

<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-Mejía

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.

Introducción

El presente trabajo se inspiró en los estudiantes de la carrera de Ingeniería Electrónica del Instituto Tecnológico de Chihuahua del Tecnológico Nacional de México (I.T. de Chihuahua del TecNM). A lo largo de su carrera, estos estudiantes cursan cinco materias de matemáticas y, en los últimos años, se ha notado una separación considerable en aplicar las matemáticas cuando desarrollan algún proyecto de monitoreo o control. Están tan concentrados en hacer funcionar un programa en un hardware que se olvidan de aplicar las matemáticas, herramienta muy útil a la que han dedicado muchas horas de su vida. Por lo tanto, este trabajo se centra en consolidar en un solo documento las herramientas matemáticas para análisis de sistemas, presentar un método para determinar experimentalmente el tiempo de muestreo y crear confianza en el uso de estas herramientas, mostrando su utilidad en el diseño de sistemas de monitoreo y/o control.

Es importante recordar que el modelo matemático utilizado en muchas áreas de análisis de la ingeniería es la ecuación diferencial lineal ordinaria en el tiempo t , que se abordará más adelante. Para facilitar su análisis, se recomienda utilizar la transformada de Laplace s ($F(s)$) y, para el análisis de sistemas en tiempo discreto, se utiliza la transformada z ($F(z)$) (Dobelin, 1985; Ogata, 1995). Los modelos en el tiempo y los modelos en Laplace están directamente relacionados, y existen tablas que facilitan el uso de estas teorías. Sin embargo, para relacionar estos modelos con modelos de tiempo discreto, se depende del valor del tiempo de muestreo, es decir, un sistema tiene un modelo único en t y s , pero z tendrá una ecuación diferente si el tiempo de muestreo es diferente.

En la literatura disponible, revisando diferentes áreas de aplicación de sistemas discretos, se puede observar cómo los diferentes autores coinciden en que el tiempo de muestreo es una de las variables más importantes que influye directamente en los resultados obtenidos. Por ejemplo, para el estudio de la teoría de control clásico es indispensable la identificación de sistemas para determinar las características de la planta a controlar y poder diseñar el control más adecuado, como lo muestran diferentes trabajos para la identificación de sistemas de tiempo continuo a partir de datos muestreados (Tuma y Jura, 2017; Figwer, 2018; Ghoul et al., 2016).

También se encuentran trabajos en los que se demuestra cómo el poder computacional del hardware utilizado y el tiempo de muestreo son factores importantes que limitan el



desarrollo de aplicaciones en tiempo real (Dobrogowski y Kasznia, 2005). En la misma línea, existen análisis sobre las diferencias en los cálculos de algoritmos en un sistema de medición con computadoras personales a diferentes frecuencias de operación de los procesadores (Dobrogowski y Kasznia, 2007). Se han propuesto métodos para mejorar los sistemas de comunicación multicanal, reduciendo el error en el tiempo de medición con una computadora personal (Dobrogowski y Kasznia, 2013). Se ha hecho énfasis en la importancia de la frecuencia de muestreo cuando se desarrollan sistemas de medición con múltiples dispositivos de medición, y en el cuidado que se debe tener debido a los errores de medición y la interpretación de resultados (Vollmer et al., 2019). Se han desarrollado algoritmos para utilizar diferentes tiempos de muestreo y corregir los principales problemas en procesos industriales modernos, dinámicos y no lineales, mejorando así la calidad de los datos adquiridos (Yuan et al., 2021). Esto incluye la detección y corrección de errores en el tiempo de muestreo (Chakravarthi y Mohan, 2017; Leuciuc, 2017; Neitola, 2020; Qi y Chen, 2018; Quian et al., 2019).

Se ha trabajado en el desarrollo de sistemas de adquisición basados en FPGA para obtener mayor velocidad en el tiempo de muestreo (Krishnamoorthy et al., 2018) y se ha evaluado el impacto que tiene el número de muestras y el tiempo de muestreo en el cálculo de variables, como la estimación de densidad de energía eólica (Gross et al., 2020) y el análisis de los efectos del tiempo de muestreo en una red de un control proporcional integral y una planta de segundo orden (Santhosh y Noronha, 2018).

En la ciencia de los datos, hay trabajos que resaltan la importancia del tiempo de muestreo y la velocidad de procesamiento de la información (Bei-Bei y Xue-bo, 2015). Se destaca la importancia de la utilidad de los datos en la medida en que pueden ser rápidamente analizados para revelar información valiosa. Se demuestra que, con datos de alta calidad, podemos aumentar los ingresos, reducir los costos y reducir el riesgo. Se concluye que la técnica de adquisición de muestras de datos es muy importante e impacta en la calidad de estos (Liu et al., 2019).

Como se puede observar en la literatura consultada, el tiempo de muestreo o su equivalente, la frecuencia de muestreo, es la variable fundamental de estudio y análisis para obtener resultados confiables en muchas aplicaciones como adquisición de datos, control de sistemas, ciencia de los datos, entre otras. El principal problema es la determinación del tiempo de muestreo (T), ya que este depende de las características del hardware a utilizar, la velocidad de la unidad central de procesamiento (CPU), la velocidad de los periféricos de

entrada y salida (E/S) y el tamaño del programa a utilizar. Al obtener el tiempo de muestreo, ya podemos hacer uso de la transformada z , la herramienta más utilizada en análisis para describir sistemas en tiempo discreto (Horváth, 2020; Ogata, 1995). Es por ello que el presente trabajo se enfoca en concentrar en un solo documento las herramientas clásicas para el análisis de sistemas en los dominios del tiempo (t), la frecuencia (s) y la frecuencia discreta (z). Además, se presenta un método práctico para determinar el tiempo de muestreo (T) en sistemas con microcontroladores de bajo poder computacional, con el objetivo de desarrollar aplicaciones de monitoreo y/o control. Con este trabajo, se busca facilitar el uso de las matemáticas en los dominios t , s y z , y fomentar su utilización para simplificar el diseño de sistemas de monitoreo y/o control, utilizando simuladores y obteniendo resultados óptimos.

