Ciencia Latina Revista Científica Multidisciplinar, Ciudad de México, México.
ISSN 2707-2207 / ISSN 2707-2215 (en línea), enero-febrero 2025,
Volumen 9, Número 1.

https://doi.org/10.37811/cl_rcm.v9i1

RESPUESTA AL IMPULSO PARA LA OBTENCIÓN DE LA FUNCIÓN DE TRANSFERENCIA PULSO DE UN SISTEMA DIGITAL DE PRIMER Y DE SEGUNDO ORDEN.

IMPULSE RESPONSE FOR OBTAINING THE TRANSFER FUNCTION OF A PULSE IN A FIRST-ORDER AND SECOND-ORDER DIGITAL SYSTEM

Francisco Carlos Mejía Alanís
Tecnológico Nacional de México, Instituto Tecnológico de Lázaro Cárdenas

Julio Cesar Gallo Sánchez
Tecnológico Nacional de México, Instituto Tecnológico de Lázaro Cárdenas

DOI: https://doi.org/10.37811/cl_rcm.v9i1.15746

Respuesta al impulso para la obtención de la función de transferencia pulso de un sistema digital de primer y de segundo orden.

Francisco Carlos Mejía Alanís[1]
fcarlos.malanis@lcardenas.tecnm.mx
https://orcid.org/0000-0001-9257-4009
Tecnológico Nacional de México, Instituto Tecnológico de Lázaro Cárdenas
México

Julio Cesar Gallo Sánchez
jcesar.cesar@lcardenas.tecnm.mx
https://orcid.org/0000-0003-1034-1610
Tecnológico Nacional de México, Instituto Tecnológico de Lázaro Cárdenas
México

RESUMEN

El presente trabajo se centra en la implementación de una estrategia para obtener las funciones de transferencia en el dominio del tiempo discreto de sistemas de primer orden y sistemas de segundo orden en tiempo continuo. La metodología empleada se basa en la simulación de la respuesta al impulso de estos sistemas continuos y muestreando su respuesta con diferentes valores del tiempo de muestreo para obtener diferentes funciones de transferencia pulso de los mismos sistemas, lo cual permite comprender cómo la discretización afecta la respuesta del sistema y cómo se puede utilizar la respuesta al impulso para derivar las funciones de transferencia pulso, que son esenciales para el diseño y análisis de sistemas de control digital y filtros. Ademas, estas funciones de transferencia pulso permiten realizar un analisis de las características de los sistemas discretos, como la estabilidad, el amortiguamiento y el tiempo de respuesta.

Para llevar a cabo este análisis, se utiliza Python como herramienta de simulación, generando las respuestas temporales de los sistemas discretos ante distintas entradas, tales como las señales escalón y rampa. Estas respuestas permiten observar cómo evoluciona la salida del sistema en función del tiempo, lo que proporciona información clave sobre las características dinámicas de los sistemas de primer y segundo orden en su forma discreta.

Además, el trabajo presenta una comparación entre las respuestas obtenidas para los sistemas discretos y las de sus versiones continuas correspondientes. Las respuestas de ambos sistemas, digitales y continuos, se simulan también en Python y se contrastan, lo que permite evaluar las diferencias y similitudes en el comportamiento de ambos tipos de sistemas bajo las mismas condiciones de entrada.

Palabras clave: función de transferencia pulso, sistema en tiempo discreto, respuesta al impulso, entrada escalón y rampa

[1] Autor principal.
Correspondencia: fcarlos.malanis@lcardenas.tecnm.mx

Impulse response for obtaining the transfer function of a pulse in a first-order and second-order digital system

ABSTRACT

This work focuses on the implementation of a strategy to obtain the transfer functions in the discrete-time domain for first-order systems and second-order continuous-time systems. The methodology used is based on simulating the impulse response of these continuous systems and sampling their response with different sampling times to obtain different pulse transfer functions of the same systems. This approach allows for understanding how discretization affects the system's response and how the impulse response can be used to derive pulse transfer functions, which are essential for the design and analysis of digital control systems and filters. Furthermore, these pulse transfer functions enable the analysis of the characteristics of discrete systems, such as stability, damping, and response time.

To carry out this analysis, Python is used as a simulation tool, generating the time-domain responses of the discrete systems to different inputs, such as step and ramp signals. These responses allow for observing how the system's output evolves over time, providing key information about the dynamic characteristics of first- and second-order systems in their discrete form.

