where u(t) is the unit step in which its value is zero for t<0 and its value is 1 when t≥0 , as shown in Figure 1c and mathematically defined as:

$$u(t) = \begin{cases} 0 t < 0 \\ 1 t \geq 0 \end{cases} \quad (17)$$

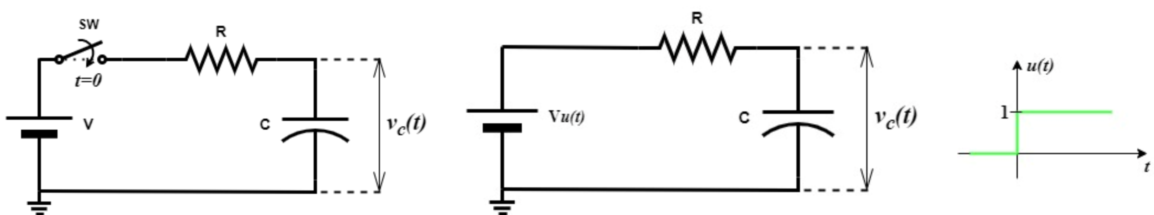
Applying Kirchhoff's current law to the circuit in Figure 1c, we obtain the equation:

$$\frac{v_c(t) - Vu(t)}{R} + C\frac{dv_c(t)}{dt} = 0 (18)$$

o

$$C\frac{dv_c(t)}{dt} + \frac{v_c}{R} = \frac{Vu(t)}{R} (19)$$

**Figure 1.-** RC circuit. **a)** RC network in t<0, **b)** RC network in t>0 and **c)** Unit step $\mu(t)$.




(a) (b) (c)

Own work

Equation (19) represents a first-order system of the form of Equation (7). Now to solve the equation, dividing (19) by C:

$$\frac{dv_c(t)}{dt} + \frac{v_c}{RC} = \frac{Vu(t)}{RC} (20)$$

Applying the Laplace transform to (20) and making the change of variable $\tau = RC$:

$$sV_c(s) + v_c(0) + \frac{V_c(s)}{\tau} = \frac{V}{\tau s} (21)$$

Applying initial conditions $v_c(0)=0$

$$\frac{V_c(s)(\tau s+1)}{\tau}=\frac{V}{\tau s} \quad (22)$$

$$V_c(s)=\frac{V\left(\frac{1}{\tau}\right)}{s\left(s+\frac{1}{\tau}\right)} \quad (23)$$

Using the tables to antitransform Equation (23), $v_c(t)$ is obtained in response to the step of magnitude V:

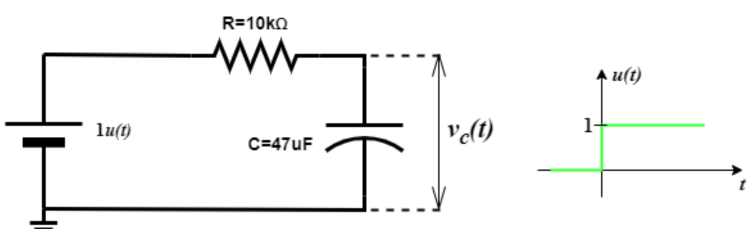
$$v_c(t)=V\left(1+e^{-t/\tau}\right) \quad (24)$$

With this, we have the model of the first order system in the time defined by Equation (24) and its equivalent in Laplace defined by Equation (23). Using tables of the z transform we obtain the discrete circuit model of Figure 1:

$$V_c(z)=\frac{V\left(1-e^{-\left(t/\tau\right)T}\right)z^{-1}}{(1-z^{-1})\left(1-e^{-\left(t/\tau\right)T}z^{-1}\right)} \quad (25)$$

As can be seen, Equations (23) and (24) are unique models in t and s for the circuit of Figure 1, but analyzing Equation (25) it is observed that there is an infinite number of equations in z that depends on the value that is use for the sampling time T. Analyzing the circuit of Figure 2, we obtain its response to the unit step, by means of (23) and (24) Equations (26) and (27) are obtained. Evaluating (25) for four different values of sampling time (T=0.1s, T=0.5, T=1s and T=2s) four equations are obtained in the z plane, as shown in (28a) to (28d) respectively:

**Figure 2.-** RC circuit with unit step input.




Own work

$$v_c(t) = 1 - e^{-2.127t} (26)$$

$$V_c(s) = \frac{2.127}{s(s+2.127)} (27)$$

Using (25) for $T = 0.1s$ is obtained:

$$V_c(z) = \frac{0.1917}{(z-0.8083)} (28a)$$

Using (25) for $T = 0.5s$ is obtained:

$$V_c(z) = \frac{0.6549}{(z-0.3451)} (28b)$$

Using (25) for $T = 1s$ is obtained:

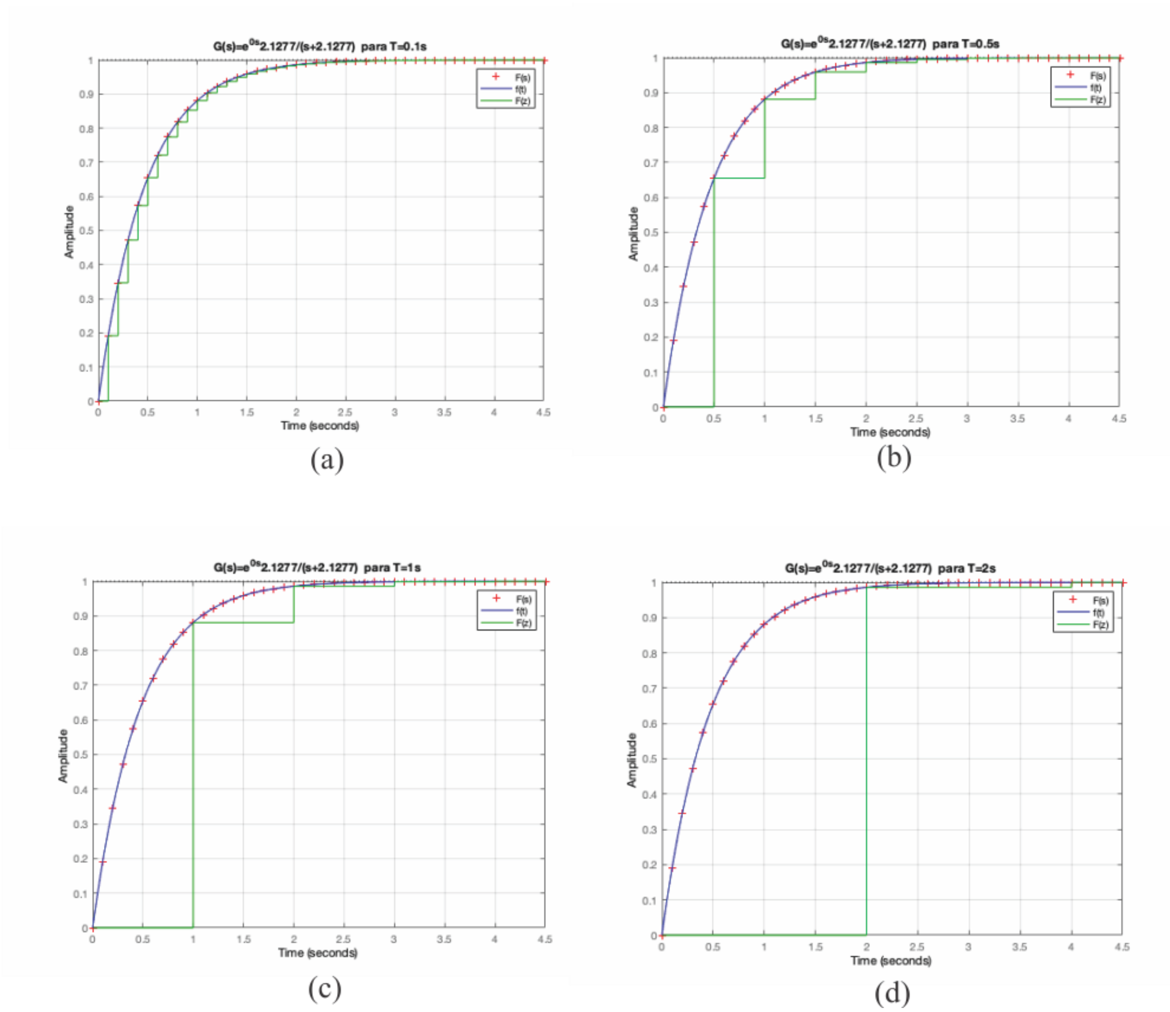
$$V_c(z) = \frac{0.8809}{(z-0.1191)} (28c)$$

Using (25) for $T = 2s$ is obtained:

$$V_c(z) = \frac{0.9858}{(z-0.1419)} (28d)$$

Using MATLAB®, Equations (26) to (28) were simulated and Figure 3 was obtained, in which it can be seen how with a sampling time of T=0.1 s we can obtain good information on the capacitor charge. the discrete-time analysis, Figure 3a. With a sampling time of T=0.5s, information on the charge of the capacitor is lost, there are only 4 samples with respect to the response in t or s and the first sample and the capacitor reached more than 60% of its charge, Figure 3b. With a sampling time of T=1s, only two samples of the capacitor charge can be obtained and, as seen in Figure 3c, more than 80% of the information is lost. With a sampling time of T = 2s, there is only one sample and the capacitor has already reached more than 90% of its charge, so there is not good information on the charge of the capacitor Figure 3d.