Método

Se utilizaron modelos matemáticos de sistemas en tiempo continuo y en el dominio de Laplace. Como se mencionó anteriormente, el modelo matemático ampliamente utilizado en diversas áreas de análisis de ingeniería es la ecuación diferencial lineal ordinaria con coeficientes constantes (Dobelin, 1985):

$$a_n \frac{d^n q_o}{dt^n} + a_{n-1} \frac{d^{n-1} q_o}{dt^{n-1}} + \dots + a_1 \frac{dq_o}{dt} + a_0 q_o = b_m \frac{d^m q_i}{dt^m} + b_{m-1} \frac{d^{m-1} q_i}{dt^{m-1}} + \dots + b_1 \frac{dq_i}{dt} + b_0 q_i \quad (1)$$

Donde:

$q_o \rightarrow$ Es la variable de salida o respuesta de un sistema físico.

$q_i \rightarrow$ Es la variable de entrada o excitación del sistema físico

t -- Es el tiempo

a_n, b_n Son los parámetros físicos del sistema.

Definiendo el operador diferencial $D = \frac{d}{dt}$ la Ecuación (1) se puede reescribir como (Dobelin 2004):

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

El uso de este modelo es generalizado porque esta directo a soluciones analíticas disponibles sin importar el n orden de la ecuación, permitiendo el tratamiento de sistemas lineales de complejidad arbitraria.

En el análisis, diseño y aplicaciones de sistemas de medición y/o control, el concepto de función de transferencia operacional es muy útil. La función de transferencia operacional relaciona la salida q_o con la entrada q_i y se define como (Dobelin 2004):

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

A partir de la Ecuación (1), se pueden definir los modelos de sistemas dinámicos más comunes: los de orden cero, primer orden y segundo orden (Dobelin, 1985). La Transformada

de Laplace se utiliza habitualmente para el estudio de sistemas lineales. La función de transferencia de Laplace se define como la relación entre la transformada de Laplace de la salida y la transformada de Laplace de la entrada, con condiciones iniciales iguales a cero. Por lo tanto, la Ecuación (3) puede reescribirse como se muestra a continuación (Dobelin, 2004):

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

El modelo de un sistema dinámico de orden cero está definido por la ecuación:

$$a_0 q_o = b_0 q_i \quad (5)$$

La Ecuación (5) de forma estandarizada se escribe como $q_o = K q_i$ donde K es la sensibilidad estática del sistema o ganancia de estado estable definida por $K = \frac{b_0}{a_0}$.

La función de transferencia en Laplace es:

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

El modelo básico de un sistema dinámico de primer orden está definido por la ecuación:

$$a_1 \frac{dq_o}{dt} + a_0 q_o = b_0 q_i \quad (7)$$

Y su forma estandarizada es:

$$\tau \frac{dq_o}{dt} + q_o = K q_i \quad (8)$$

Donde $\tau = \frac{a_1}{a_0}$ es la constante de tiempo y $K = \frac{b_0}{a_0}$ es la ganancia de estado estable.

Y su función de transferencia en Laplace es:

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

El modelo básico de un sistema dinámico de segundo orden está definido por la ecuación:

$$a_2 \frac{d^2 q_o}{dt^2} + a_1 \frac{dq_o}{dt} + a_0 q_o = b_0 q_i \quad (10)$$

EL modelo estándar es de la forma:

$$\frac{1}{w_n^2} \frac{d^2 q_o}{dt^2} + \frac{2\zeta}{w_n} \frac{dq_o}{dt} + q_o = K q_i \quad (11)$$

Donde $w_n = \sqrt{\frac{a_0}{a_2}}$ es la frecuencia natural sin amortiguar, $\zeta = \frac{a_1}{2} \sqrt{\frac{a_2}{a_0}}$ es la razón de amortiguamiento y $K = \frac{b_0}{a_0}$ es la ganancia de estado estable.

Modelos matemáticos de sistemas en tiempo discreto, z

La transformada-z (z) es una herramienta importante en el análisis y diseño de sistemas en tiempo discreto. Juega un rol similar al que cumplen la transformada de Laplace en problemas de tiempo continuo.

La transformada-z unilateral de una función del tiempo $x(t)$, o de la secuencia de valores $x(kT)$, donde $k \geq 0$ y T es el tiempo de muestreo, se define por la siguiente ecuación (Ogata, 1995):

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

La transformada- z unilateral es la utilizada en todas las aplicaciones asociadas con sistemas de tiempo discreto ya que el tiempo de muestreo no está definido para valores negativos.

Muestreo mediante impulsos

Se considera un muestro ficticio comúnmente llamado muestreador mediante impulsos. La salida de este muestreador se considera como un tren de impulsos que inicia en $t = 0$, con el periodo de muestreo igual a T y la magnitud de cada impulso igual al valor muestreado de la señal en tiempo continuo en el instante de muestreo correspondiente. Se supone que $x(t) = 0$ para $t < 0$, entonces la salida muestreada mediante impulsos es una secuencia de impulsos, con la magnitud de cada impulso igual al valor de $x(t)$ en el instante de tiempo correspondiente. Si utilizamos la notación $x^*(t)$ para representar la salida muestreada mediante impulsos, esta señal se puede representar como (Ogata, 1995):

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

o

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

La transformada de Laplace de (13) es:

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

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

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

Si definimos $e^{-Ts} = z$ ó $s = \frac{1}{T} \ln z$ $s = \frac{1}{T} \ln z$, entonces:

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

Como se puede observar el segundo miembro de (12) y (15) son exactamente iguales por lo que podemos expresar:

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

En la literatura disponible es fácil encontrar tablas de funciones en el tiempo y sus correspondientes transformadas en Laplace- s o transformada- z (Dobelin, 1985; Ogata, 1995). Por ejemplo, la función en el tiempo $f(t) = e^{-at}$ su transformada de la Laplace es $F(s) = \frac{1}{s+a}$, su equivalente en muestreo mediante impulsos $f(kT) = e^{-akT}$ y su transformada z de la forma: $F(z) = \frac{1}{1-e^{-aT}z^{-1}}$.