Additionally, the work presents a comparison between the responses obtained for the discrete systems and those of their corresponding continuous versions. The responses of both digital and continuous systems are also simulated in Python and contrasted, allowing for the evaluation of the differences and similarities in the behavior of both types of systems under the same input conditions..

***Keywords*:** pulse transfer function, discrete-time system, impulse response, step input and ramp

Artículo recibido 19 noviembre 2024

Aceptado para publicación: 24 diciembre 2024

INTRODUCCIÓN

El análisis de respuesta temporal tiene un papel clave en los sistemas de control para estudiar el desempeño de un sistema (*Analysis of Transient Response of First & Second Order System Using EXPEYES*, n.d.). Por este motivo es que el análisis de la respuesta transitoria y permanente de los sistemas de primer y segundo orden es un tema fundamental en la asignatura de control digital o sistemas de control en tiempo discreto. Estos sistemas son ampliamente utilizados en diversas aplicaciones, desde la automatización industrial hasta el procesamiento de señales. Comprender cómo responden estos sistemas a diferentes entradas es crucial para el diseño eficiente de controladores y filtros digitales. Uno de los aspectos más relevantes en este campo es el diseño de controladores o compensadores digitales. Aunque estos controladores son implementados en hardware digital, su diseño a menudo se realiza en el dominio del tiempo continuo. Posteriormente, se aplican técnicas de discretización para adaptarlos al entorno digital (Rabbath & Léchevin, 2014). Este enfoque permite aprovechar las herramientas analíticas desarrolladas para sistemas continuos, asegurando que las características del controlador se mantengan al ser llevadas al dominio discreto. El diseño de filtros digitales también es un componente crítico dentro de esta disciplina. La discretización del filtro analógico previamente diseñado es esencial para garantizar que el filtro mantenga su efectividad en el dominio discreto (Pérez et al., 2018). Asimismo, es común realizar el diseño de controladores en el dominio de tiempo discreto. Por lo tanto, es necesario realizar la transformación de un sistema en tiempo continuo a sistema en tiempo discreto (Ádám et al., 2003). Para lograr esto, se utilizan diversas técnicas conocidas como transformaciones s a z (Franklin et al., 1998). Estas técnicas incluyen la respuesta al impulso, la respuesta al escalón (Chen & Francis, 1995), diferencias hacia atrás o hacia adelante, y la transformada bilineal o Tustin (Vadhavkar et al., 2007). También, se tiene el método basado en la teoría del control H^{∞} de datos muestreados (Nagahara & Yamamoto, 2013). En este método se formula el problema de discretización como la minimización de la norma H^{∞} del sistema de error entre un filtro analógico objetivo con retardo y un sistema digital que incluye un muestreador ideal, un retentor de orden cero y un filtro digital. El problema se reduce a la optimización $h\infty$ en tiempo discreto mediante el método de aproximación rápida de muestreo/retención. Por otra parte, se tiene la técnica de identificación paramétrica por mínimos cuadrados para obtener la función de transferencia discreta (Diaz, 2002).Cada una de estas metodologías

busca obtener un equivalente en tiempo discreto que represente adecuadamente la función de transferencia del sistema en tiempo continuo. Sin embargo, a medida que aumenta el orden de los sistemas, la complejidad para realizar estas transformaciones manualmente se incrementa. Esto se debe a que las transformaciones s a z de orden superior requieren un conocimiento profundo sobre los comportamientos dinámicos del sistema y las interacciones entre sus componentes. Muchos estudios que abordan el método de la respuesta al impulso mencionan su utilidad para obtener la función de transferencia de un sistema; sin embargo, frecuentemente omiten detallar los pasos necesarios para su implementación. La implementación práctica de estas teorías se realiza comúnmente utilizando herramientas computacionales como Python (*Control.Matlab.C2d — Python Control Systems Library 0.10.1 Documentation*, n.d.) o MATLAB (*C2d*, n.d.). Por ejemplo, MATLAB y Python Control Systems Library ofrecen el comando c2d para convertir funciones de transferencia continuas a discretas especificando un tiempo de muestreo T_s adecuado. Esta capacidad permite validar teorías y observar cómo las características dinámicas cambian con diferentes configuraciones y tiempos de muestreo. En este trabajo se presenta una metodología que utiliza Python para simular la respuesta al impulso y obtener funciones de transferencia pulso para sistemas de primer y segundo orden. Se generan respuestas temporales ante distintas entradas (escalón y rampa), lo que proporciona información clave sobre las características dinámicas del sistema en su forma discreta. Además, se realiza una comparación entre las respuestas obtenidas para los sistemas discretos y sus versiones continuas correspondientes. Este análisis no solo contribuye a una mejor comprensión teórica del comportamiento de los sistemas discretos, sino que también proporciona herramientas prácticas para el diseño y análisis eficiente en aplicaciones reales.