**Figure 3.-** Voltage *v(t)* of Figure 2 RC Network. a) *T*=0.1s, b) *T*=0.5, c) *T*=1 s y d) *T*= 2s




Own work

So far, the importance of correctly defining the sampling time T to obtain quality information has been demonstrated. When this information is obtained by means of a data acquisition system, the task of defining the real sampling time is not easy since, in a real data acquisition system, T will depend on the hardware used and the software developed. The sampling time in a data acquisition system $T_m$ can be defined as follows:

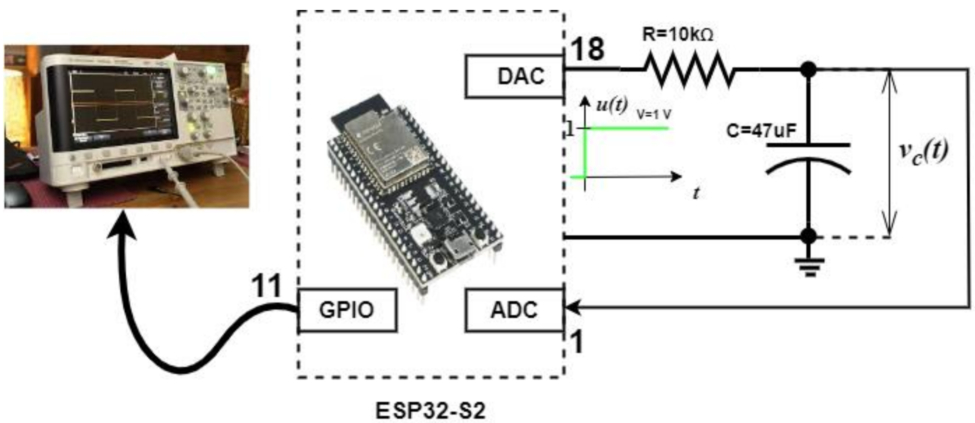
$$T_m = T_H + T_S \quad (29)$$

Where $T_H$ is the time spent by the hardware and $T_S$ is the time spent executing the software.

**Design of experiments to determine the sampling time in a data acquisition system based on the response to an input of a step function**

To determine the $T_S$ it is possible to use the development tools available for microcontrollers, in which the assembly code can be analyzed and the program block execution time estimated. But this technique is feasible on small microcontrollers and is not widespread for more powerful microcontrollers or for all brands. For this reason, we propose to program a digital output as a high indicator ("logical 1") at the beginning of a program and change to low ("logical 0") at the end of the program and determine the. To demonstrate this method, an ESP-32-S2 (ADQ) general purpose development kit was used and the circuit of Figure 4 was assembled.

**Figure 4.-** Data acquisition system to determine the sampling time, $T_m$




Own work

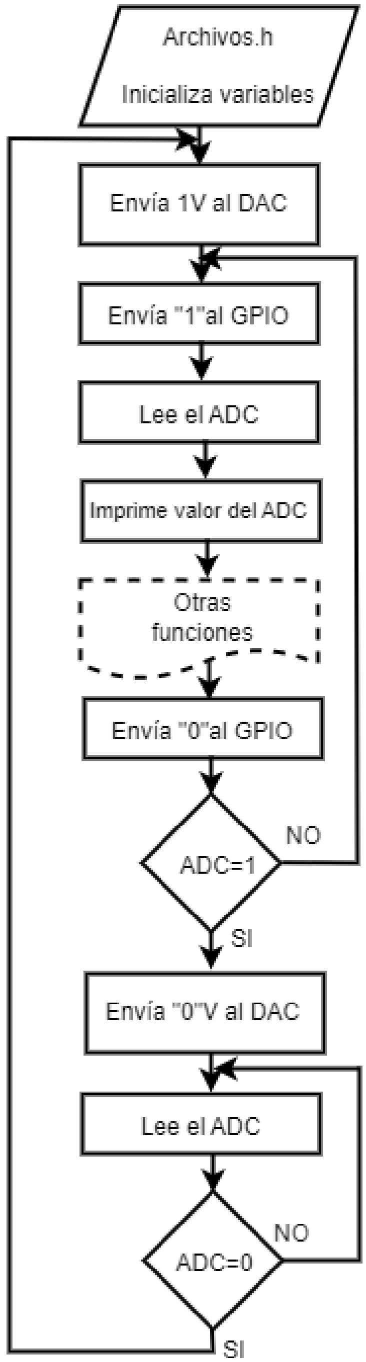
The ESP-32-S2 is a powerful tool for developing projects related to the Internet of Things, Internet of Things (IoT). Its main features are (Ozan, 2021):

- CPU and Memory. It is a 32-bit Xtesa® LX6 microprocessor with a 240 Mhz clock and 600 million Instructions Per Second (MIPS). It has memory of 448 KB of ROM, 520 KB of RAM and 16 KB of RTC.
- Peripherals. GPIOs, ADC, DAC, SPI, I2C, I2S, UART, CAN, IR, PWM, Contact sensor and hall effect sensor.
- Connectivity. Wi-Fi 802.11n (2.4 GHZ) up to 150 Mgps.
- Energy consumption. Ultra low power consumption, Ultra-Low-Power (ULP). 100 in standby mode (sleep mode).