Como se puede observar cualquier función en el tiempo tiene una función única equivalente en Laplace pero tendrá un número infinito de funciones equivalentes en el muestreo mediante impulsos y en la Transformada- z , que dependerá del tiempo de muestreo que se utilice. Es por ello por lo que para utilizar estas herramientas en el diseño y análisis de sistemas es necesario definir correctamente el tiempo de muestreo T .

Análisis de la respuesta de un sistema real de primer orden a una entrada de una función escalón

Para este ejercicio utilizaremos como modelo un circuito básico formado por una fuente voltaje constante V , una resistencia R y un capacitor C (red RC). El voltaje en el capacitor en definido por $v_c(t)$. El interruptor SW se encuentra abierto en $t < 0$ y en $t \geq 0$ se cierra, como se muestra en la Figura 1. La Figura 1a, representa el circuito cuando el $t < 0$, la Figura 1b muestra el circuito para $t \geq 0$, en el momento de cerrar el interruptor aplicamos un voltaje al circuito RC de la forma $Vu(t)$ donde $u(t)$ es el escalón unitario en el que su valor es de cero para $t < 0$ y su valor de 1 cuando $t \geq 0$, como se muestra en la Figura 1c y matemáticamente definido como:

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

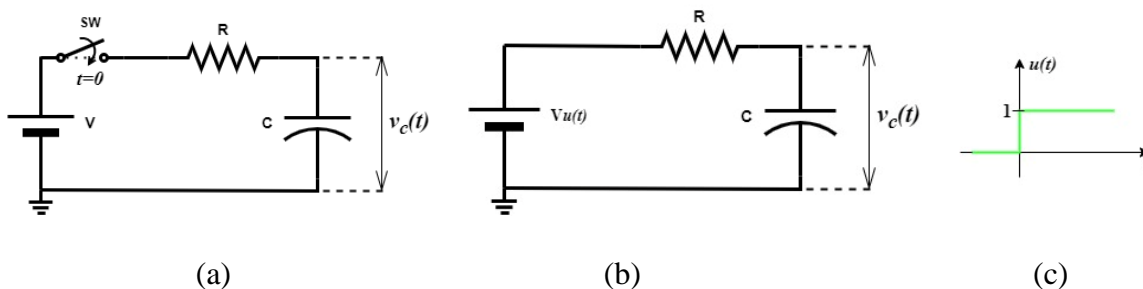
Aplicando la ley de Kirchhoff de corrientes al circuito de la Figura 1c obtenemos la ecuación:

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

o

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

Figura 1.- Circuito RC. a) Red RC en $t < 0$ b) Red RC para $t \geq 0$ y c) Escalón unitario $u(t)$.



Fuente: Elaboración propia

La Ecuación (19) representa un sistema de primer orden de la forma de la Ecuación (7). Ahora para resolver la ecuación, dividiendo (19) entre C:

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

Aplicando la transformada de Laplace a (20) y realizando el cambio de variable $\tau = RC$:

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

Aplicando condiciones iniciales $v_c(0) = 0$

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

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

Utilizando las tablas para antitransformar la Ecuación (23) se obtiene $v_c(t)$ en respuesta al escalón de magnitud V :

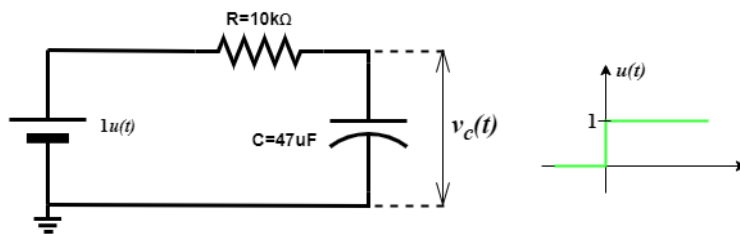
$$v_c(t) = V \left(1 + e^{-t/\tau} \right) \quad (24)$$

Con esto se tiene el modelo del sistema de primer orden en el tiempo definido por la Ecuación (24) y su equivalente en Laplace definido por la Ecuación (23). Utilizando tablas de la transformada z obtenemos el modelo discreto de circuito de la Figura 1:

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

Como se puede ver las Ecuaciones (23) y (24) son modelos únicos en tys para el circuito de la Figura 1, pero analizando la Ecuación (25) se observa que existe un número infinito de ecuaciones en z que depende del valor que se utilice para el tiempo de muestreo T . Analizando el circuito de la Figura 2, obtenemos su respuesta al escalón unitario, por medio de (23) y (24) se obtienen las Ecuaciones (26) y (27). Evaluando (25) para cuatro valores diferentes de tiempo de muestreo ($T = 0.1s, T = 0.5, T = 1s$ y $T = 2s$) se obtienen cuatro ecuaciones en el plano z , como se muestra en (28a) a (28d) respectivamente:

Figura 2.- Circuito RC con entrada escalón unitario.