Muestreo de señales analógicas

En el contexto de un sistema físico, una señal contiene la información que describe el comportamiento o la naturaleza de un fenómeno. Por otro lado, en el contexto matemático, una señal se define como una función de una o más variables. Típicamente se tienen dos tipos de señales: señales de tiempo continuo o analógicas e(t), las cuales son definidas para todo instante de tiempo y las señales de tiempo discreto $e(nT_s)$, las cuales son definidas solo en los instantes de tiempo $t=nT_s$, siendo n un número entero y T_s el período de muestreo. Una señal en tiempo discreto puede representarse especificando una regla para

calcular el n-esimo valor de la secuencia $\{x_n\} = \left\{1, \frac{1}{2}, \frac{1}{4}, \ldots, \left(\frac{1}{2}\right)^n, \ldots\right\}$ o enumerando de manera explícita los valores de la secuencia $\{x_n\} = \left\{\ldots, 0, 0, 1, \underset{\smile}{2}, 2, 1, 0, 1, 0, 2, 0, 0, \ldots\right\}$. En la representación de una señal discreta mediante una secuencia, se usa una flecha o un distintivo para denotar el termino n=0. Si no se indica la flecha o el distintivo entonces el primer elemento es el termino n=0 y todos los valores de la secuencia son 0 para n<0 (Hsu et al., 2013).

Las señales en tiempo discreto son obtenidas mediante el proceso de muestreo de una señal continua en el tiempo, a intervalos de tiempo T_s. Esta señal discreta, vista como una secuencia x[n], puede describirse como (Esparza Arellano María Elena & Avalos Briseño J. Benito, 2003):

$$x[nT_s] = \sum_{k=-\infty}^{\infty} x[kT_s]\delta[nT_s - kT_s] \qquad (1)$$

En la ecuación (1), $x[kT_s]$ describe la señal analógica evaluada en los instantes de tiempo kT_s y la función delta de Dirac $\delta[nT_s - kT_s]$ representa la función impulso desplazada kT_s instantes de tiempo.

Una característica a tomar en cuenta en el proceso de muestreo es que se presenta una pérdida de información en la señal x(t) (Pastor et al., 2004). Para evitar esto, el teorema del muestreo establece que una señal x(t) con un espectro limitado a la frecuencia f_B ($|f| \leq f_B$) puede ser muestreada sin pérdida de información si la frecuencia de muestreo $f_S = 1/T_s$ supera la cantidad $2f_B$, es decir $f_S \geq 2f_B$ (Alexander Cortés Osorio et al., n.d.). Si no se muestrea como mínimo a esa frecuencia tiene lugar el fenómeno denominado “aliasing”. Sin embargo, en la práctica ω se elige como 6ω o 10ω, o se escoge un período máximo de muestreo que sea entre dos y tres veces el tiempo de subida (Pastor et al., 2004).

Continuando con el proceso de muestreo, la aplicación de la transformada de Laplace a la ecuación (1) , nos conduce a la expresión siguiente:

$$x^*[s] = \sum_{k=-\infty}^{\infty} x[kT]e^{-kTs} \qquad (2)$$

Donde $x^*(s)$ es la transformada de Laplace de la señal muestreada *x[nT]*. Además, el reordenamiento de la ecuación (2), proporciona la expresión siguiente:

$$x^*[s] = \sum_{k=-\infty}^{\infty} x[kT]\left(e^{Ts}\right)^{-k} \qquad (3)$$

En la ecuación (3), si se realiza el cambio de variable dado por z= e^{Ts}, se tiene la definición de la transformada Z bilateral, como se muestra en la expresión siguiente:

$$X[z]=\sum_{k=-\infty}^{\infty} x[kT]z^{-k} \tag{4}$$

Si las muestras para k<0 son cero, entonces se tiene la transformada Z unilateral de una función discreta, descrita mediante la expresión:

$$X[z]=\sum_{k=0}^{\infty} x[kT]z^{-k}=x[0]z^{0}+x[1]z^{(-1)}+x[2]z^{(-2)}+\ldots \tag{5}$$