There are different versions of the ESP-32, with Bluetooth connectivity, eMMC/SD, for operation in high temperatures, among others.

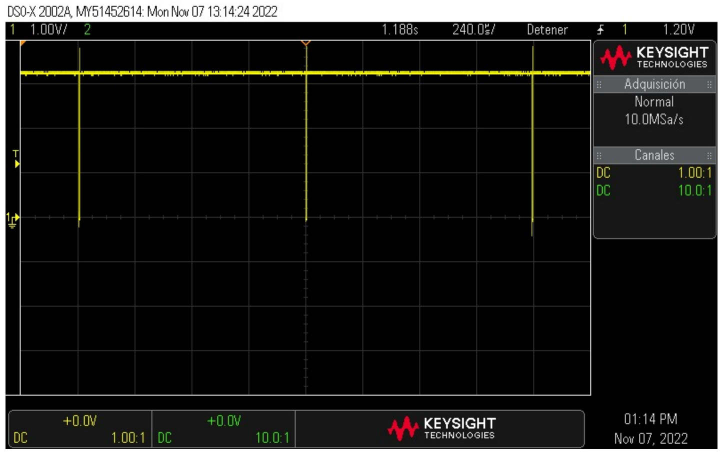
To determine the sampling time $T_m$ using the ECLIPSE development platform, a program was developed in C/C++ language. The flowchart is shown in Figure 5, in which the ".h files" are loaded at the beginning and the variables are initialized. Then 1 V is sent through the digital-analog converter (DAC) to the RC network and immediately after a digital port is set to logic "1" indicating the start of the step input to the RC network. Next, the value of the capacitor charge is read through the digital-analog converter (DAC). A block called "Other functions" is added in which the user can program functions such as processing information, displaying information, communicating information, among others. Then, a logic "0" is sent to the GPIO to indicate the end of sampling to the RC network. These instructions are repeated until the ADC reading has the value of 1 V, which indicates that the capacitor has already charged to the value of the step sent through the DAC. After the capacitor is charged that routine ends. Now 0 V is sent through the DAC to the RC network and the capacitor voltage is continuously read through the DAC until it reads 0 V, indicating that the capacitor has finished discharging. Then the entire program is repeated several times, to have enough information to be able to make measurements with the oscilloscope. By connecting an oscilloscope to the output of the GPIO, a signal is obtained as shown in Figure 6, this signal was obtained with a KEYSIGHT model oscilloscope DSO-X-20002ª.

**Figure 5.-** Flowchart of the developed program




Own work

**Figure 6.-** Signal obtained from the ADQ GPIO with a DSO-X-20002A oscilloscope.




Own work

Following what was previously presented, 5 experiments were programmed: In the first, in the "Other functions" block of the flowchart of Figure 5, nothing was programmed, which generated the maximum speed of the ADQ and the lowest speed was obtained. $T_m$. Later, different delays were programmed in the "Other functions" block to obtain a sampling time of 10ms, 50ms, 0.166s and 0.333s. The following section shows the results obtained.

## Results

The first thing that was done was to physically measure the actual value of the resistor, R, and the capacitor, C used. The values obtained were *R*=9.76 KΩ and *C*=48 µF. With this we recalculate Equations (26) and (27) for $\frac{1}{\tau} = 2.135s^{-1}$:

$$v_c(t) = 1 - e^{-2.135t} (30)$$

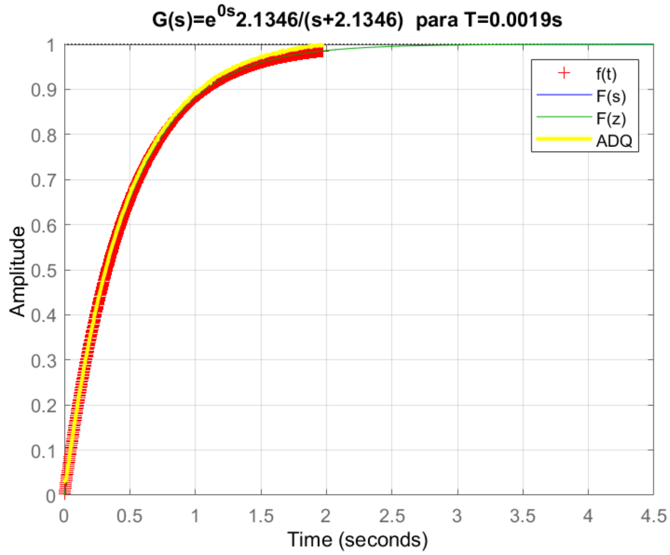
$$V_c(s) = \frac{2.135}{s(s + 2.135)} (31)$$

By measuring the oscilloscope signal, the $T_m = 1.9$ ms, is the minimum sampling time, with which it was obtained that the equivalent equation in $z$.

$$T_m = 0.0019s: V_c(z) = \frac{0.1917}{(z - 0.8083)} (32)$$

Through the developed program, the real readings obtained with the ADQ system of Figure 4 were recorded. Figure 7 graphically shows the results obtained from the mathematical model of the load of the RC network in ***t***, ***s***, ***z*** and the practical result.