Fuente: Elaboración propia

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

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

Utilizando (25) para $T = 0.1s$ se obtiene:

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

Utilizando (25) para $T = 0.5s$ se obtiene:

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

Utilizando (25) para $T = 1s$ se obtiene:

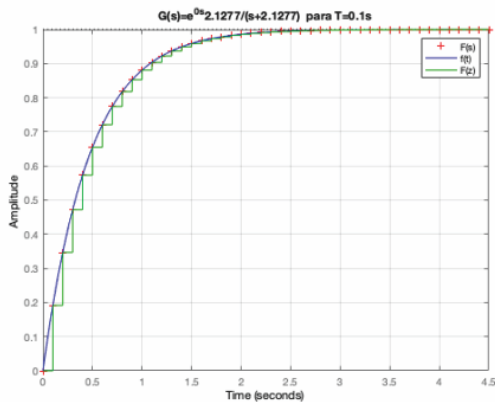
$$V_c(z) = \frac{0.8809}{(z - 0.1191)} \quad (28c)$$

Utilizando (25) para $T = 2s$ se obtiene:

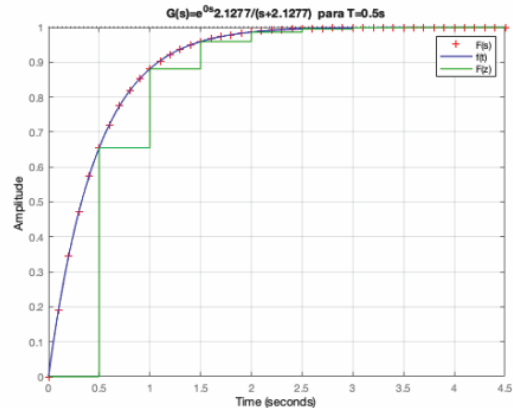
$$V_c(z) = \frac{0.9858}{(z - 0.1419)} \quad (28d)$$

Utilizando MATLAB[®] se simularon las Ecuaciones de la (26) a la (28) y se obtuvo la Figura 3, en la que se puede observar como con un tiempo de muestreo de $T=0.1$ s podemos obtener buena información de la carga del capacitor el análisis de tiempo discreto, Figura 3a. Con un tiempo de muestreo de $T=0.5$ s se pierde información de la carga del capacitor, solo se tienen 4 muestras con respecto a la respuesta en t o s y la primera muestra ya el capacitor llegó a más del 60% de su carga, Figura 3b. Con un tiempo de muestreo de $T=1$ s apenas dos muestras de la carga del capacitor se pueden obtener y como se observa en la Figura 3c, más del 80% de la información se pierde. Con un tiempo de muestreo de $T = 2$ s solo se tiene una muestra y el capacitor ya alcanzó más del 90% de su carga, por lo que no se tiene una buena información de la carga del capacitor Figura 3d.

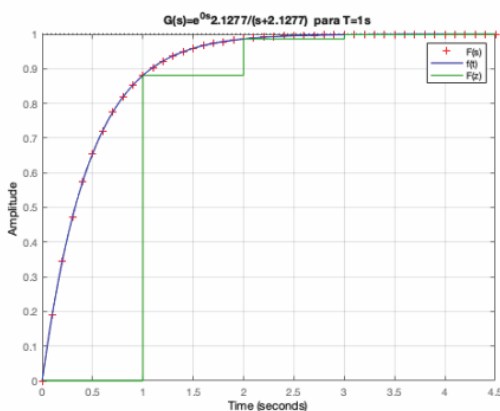
Figura 3.- Voltaje $v(t)$ de la red RC Figura 2. a) $T=0.1$ s, b) $T=0.5$, c) $T=1$ s y d) $T= 2$ s



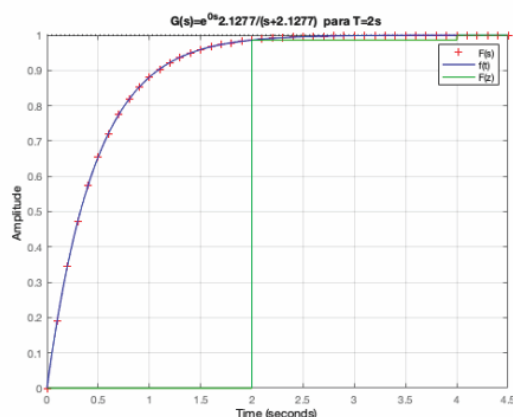
(a)



(b)



(c)



(d)

Fuente: Elaboración propia

Hasta aquí se ha demostrado la importancia de definir correctamente el tiempo de muestreo T para obtener información de calidad. Cuando esta información se obtiene por medio de un sistema de adquisición de datos, la tarea de definir el tiempo de muestreo real no es fácil ya que, en un sistema de adquisición de datos real, T dependerá del hardware utilizado y el software desarrollado. El tiempo de muestreo en un sistema de adquisición de datos T_m se puede definir como sigue:

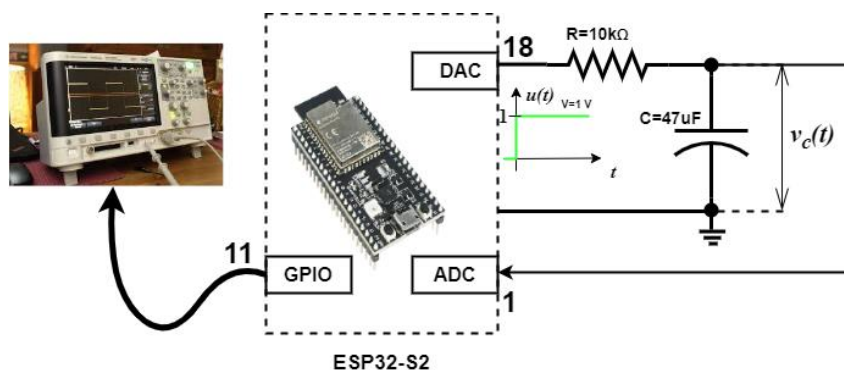
$$T_m = T_H + T_S(29)$$

Donde el T_H es el tiempo empleado por el hardware y el T_S es el tiempo empleado en la ejecución del software

Diseño de experimentos para determinar el tiempo de muestreo en un sistema de adquisición de datos con base a la respuesta a una entrada de una función escalón

Para determinar el T_S es posible utilizar las herramientas de desarrollo disponibles para microcontroladores, en el que se puede analizar el código ensamblador y estimar el tiempo de ejecución de bloques del programa. Pero esta técnica es factible en microcontroladores pequeños y no esta generalizada para microcontroladores más potentes o para todas las marcas. Por lo que proponemos programar una salida digital como indicador en alto (“1” lógico) del inicio de un programa y cambie a bajo (“0” lógico) al terminar el programa y determinar el T_m . Para demostrar este método se utilizó un kit de desarrollo de propósito general el ESP-32-S2 (ADQ) y se armó el circuito de la Figura 4.