La transformada Z de una secuencia discreta x[n] es convergente solo para un conjunto de valores del plano complejo z denominado región de convergencia ROC. La condición de ROC está dada por la ecuación:

$$\sum_{k=0}^{\infty} r^{k}=\frac{1}{1-r}\leftrightarrow|r|<1 \tag{6}$$

Retenedor de orden cero

El retenedor más elemental y utilizado que convierte la señal muestreada en una señal que es constante entre dos instantes de muestreo consecutivos es el retenedor de orden cero. La exactitud del retenedor de orden cero en la reconstrucción de la señal depende de la magnitud del periodo de muestreo T_s. La salida del retenedor de orden cero se ve como una serie de pasos escalonados, donde cada escalón tiene una duración igual al período de muestreo. La función de transferencia del retenedor de orden cero en el dominio de Laplace es

$$H(s)=\frac{1-e^{-sT}}{s} \tag{7}$$

donde T es el período de muestreo.

Función de transferencia pulso

La función de transferencia pulso es la relación de la transformada Z de la salida del sistema y la transformada Z de la entrada del sistema como se representa en la siguiente expresión:

$$G(z)=\frac{C(z)}{R(z)} \tag{8}$$

Debe tenerse en cuenta que como la transformada Z depende del periodo de muestreo, entonces un mismo sistema en tiempo continuo puede tener diferentes representaciones de su correspondiente función de transferencia pulso debido a la elección del periodo de muestreo. Como se describió previamente, existen varios métodos para obtener la función de transferencia pulso de un sistema en tiempo continuo, uno de los métodos es mediante la respuesta al impulso. Debido a que la función de transferencia de un sistema está dada por la ecuación:

$$G(s)=\frac{C(s)}{R(s)} \tag{9}$$

Entonces, de la ecuación (9), se tiene que la salida del sistema queda establecida por la ecuación:

$$C(s)=G(s)\cdot R(s) \tag{10}$$

En este método se tiene que como la transformada de Laplace de la función impulso es la unidad, entonces la función de transferencia del sistema queda determinada por la salida del sistema, como se establece en la expresión:

$$C(s)=G(s) \tag{11}$$

La función de transferencia de la ecuación (11) se le debe aplicar la transformada inversa de Laplace para obtener la expresión en el dominio temporal, como se muestra en la ecuación (12), y poder obtener la función de transferencia pulso.

$$C(t) = G(t) \tag{12}$$

Finalmente, si se aplica la transformada Z a la ecuación (11) se podría obtener la función de transferencia pulso del sistema, mediante la siguiente expresión:

$$G[z] = \sum_{k=0}^{\infty} c[kT] z^{-k} = c[0]z^{0} + c[1]z^{(-1)} + c[2]z^{(-2)} + \ldots \tag{13}$$

Hay que tener en cuenta que la ecuación (13) debe ser dividida por la suma de los coeficientes de la respuesta al impulso c[kT] para que G[z] corresponda con la respuesta al impulso unitario y la función de transferencia analógica debe estar en su forma estándar para un sistema de primer orden y segundo orden como se ilustra en las ecuaciones siguientes:

$$G(s) = K \frac{1}{Ts + 1} \tag{14}$$

$$G(s) = K \frac{\omega_n^{\,2}}{s^2 + 2\xi\omega_n s + \omega_n^{\,2}} \tag{15}$$

METODOLOGÍA

Obtención de la función de transferencia pulso de un sistema de primer orden mediante la respuesta al impulso

El primer paso para obtener la función de transferencia pulso de la función de transferencia de un sistema en tiempo continuo es obtener la respuesta al impulso de dicho sistema. Esta respuesta al impulso del sistema de primer orden es obtenida mediante el código en Python mostrado en la figura 1. La función de transferencia del sistema de primer orden es expresada por la ecuación:

$$G(s) = \frac{1}{0.1s + 1} \tag{16}$$

La grafica de la respuesta al impulso del sistema de primer orden, representado por la ecuación (16), es mostrada en la figura 2. En ella puede observarse que la curva de respuesta alcanza su valor en estado estacionario a los 0.6 segundos aproximadamente.