**Figure 7.-** Results obtained by Equations (30-32) and the ADQ




Own work

The equations in for the sampling times of 10ms, 50ms, 0.166s and 0.333s are:

Using (25) for $T = 10ms$ is obtained:

$$V_c(z) = \frac{0.02112}{(z - 0.9789)} (33)$$

Using (25) for $T = 50ms$ is obtained:

$$V_c(z) = \frac{0.1012}{(z - 0.8988)} (34)$$

Using (25) for $T = 0.166s$ is obtained:

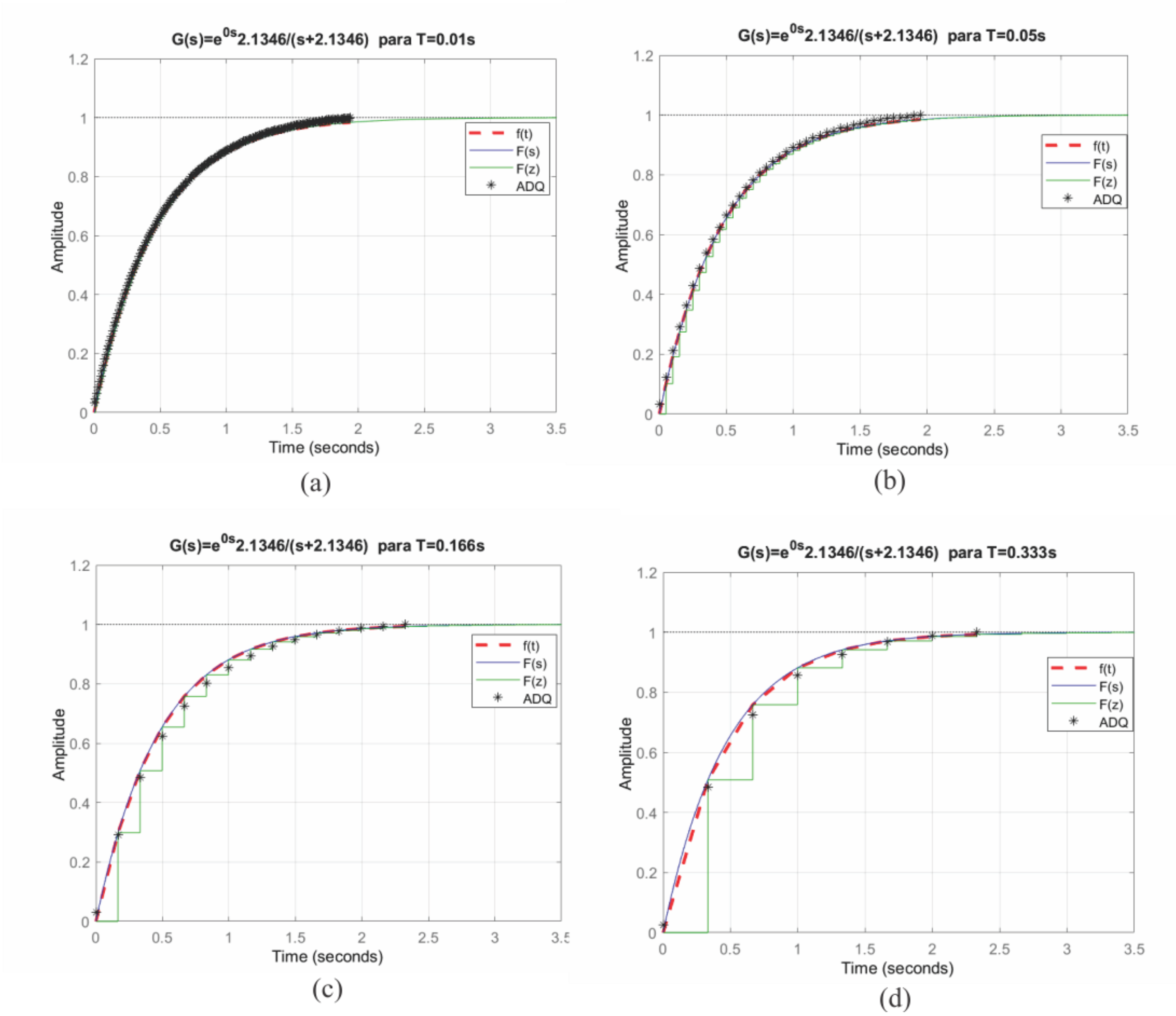
$$V_c(z) = \frac{0.2984}{(z - 0.7016)} (35)$$

Using (25) for $T = 0.333s$ is obtained:

$$V_c(z) = \frac{0.5088}{(z - 0.4912)} (36)$$

Figure 8 shows the results of the simulation of Equations (30,31) and (33-36) together with the results obtained practically at the different sampling times programmed in the ADQ. Each graph shows the results of the models in t,s,z and those obtained in a practical way with the ADQ.