Figura 4.- Sistema de adquisición para determinar el tiempo de muestreo, T_m



Fuente: Elaboración propia

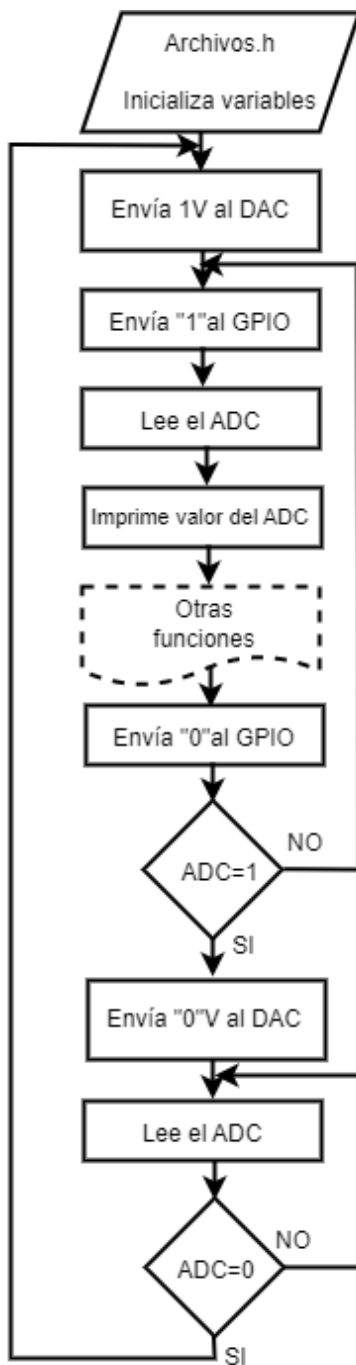
El ESP-32-S2 es una herramienta poderosa para desarrollar proyectos relacionados con Internet de las Cosas, *Internet of Things* (IoT). Sus principales características son (Ozan, 2021):

- CPU y Memoria. Es un microprocesador Xtesa® LX6 de 32-bits con un reloj de 240 Mhz y 600 Millones de Instrucciones por Segundo (MIPS). Cuenta con memoria de 448 KB de ROM, 520 KB de RAM y 16 KB de RTC.
- Periféricos. GPIOs, ADC, DAC, SPI, I²C, I²S, UART, CAN, IR, PWM, Sensor de contacto y sensor de efecto hall.
- Conectividad. Wi-Fi 802.11n (2.4 GHZ) arriba de 150 Mgps.
- Consumo de energía. Ultra bajo consumo de energía, *Ultra-Low-Power* (ULP). 100 μA en modo de suspensión (*sleep mode*).

Existen diferentes versiones del ESP-32, con conectividad Bluetooth, eMMC/SD, para operación en altas temperaturas, entre otras.

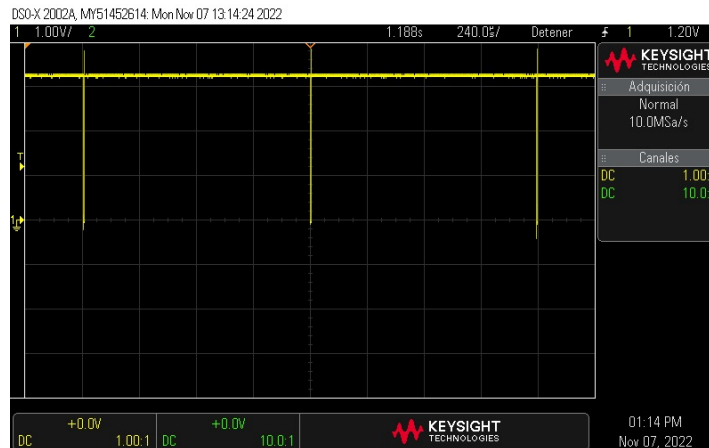
Para determinar el tiempo de muestreo T_m utilizando la plataforma de desarrollo de ECLIPSE se desarrolló un programa en lenguaje C/C++. El diagrama de flujo se muestra en la Figura 5, en el que al principio se cargan los “archivos.h” y se inicializan las variables. Después se envía 1 V por medio del convertidor-digital-análogo (DAC) a la red RC e inmediatamente después un puerto digital se pone en “1” lógico indicando el arranque de la entrada escalón a la red RC. A continuación, se lee el valor del de la carga del capacitor por medio del convertidor-digita-análogo (DAC). Se agrega un bloque llamado “Otras funciones” en la cual el usuario puede programar funciones como, procesar la información, desplegar información, comunicar información, entre otras. Después, se envía un “0” lógico al GPIO para indicar el fin del muestreo a la red RC. Estas instrucciones se repiten hasta que la lectura del ADC tenga el valor de 1 V, que nos indica que el capacitor ya se cargó al valor del escalón enviado a través del DAC. Después de que el capacitor se carga termina esa rutina. Ahora se envían 0 V por medio del DAC a la red RC y se lee continuamente el voltaje del capacitor por medio del DAC hasta que su lectura es de 0 V, lo que indica que el capacitor se terminó de descargar. Luego se repite el programa completo varias veces, para tener la información suficiente y poder hacer mediciones con el osciloscopio. Conectando un osciloscopio a la salida del GPIO se obtiene una señal como se muestra en la Figura 6, esta señal fue obtenida con un osciloscopio KEYSIGHT modelo DSO-X-20002^a.