Figura 1. Código en Python para obtener la respuesta al impulso de un sistema de primer orden

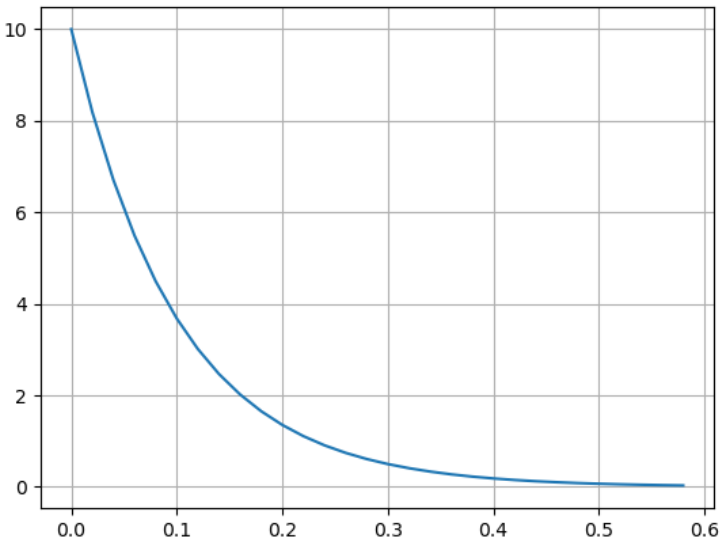
```python
# Obtiene la respusta al impulso de un sistema de segundo orden
import numpy as np
import control
from control.matlab import *
from matplotlib import pyplot as plt
import sympy as sp
x = sp.Symbol('z')
Gz = sp.Symbol('Gz')

# Funcion de transferencia de la planta
num = [1]
den = [0.1, 1]
sysGp = TransferFunction(num, den)  # Func. Tranf. del controlador
print(sysGp)

# Genera el vector tiempo
Ts = 0.02  # tiempo de muestreo
ts = 0.6   # Tiempo de asentamiento
t = np.arange(0,ts,Ts)
l = len(t)

# Respuesta al impulso
c_nT,yy = impulse(sysGp,t)
plt.plot(yy, c_nT)
plt.grid()
plt.show()
```

Figura 2. Respuesta al impulso de un sistema de primer orden



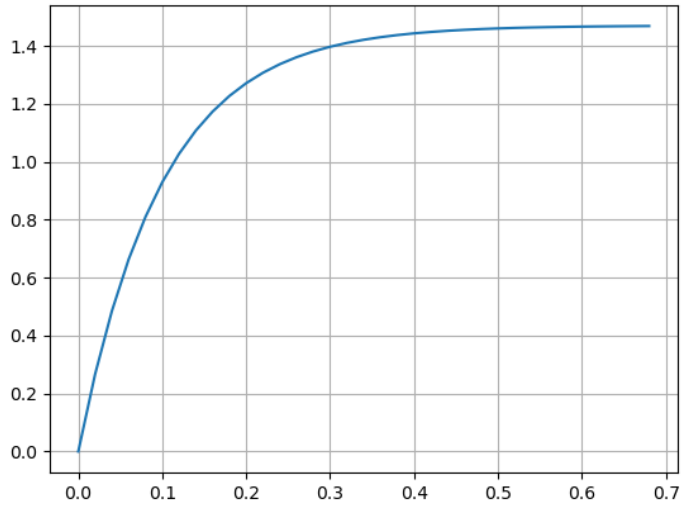
Además, de acuerdo a (Pastor et al., 2004), se tiene que el tiempo de muestreo puede ser tres veces más pequeño que el tiempo de subida, por consiguiente, se simula la respuesta al escalón del sistema de la ecuación (16), como se ilustra en el código de la figura 3, para obtener el tiempo de subida.

Figura 3. Código en Python para obtener la respuesta al escalón de un sistema de primer orden

```
# Respuesta al escalon
plt.figure('Respuesta al escalon del sistema en tiempo continuo')
c,yy=step(sysGp/0.68,t)
plt.plot(yy, c)
plt.grid()
plt.show()
```

Conforme a la respuesta al escalón de la figura 4, se puede determinar que el tiempo de subida es de 6 segundos, por lo que el tiempo de muestreo podría ser de 2 segundos.