**Figura 8.-** Resultados obtenidos con el ADQ a diferentes tiempos de muestreo y la comparación con los modelos obtenidos en t,s y z : a) $T_m = 10\text{ms}$, b) $T_m = 50\text{ms}$, c) $T_m = 0.166\text{s}$ y d) $T_m = 0.333\text{ms}$




Own work

As can be seen in Figure 8, the data obtained practically coincides with the models defined in t,s,z . To make a quantitative statement of the results obtained, the root mean square error (RMSE) equation was used (37). Comparing the results calculated with Equation (30) in the time domain and the values obtained by the ADQ at the different $T_m$.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(v_{c,i} - v_{ADQ,i}\right)^2} \quad (37)$$

As $v_{c,i}$ represents the voltage on the capacitor obtained by Equation (30), v_(ADQ,i) represents the voltage on the capacitor obtained with the ADQ of Figure 4 and N is the number of values of $v_c(t)$. Table 1 shows the results. Column 1 lists the different sampling

Revista Iberoamericana para la Investigación y el Desarrollo Educativo
ISSN 2007 - 7467

times $T_m$, Column 2 shows the sampling frequency $f_m = {}^{1}/_{T_m}$, Column 3 shows the RMSE and Column 4 shows percent of the RMSE.

With the results obtained, it can be demonstrated that the proposed method presents a good accuracy to determine the $T_m$ of an ADQ system in which its response is combined between the hardware used and the program that is implemented. The results obtained in practice are similar to the data from the t,s,z models. With the results obtained in Table 1, it can be verified that as the slower T the error increases. For all of the above, the importance of knowing the $T_m$ is demonstrated. In this way, all the mathematics learned in the training of engineers can easily be applied, resulting in the design of more robust and efficient monitoring or control systems. This methodology will also facilitate the design of next-generation systems with the Internet of Things or Industry 4.0 approach.

**Table 1.** Root means square error between $v_c(t)$. and the data obtained practically.

| $T_m$ (s) | $f_m(Hz)$ | | RMSE ( $v_c(t), v_{c,ADQ}$ ) | %RMSE |
|---|---|---|---|---|
| 0.0019 | 526.31 | | 0.0117 | 1.17 |
| 0.010 | 100 | | 0.0122 | 1.22 |
| 0.050 | 20 | | 0.0127 | 1.27 |
| 0.166 | 6 | | 0.0198 | 1.98 |
| 0.333 | 3 | | 0.0202 | 2.02 |

Own work

## Discussion

In the present work, the mathematical modeling of systems in time based on the ordinary equation with constant coefficients has been presented. Its equivalence in the Laplace transform and the z transform. A practical example of these models is shown in Figure 1 and Figure 2, an RC network in which its response to a supply voltage of the unit step type can be represented by Equation (24) in the t domain, by the Equation (23) in the s domain and by Equation (25) in the z plane. The importance of the sampling time was also demonstrated so that there is a good relationship between the results of the z plane and the models in t and s, as can be seen in Equations (26-28) and the graphic results presented in Figure 3. As can be seen, the shorter the sampling time, the better information on the behavior of the system under study will be obtained. This is consistent with what has been reviewed in the available literature on sampling time.

To show a practical example of this type of modeling and compare the results, a data acquisition system like the one shown in Figure 4 was used. In which the flowchart of Figure 5 was programmed in C/C++ language With this, the minimum sampling time of the ADQ was obtained, which was $T_m = 0.0019s$ or a frequency of $f_m = \frac{1}{T_m} = 526.31Hz$, this represents the highest speed of the ADQ since nothing was programmed in the "Other functions" block of Figure 5. The theoretical model obtained in the z plane is shown in Equation (32) and Figure 7 graphically shows the results obtained practically with the ADQ of the response of the RC network to a unit step input and the models in t, syz. In which a very good agreement is observed in their results. This is because of the speed of the ADQ and that it was programmed exclusively to record the voltage reading on the capacitor as shown in Figure 4. Simulating the programming of other activities in the ADQ, different time delays were programmed in the "Other functions" block of the program in Figure 5. The models in t and s do not change, Equations (30 and 31) and their equivalents in z for four different sampling times are represented by Equations (33-36). The results obtained practically and the models in t, s and z are shown in Figure 8. As can be seen in the graphs, there is a concordance between the data obtained with the ADQ represented by the asterisks in the graph and the green line that represents the z model As can be seen, as the sampling time slows down, the recorded information on the capacitor charge is lost.

In addition, the results obtained practically and the model in time represented by Equation (30) were quantitatively evaluated using the RMSE with Equation (37). Getting a very small error. And confirming a good agreement in the practical results and the mathematical models presented.