Figura 5.- Diagrama de flujo del programa desarrollado



Fuente: Elaboración propia

Figura 6.- Señal obtenida en el GPIO del ADQ con un osciloscopio DSO-X-20002A.



Fuente: Elaboración propia

Siguiendo lo presentado anteriormente se programaron 5 experimentos: En el primero, en el bloque “Otras funciones” del diagrama de flujo de la Figura 5 no se programó nada, lo que nos generó la máxima velocidad del ADQ y se obtiene el menor T_m . Después se programaron diferentes retardos en el bloque “Otras funciones” para obtener un tiempo de muestreo de 10ms, 50ms, 0.166s y 0.333s. En la siguiente sección se muestran los resultados obtenidos.

Resultados

Lo primero que se hizo fue medir físicamente el valor de real de la resistencia, R y el capacitor, C utilizado. Los valores obtenidos fueron $R=9.76 \text{ K}\Omega$ y $C=48 \mu\text{F}$. Con esto recalculamos las Ecuaciones (26) y (27) para $\frac{1}{\tau} = 2.135\text{s}^{-1}$:

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

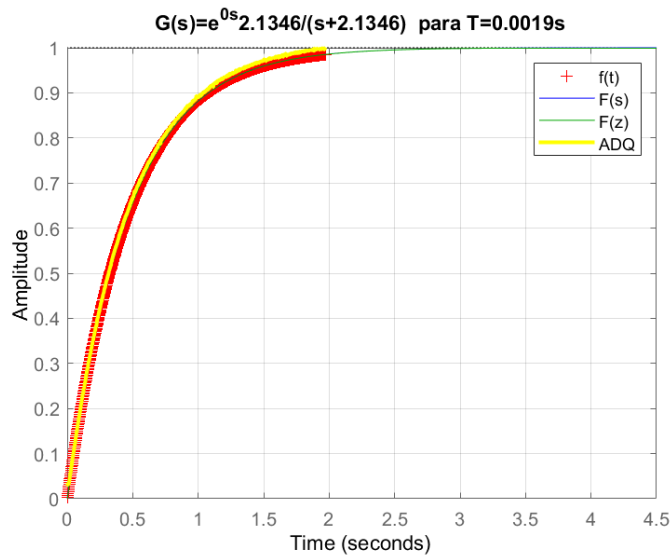
$$V_c(s) = \frac{2.135}{s(s + 2.135)} \quad (31)$$

Midiendo la señal del osciloscopio se obtuvo el $T_m = 1.9 \text{ ms}$, es el tiempo mínimo de muestreo, con lo que se obtuvo que la ecuación equivalente en z .

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

Por medio del programa desarrollado se registraron las lecturas reales obtenidas con el sistema ADQ de la Figura 4. La Figura 7 muestra gráficamente los resultados obtenidos del modelo matemático de la carga de la red RC en t , s , z y el resultado práctico.

Figura 7.- Resultados obtenido por las Ecuaciones (30-32) y el ADQ.



Fuente: Elaboración propia

Las ecuaciones en z para los tiempos de muestreo de 10ms, 50ms, 0.166s y 0.333s son:

Utilizando (25) para $T = 10ms$ se obtiene:

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

Utilizando (25) para $T = 50ms$ se obtiene:

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

Utilizando (25) para $T = 0.166s$ se obtiene:

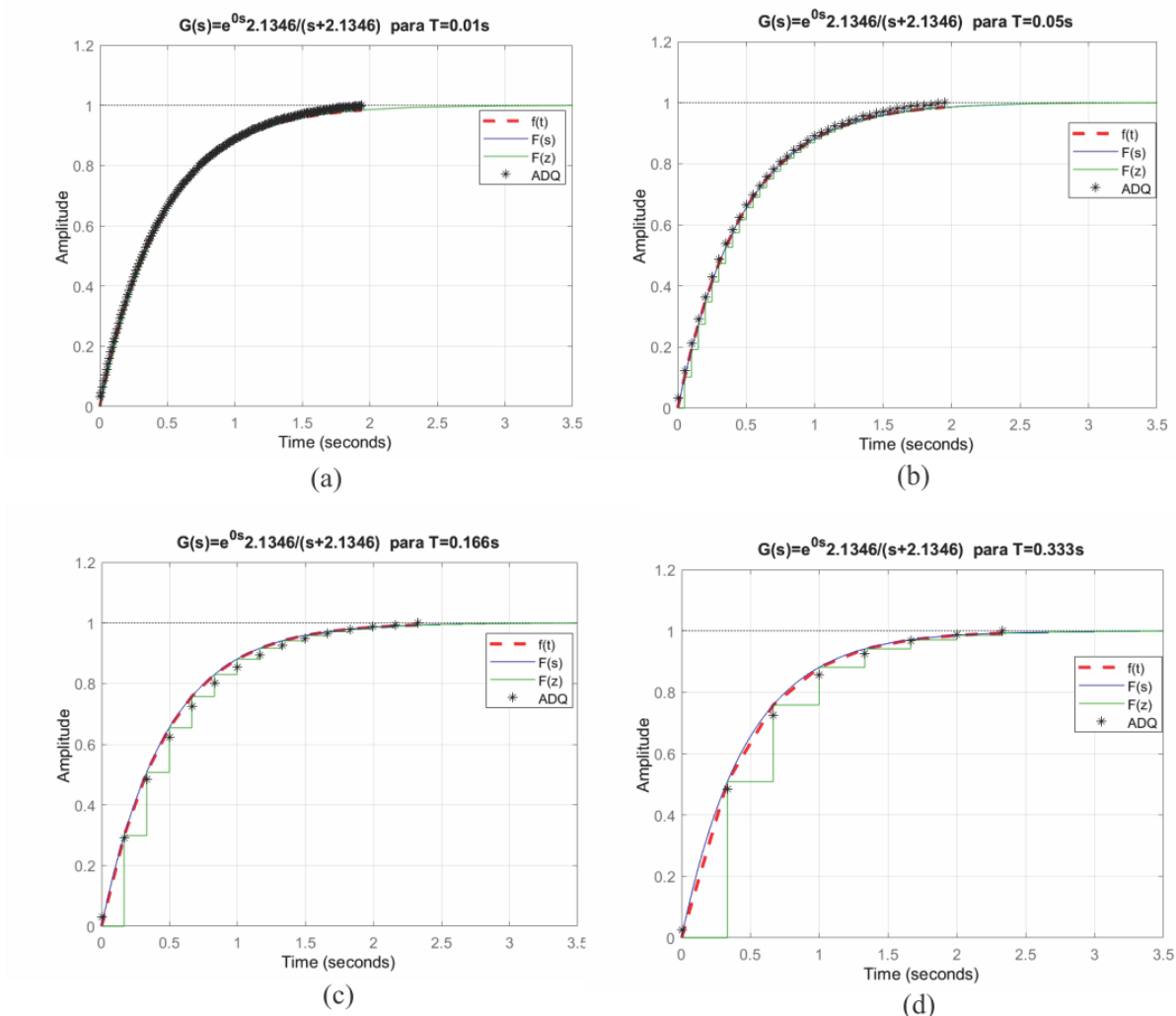
$$V_c(z) = \frac{0.2984}{(z - 0.7016)} \quad (35)$$