Figura 4. Respuesta al escalón de un sistema de primer orden



No obstante, se eligió un periodo de muestreo de 0.02 segundos para realizar la simulación de la respuesta al impulso del sistema de la ecuación (19), la cual arrojó la secuencia c[nT] = [10, 8.18731, 6.7032, 5.48812, 4.49329, 3.67879, 3.01194, 2.46597, 2.01897, 1.65299, 1.35335, 1.10803, 0.90718, 0.742736, 0.608101, 0.497871, 0.407622, 0.333733, 0.273237, 0.223708, 0.183156, 0.149956, 0.122773, 0.100518, 0.0822975, 0.0673795, 0.0551656, 0.0451658, 0.0369786, 0.0302755]. Esta secuencia posteriormente es dividida por la suma de todos los valores del vector c[nT] para que la función de transferencia pulso corresponda con la respuesta al impulso unitario de un sistema discreto. Por tanto, la función de transferencia pulso queda determinada por la transformada Z, indicada por la ecuación (5), de la secuencia c[nT] multiplicada por el factor previamente mencionado, como se describe en la ecuación siguiente:

$$G(z)=\frac{1}{sum(c[nT])}\Im\{c[nT]\} \tag{17}$$

Obtención de la función de transferencia pulso de un sistema de segundo orden mediante la respuesta al impulso

Para obtener la función de transferencia pulso del sistema en tiempo continúo descrito por la ecuación:

$$G(s)=\frac{1}{s^2+0.4s+0.68} \tag{18}$$

es necesario reescribir la función de transferencia de segundo orden en su forma estándar, como se muestra a continuación

$$G(s)=\frac{0.68}{s^2+0.4s+0.68} \tag{19}$$

Posteriormente, se simula la respuesta al impulso en Python, mediante el código mostrado en la figura 5, del sistema de segundo orden dado por la ecuación (19).

Figura 5. Código en Python para simular la respuesta al impulso de un sistema de segundo orden

```
# Obtiene la respusta al impulso de un sistema de segundo orden
import numpy as np
import control
from control.matlab import *
from matplotlib import pyplot as plt
import sympy as sp
x = sp.Symbol('z')
Gz = sp.Symbol('Gz')


# Funcion de transferencia de la planta
num = [0.68]
den = [1, 0.4, 0.68]
sysGp = TransferFunction(num, den)  # Func. Tranf. del controlador
print(sysGp)

# Genera el vector tiempo
Ts = 0.8  # tiempo de muestreo
ts = 30   # Tiempo de asentamiento
t = np.arange(0,ts,Ts)
l = len(t)

# Respuesta al impulso
c_nT,yy = impulse(sysGp,t)
plt.plot(yy, c_nT)
plt.grid()
plt.show()
```

El resultado de la simulación es mostrado en la gráfica de la figura 6, en la que se puede observar que la salida alcanza su valor en estado estacionario en aproximadamente 30 segundos. Pero, de acuerdo a

(Pastor et al., 2004), el tiempo de muestreo puede ser tres veces más pequeño que el tiempo de subida, por lo tanto, también se simula la respuesta al escalón del sistema de la ecuación (19), como se ilustra en el código de la figura 7, para obtener el tiempo de subida. De acuerdo a la respuesta escalón de la figura 8, el tiempo de subida es de 2.28 segundos, por lo que el tiempo de muestreo podría ser de 0.76 segundos.