Determining the sampling time for the design of monitoring and/or digital control systems is a fundamental part. What is presented here coincides with Tuma and Jura (2017), Figwer (2018) and Ghoul et al., (2016) in which the identification of the plant is exposed from sampled data, allowing the design of more appropriate controllers.

In addition, we have explained how adding more lines to a program in a monitoring and/or control system increases the sampling time, therefore, in some works, hardware with greater speed in its computational power will be needed, coinciding with what Dobrogowski and Kasznia, (2005) and Bei-Bei and Xue-bo (2015), that the computational power of the hardware used limits the development of applications in real time. Therefore, applying what is presented here will help designers, in a short time, to test and select the most suitable hardware for their applications.

It is important to highlight that currently the use of data acquisition systems has become widespread and it is required that they generate high quality data since they are the basis of many applications and as mentioned by Liu et al., (2019), with data of high quality we can increase revenue, reduce costs and reduce risk. Data acquisition systems are the basis for the development of monitoring and/or control systems, so by applying the tools presented here we can guarantee the development of systems that generate quality data.

## Conclusions

In the present work, the mathematical modeling of systems based on the linear ordinary differential equation with constant coefficients, which is used in different engineering applications, was presented. The operational transfer function in time and in the Laplace transform, were shown in the s-plane. The mathematical models of zero, first and second order systems were exposed. In addition, the impulse sampling theory and the z-transform were presented, since this theory is widely used in the design and applications of discrete systems, such as digital monitoring and/or control.

Monitoring and/or control systems modeled with the z transform have multiple mathematical representations that depend on the sampling time used. In practice, it is not easy to obtain the sampling time, since it depends on the hardware used and the size of the software that is programmed. Therefore, it is often not easy to apply the knowledge acquired.

In this work, a practical way to find the sampling time of a data acquisition system was shown, using an RC network and an oscilloscope. It was shown that it is possible to obtain the sampling time with a good accuracy and that the response of a system is similar in the different mathematical models and in practice. With this, we hope that it can motivate the study and application of monitoring and/or control systems, and the development of more efficient systems in which what has been learned in engineering training is applied. In addition, this methodology can help characterize the performance of embedded systems and modern systems based on the philosophy of the Internet of things or industry 4.0.

## Contributions to Future Research Lines

This work can contribute to the design of digital control algorithms based on the classical theory in z. The exposed method to determine the sampling time can be easily implemented in any electronic system in which the application is developed, allowing the user to evaluate if their hardware meets the design requirements or will allow them to select the appropriate hardware for their application.

Nowadays, monitoring and/or control systems must be developed under the philosophies of the Internet of Things or Industry 4.0, which require communication with other systems. Therefore, monitoring and/or control systems must be developed taking into account the time spent in data communication and analysis. The proposed methodology can be used to characterize these systems and develop more robust and better performing products.

In the educational field, this will allow the study of monitoring and/or control systems to be more complete by relating the agreement between mathematical models and practice, which will result in improved study programs and better prepared professionals.

## References

Bei-Bei, M. y Xue-bo, J. (julio de 2015). *Compressing Sampling for Time Series Big Data*. IEEE 2015, 34th Chinese Control Conference (CCC), Hangzhou, China. https://doi.org/10.1109/ChiCC.2015.7260410

Chakravarthi, M.V.N. y Mohan, B.Ch. (diciembre de 2017). *Detection and Correction of Sampling-Time-Errors in an N-Channel Time-Interleaved ADC using Genetic Algorithm*. 14th IEEE India Council International Conference, Roorkee, India. https://doi.org/10.1109/INDICON.2017.8487561

Dobelin, O.E. (1985). *Control System Principle and Design*.

Dobelin, O.E. (2004). *Measurement Systems Applications and Design*.

Dobrogowski, A. y Kasznia, M. (agosto de 2005). *Testing of the Methods of Real-Time MTIE Calculation*. IEEE International Frequency Control Symposium and Exposition, Vancouver, Canada. https://ieeexplore.ieee.org/document/1573965.

Dobrogowski, A. y Kasznia, M. (mayo de 2007). *Real-time Assessment of Allan Deviation and Time Deviation*. IEEE International Frequency Control Symposium Joint with the 21st European Frequency and Time Forum, Geneva, Switzerland. https://ieeexplore.ieee.org/document/4319204

Dobrogowski, A. y Kasznia, M. (julio de 2013). *Experimental Tests of the Real-Time MTIE Assessment Methods for Multi-Channel Time Error Measurement*. Joint European Frequency and Time Forum & International Frequency Control Symposium (EFTF/IFC), Prague, Czech Republic. https://doi.org/10.1109/EFTF-IFC.2013.6702132

Figwer, J. (agosto de 2018). *Continuous-Time Nonlinear Block-Oriented Dynamic System Identification from Sampled Step and Step-Like Responses*. 23rd International Conference on Methods & Models in Automation & Robotics (MMAR), Miedzyzdroje, Poland. https://ieeexplore.ieee.org/document/8486118.