Utilizando (25) para $T = 0.333s$ se obtiene:

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

En la Figura 8 se muestran los resultados de la simulación de las Ecuaciones (30,31) y (33-36) junto con los resultados obtenidos prácticamente a los diferentes tiempos de muestreo programados en el ADQ. En cada grafica se muestran los resultados de los modelos en t, s, z y los obtenidos de forma práctica con el 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) $T_m = 10ms$, b) $T_m = 50ms$, c) $T_m = 0.166s$ y d) $T_m = 0.333ms$



Fuente: Elaboración propia

Como se puede observar en la Figura 8 los datos obtenidos prácticamente coinciden con los modelos definidos en t, s, z . Para hacer una declaración cuantitativa de los resultados obtenidos se utilizó la ecuación de la raíz del error cuadrático medio (RMSE) (37). Comparando los resultados calculados con la Ecuación (30) en el dominio del tiempo y los valores obtenidos por el ADQ a los diferentes T_m .

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

Donde $v_{c,i}$ representa el voltaje i en el capacitor obtenido por la Ecuación (30), $v_{ADQ,i}$ representa el voltaje en el capacitor i obtenido con el ADQ de la Figura 4 y N es el número de valores de $v_c(t)$. En la Tabla 1 muestra los resultados. En la Columna 1 se listan los diferentes tiempos de muestreo T_m , en la Columna 2 se muestra la frecuencia de muestreo $f_m = 1/T_m$, en la Columna 3 se muestra el RMSE y en la Columna 4 se muestra por ciento del RMSE.

Con los resultados obtenidos se puede demostrar que el método propuesto presenta una buena exactitud para determinar el T_m de un sistema de ADQ en el que su respuesta se combina entre el hardware utilizado y el programa que se implemente. Los resultados obtenidos de forma práctica son similares a los datos de los modelos t, s, z . Con los resultados obtenidos en la Tabla 1, se puede comprobar que a medida que el T_m es más lento el error se incrementa. Por todo lo anterior se demuestra la importancia de conocer el T_m . De esta manera fácilmente pueden ser aplicadas todas las matemáticas aprendidas en la formación de ingenieros, dando como resultado el diseño de sistemas de monitoreo o control más robustos y eficientes. Esta metodología también facilitará el diseño de sistemas de última generación con el enfoque de Internet de las cosas o la industria 4.0.

Tabla 1. Error de la raíz cuadrática medio entre $v_c(t)$. y los datos obtenidos prácticamente.

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

Fuente: Elaboración propia

Discusión

En el presente trabajo se ha presentado el modelado matemático de sistemas en el tiempo con base a la ecuación ordinaria con coeficientes constantes. Su equivalencia en la transformada de la Laplace y la transformada z . Un ejemplo práctico de estos modelos se muestra en la Figura 1 y Figura 2, una red RC en la que su respuesta a un voltaje de alimentación del tipo escalón unitario se puede representar por la Ecuación (24) en el dominio de t , por la Ecuación (23) en el dominio de s y por la Ecuación (25) en el plano z . También se demostró la importancia del tiempo de muestreo para que exista una buena relación entre los resultados

del plano z y los modelos en tys , como se puede observar en las Ecuaciones (26-28) y los resultados gráficos que se presentan en la Figura 3. Como se puede observar entre menor sea el tiempo de muestreo se obtendrá una mejor información del comportamiento del sistema bajo estudio. Lo que concuerda con lo revisado en la literatura disponible sobre el tiempo de muestreo.

Para mostrar un ejemplo práctico sobre este tipo de modelados y comparar los resultados se utilizó un sistema de adquisición de datos como el que se muestra en la Figura 4. En el que se programó en lenguaje C/C++ el diagrama de flujo de la Figura 5. Con esto se obtuvo el tiempo mínimo de muestreo del ADQ que fue de $T_m = 0.0019s$ o una frecuencia de $f_m = \frac{1}{T_m} = 526.31Hz$, esto representa la mayor velocidad del ADQ ya que en el bloque de “Otras funciones” de la Figura 5 no se programó nada. El modelo teórico obtenido en el plano z , se muestra en la Ecuación (32) y en la Figura 7 se muestra gráficamente los resultados obtenidos prácticamente con el ADQ de la respuesta de la red RC a una entrada escalón unitario y los modelos en t, syz . En los cuales se observa una muy buena concordancia en sus resultados. Esto es por la velocidad del ADQ y que se programó exclusivamente para registrar la lectura del voltaje en el capacitor como lo muestra la Figura 4.