Figura 6. Respuesta al impulso del sistema de segundo orden

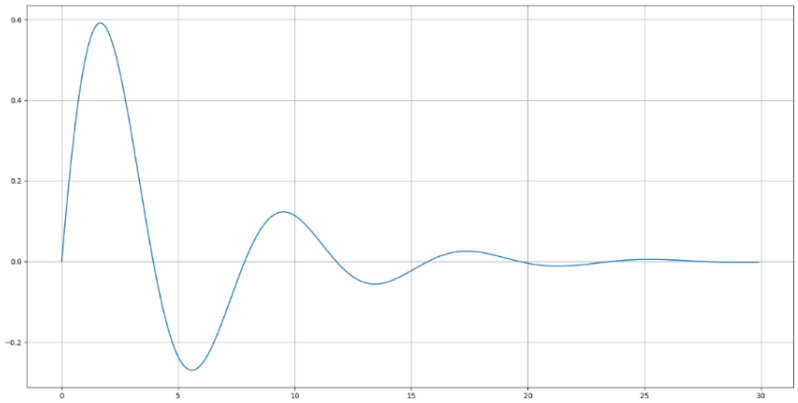
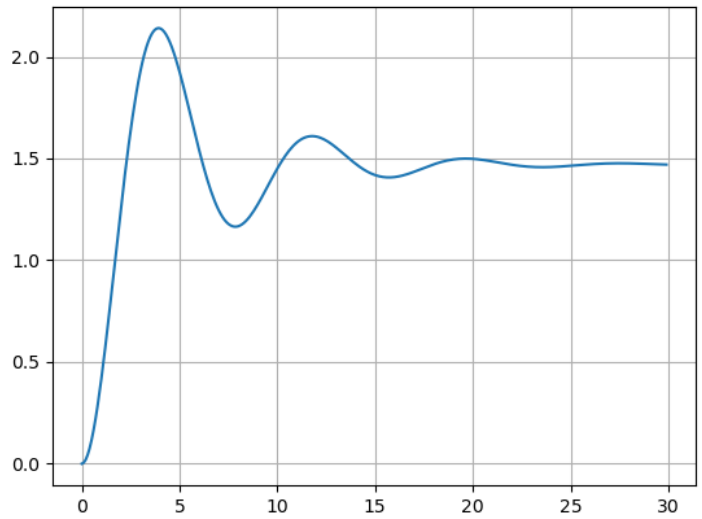


Figura 7. Código en Python para simular la respuesta al escalón de un sistema de segundo orden

```
# Respuesta al escalon
plt.figure('Respuesta al escalon del sistema en tiempo continuo')
c,yy=step(sysGp/0.68,t)
plt.plot(yy, c)
plt.grid()
plt.show()
```

Figura 8. Respuesta al escalón del sistema de segundo orden.



Tomando un tiempo de muestreo de 0.8 segundos en la simulación de la respuesta al impulso del sistema de la ecuación (19), se tiene que la secuencia c[nT] = [0, 0.432562, 0.591313, 0.494221, 0.24622, -

0.0222948, -0.20927, -0.269883, -0.216969, -0.100622, 0.0200012, 0.100408, 0.122735, 0.094867, 0.0405597, -0.0134424, -0.0478281, -0.05562, -0.0413023, -0.0160719, 0.00802132, 0.0226357, 0.0251184, 0.0179, 0.00622963, -0.00448216, -0.0106508, -0.0113049, -0.00771977, -0.0023439, 0.00240159, 0.00498499, 0.00507059, 0.00331165, 0.000845033, -0.00124959, -0.00232181, -0.00226653]. Esta secuencia posteriormente es dividida por la suma de todos los valores del vector c[nT], para que la función de transferencia pulso corresponda con la respuesta al impulso unitario del sistema discreto. Posteriormente, debe multiplicarse por el factor 1/0.68, debido a que la función de transferencia original, representada por la ecuación (18), fue transformada a la función de transferencia en su forma estándar, representada por la ecuación (19). Por tanto, la función de transferencia pulso queda determinada por la transformada Z, indicada por la ecuación (5), de la secuencia c[nT] multiplicada por los factores previamente mencionados, como se describe en la ecuación siguiente:

$$G(z)=\left[\frac{1}{0.68}\right]\left[\frac{1}{sum(c[nT])}\right]\Im\{c[nT]\} \tag{20}$$

RESULTADOS Y DISCUSIÓN

Respuesta escalón y rampa de la función de transferencia pulso obtenida del sistema de primer y segundo orden

Para contrastar la respuesta del sistema en tiempo continuo con el sistema en tiempo discreto obtenido se presenta la respuesta al escalón y a la rampa. En este sentido, en la figura 9 se muestra el código Python para obtener la respuesta al escalón del sistema discreto de primer orden obtenido mediante la respuesta al impulso. Adicionalmente, en la figura 10 se puede visualizar la respuesta del sistema discreto de primer orden ante la entrada escalón. Con este resultado es posible comprobar que, con respecto a la entrada escalón, las respuestas de los sistemas de primer orden en tiempo continuo, figura 4, y de primer orden en tiempo discreto, figura 10, son muy parecidas.