Ghoul, Y., Taarit, I. K. y Ksouri, M. (diciembre de 2016). *Online Identification of Continuous-Time Systems with Multiple-Input Time Delays from Sampled data Using Sequential Nonlinear Least Square Method from Sampled Data*. 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, Sousse, Tunisia. https://ieeexplore.ieee.org/document/7952028.

Gross, M., Magar, V. y Peña, P., (2020). The Effect of Averaging, Sampling, and Time Series Length on Wind Power Density Estimations. *Sustainability*, 12(8), https://doi.org/10.3390/su12083431

Horváth, I., Mészáros, A. y Telek, M. (2020). Numerical Inverse Transformation Methods for Z-Transform. *Mathematics*, 8(4). https://doi.org/10.3390/math8040556.

Krishnamoorthy, S., Morales, E., Ashmanskas, W.J., Mayers, G., Hu, R.W., LeGeyt, B.C., Karp, J.S. y Surti S. (noviembre de 2018). *A modular waveform-sampling data acquisition system for time-of-flight PET*. IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings, Sydney, NSW, Australia. https://ieeexplore.ieee.org/document/8824368.

Leuciuc, A. (octubre de 2017). *Sampling Time Calibration Method for MultiChannel Interleaved ADCs*. IEEE International Symposium on Circuits and Systems, Guiyang, China. https://ieeexplore.ieee.org/document/8050247.

Liu, H., Sang, Z. y Karali, S. (diciembre de 2019). *Approximate Quality Assessment with Sampling Approaches*. International Conference on Computational Science and Computational Intelligence, Las Vegas, USA. https://ieeexplore.ieee.org/document/9071249.

Neitola, M., (octubre de 2020). *Digital Timing Error Calibration of Time-Interleaved ADC with Low Sample Rate.* IEEE Nordic Circuits and Systems Conference, Oslo, Norway. https://ieeexplore.ieee.org/document/9264991.

Ogata, K., (1995). *Discrete-Time Control Systems*.

Ozan, O.V., (2021). *Developing IoT Proyects with ESP32*.

Qi, R. y Chen, Z., (diciembre de 2018). *Method of Time Sampling Error of Time-Interleaved ADC Based on Statistical Rule*. 4th International Conference on Computer and Communications, Chengdu, China. https://ieeexplore.ieee.org/document/8781041.

Quian, X., Hengkui, W., y Wenxing, Z., (noviembre de 2019). *The Fast Time-Interleaved ADC Channel Mismatch Calibration Method Based on Equivalent Sampling and Dynamic Reference Point*. 14th IEEE International Conference on Electronic Measurement & Instruments, Changsha, China. https://ieeexplore.ieee.org/document/9101419.

Santhosh, K. y Noronha, N. (junio de 2018). *Analysis of sampling on process timing parameters in a networked control system.* International Conference on Power, Energy and Environment: Towards Smart Technology, Shillong, India. https://ieeexplore.ieee.org/document/8659086.

Tuma, M. y Jura, P. (noviembre de 2017). *Comparison of different approaches to continuous-time system identification from sampled data*. European Conference on Electrical Engineering and Computer Science, Bern, Switzerland. https://ieeexplore.ieee.org/document/8411997.

Vollmer, M., Blasing, D. y Kaderali, L. (septiembre de 2019). *Alignment of Multi-Sensored Data: Adjustment of Sampling Frequencies and Time Shifts*. Computing in Cardiology (CinC), Singapore, Malasya. https://ieeexplore.ieee.org/document/9005801.

Yuan, X., Li L., Wang, K., y Wang, Y. (2021). Sampling-Interval-Aware LSTM for Industrial Process Soft Sensing of Dynamic Time Sequences with Irregular Sampling Measurements. *IEEE Sensors Journal*, 21(9), https://doi.org/10.1109/JSEN.2021.3056210.

| Rol de Contribución | Autor (es) |
|---|---|
| Conceptualización | José Rivera Mejía |
| Metodología | José Rivera Mejía |
| Software | Vicente Medina Rodríguez |
| Validación | Jesús Ezequiel Molinar-Solís |
| Análisis Formal | José Rivera Mejía |
| Investigación | Vicente Medina Rodríguez (principal), José Rivera (que apoya) |
| Recursos | José Rivera Mejía |
| Curación de datos | Vicente Medina Rodríguez |
| Escritura - Preparación del borrador original | José Rivera Mejía (principal), Jesús Ezequiel Molinar-Solís (que apoya) |
| Escritura - Revisión y edición | José Rivera Mejía (principal), Jesús Ezequiel Molinar-Solís (que apoya) |
| Visualización | José Rivera Mejía |
| Supervisión | José Rivera Mejía |
| Administración de Proyectos | José Rivera Mejía |
| Adquisición de fondos | José Rivera Mejía |