Simulando la programación de otras actividades en el ADQ se programaron diferentes retardos de tiempo en el bloque “Otras funciones” de programa de la Figura 5. Los modelos en tys no cambian, Ecuaciones (30 y 31) y sus equivalentes en z para cuatro diferentes tiempos de muestreo se representan por las Ecuaciones (33-36). Los resultados obtenidos prácticamente y los modelos en t, syz se muestran en la Figura 8. Como se puede observar en las gráficas, existe una concordancia entre los datos obtenidos con el ADQ representados por los asteriscos en la gráfica y la línea verde que representa el modelo en z . Como se puede observar a medida que el tiempo de muestreo es más lento la información registrada de la carga del capacitor se pierde.

Además, se evaluó cuantitativamente los resultados obtenidos prácticamente y el modelo en el tiempo representado por la Ecuación (30) utilizando el RMSE con la Ecuación (37). Obteniendo un error muy pequeño. Y confirmando una buena concordancia en los resultados prácticos y los modelos matemáticos presentados.

El determinar el tiempo de muestreo para el diseño de sistemas de monitoreo y/o control digital es una parte fundamental. Lo aquí presentado coincide con Tuma y Jura (2017), Figwer (2018) y Ghoul et al., (2016) en el que se expone la identificación de la planta a partir de datos muestreados, permitiendo el diseño de controladores más adecuados.

Además, hemos explicado como el agregar más líneas a un programa en sistema de monitoreo y/o control incrementa el tiempo de muestreo, por lo que en algunos trabajos se necesitará de hardware con mayor velocidad en su poder computacional coincidiendo con lo que expresa Dobrogowski y Kasznia, (2005) y Bei-Bei y Xue-bo (2015), que el poder computacional del hardware utilizado limita el desarrollo de aplicaciones en tiempo real. Por lo que aplicando lo aquí presentado ayudará a los diseñadores, en poco tiempo, a probar y seleccionar el hardware más adecuado para sus aplicaciones.

Es importante resaltar que en la actualidad el uso de sistemas de adquisición de datos se ha generalizado y se requiere que estos generen datos de alta calidad ya que son la base de muchas aplicaciones y como menciona Liu *et al.*, (2019), con datos de alta calidad podemos aumentar los ingresos, reducir costos y reducir riesgos. Los sistemas de adquisición de datos son la base del desarrollo de sistemas de monitoreo y/o control por lo que aplicando las herramientas aquí expuestas podemos garantizar el desarrollo de sistemas que generen datos de calidad.

Conclusiones

En el presente trabajo, se presentó el modelado matemático de sistemas basado en la ecuación diferencial lineal ordinaria con coeficientes constantes, que se utiliza en diferentes aplicaciones de la ingeniería. La función de transferencia operacional en el tiempo y en la transformada de Laplace, se mostraron en el plano s . Se expusieron los modelos matemáticos de sistemas de orden cero, primero y segundo. Además, se presentó la teoría de muestreo mediante impulsos y la transformada z , dado que esta teoría es ampliamente utilizada en el diseño y aplicaciones de sistemas discretos, como el monitoreo y/o control digital.

Los sistemas de monitoreo y/o control modelados con la transformada z tienen múltiples representaciones matemáticas que dependen del tiempo de muestreo utilizado. En la práctica, no es sencillo obtener el tiempo de muestreo, ya que este depende del hardware utilizado y el tamaño del software que se programe. Por ello, a menudo no es fácil aplicar los conocimientos adquiridos.

En este trabajo, se mostró una forma práctica de encontrar el tiempo de muestreo de un sistema de adquisición de datos, con el uso de una red RC y un osciloscopio. Se demostró que es posible obtener el tiempo de muestreo con una buena exactitud y que la respuesta de un sistema es semejante en los diferentes modelos matemáticos y en la práctica. Con esto, esperamos que pueda motivar el estudio y aplicación de sistemas de monitoreo y/o control,

y el desarrollo de sistemas más eficientes en los que se aplique lo aprendido en la formación de ingenieros. Además, esta metodología puede ayudar a caracterizar el desempeño de sistemas embebidos y sistemas modernos basados en la filosofía de la Internet de las cosas o la industria 4.0.

Contribuciones a Futuras Líneas de Investigación

El presente trabajo puede contribuir al diseño de algoritmos de control digital basados en la teoría clásica en z . El método expuesto para determinar el tiempo de muestreo puede implementarse fácilmente en cualquier sistema electrónico en el que se desarrolle la aplicación, permitiendo al usuario evaluar si su hardware cumple con los requerimientos del diseño o le permitirá seleccionar el hardware adecuado para su aplicación.

Hoy en día, los sistemas de monitoreo y/o control deben desarrollarse bajo las filosofías de la Internet de las cosas o la industria 4.0, que requieren comunicación con otros sistemas. Por lo tanto, los sistemas de monitoreo y/o control se deben desarrollar teniendo en cuenta los tiempos empleados en la comunicación y análisis de datos. La metodología propuesta puede utilizarse para caracterizar estos sistemas y desarrollar productos más robustos y con mejor desempeño.

En el ámbito educativo, esto permitirá que el estudio de sistemas de monitoreo y/o control sea más completo al relacionar la concordancia entre los modelos matemáticos y la práctica, lo que resultará en programas de estudio mejorados y profesionales mejor preparados.

Referencias

- 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>
- Figuer, 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 Projects 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, Malaysia.
<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