Figura 9. Código Python para obtener la respuesta al escalón del sistema discreto de primer orden.

```
# Configura la entrada del sistema
sel = 1            # escalon:1; rampa:0
A = 1;
if sel == 1:
    R = A*np.ones(l)
    plt.figure('Entrada escalon')
else:
    R = t;
    plt.figure('Entrada impulso')
plt.plot(t,R,'*')
plt.grid()
plt.show()

# Respuesta de la funcion de transferencia pulso
s=sum(c_nT)
Ct = np.ones(l)
for n in range(l):
    aux=0
    for k in range(l):
        ind = n-(k)
        if ind<0:
            r = 0
        else:
            r = R[ind]

        aux = r*c_nT[k]/s/0.68+aux

    Ct[n] = aux
plt.figure('Respuesta del sistema en tiempo discreto')
plt.stem(t, Ct)
plt.grid()
plt.show()
```

Figura 10. Respuesta al escalón del sistema de primer orden.

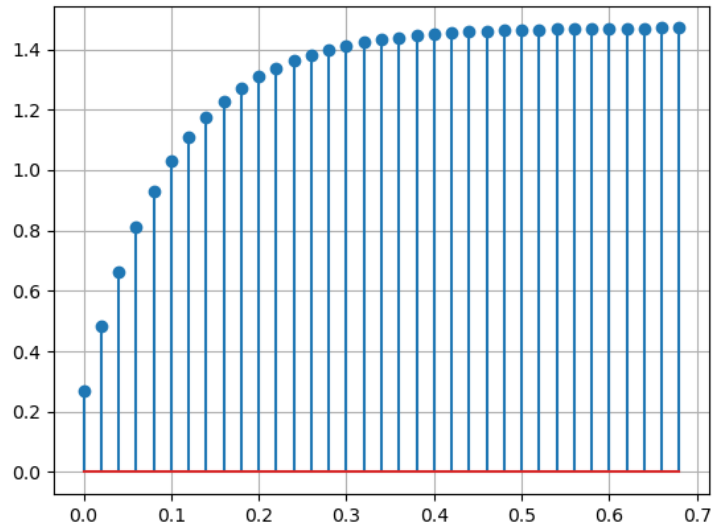


Figura 11. Código Python para obtener la respuesta al escalón del sistema discreto de segundo orden.

```
# Configura la entrada del sistema
sel = 1;            # escalon:1; rampa:0
A = 1;
if sel == 1:
    R = A*np.ones(l)
    plt.figure('Entrada escalon')
else:
    R = t;
    plt.figure('Entrada impulso')
plt.plot(t,R,'*')
plt.grid()
plt.show()

# Respuesta de la funcion de transferencia pulso
s=sum(c_nT)
Ct = np.ones(l)
for n in range(l):
    aux=0
    for k in range(l):
        ind = n-(k)
        if ind<0:
            r = 0
        else:
            r = R[ind]

        aux = r*c_nT[k]/s/0.68+aux

    Ct[n] = aux
plt.figure('Respuesta del sistema en tiempo discreto')
plt.stem(t, Ct)
plt.grid()
plt.show()
```

Figura 12. Respuesta al escalón del sistema discreto de segundo orden.

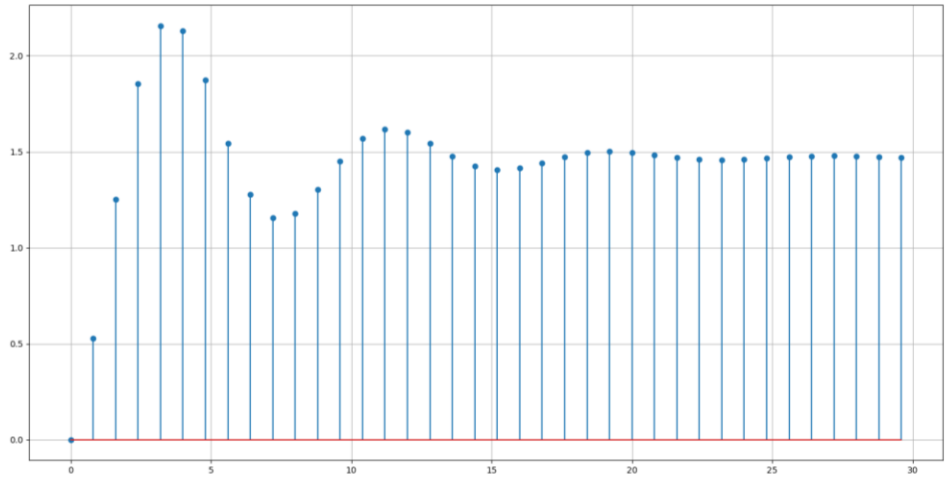
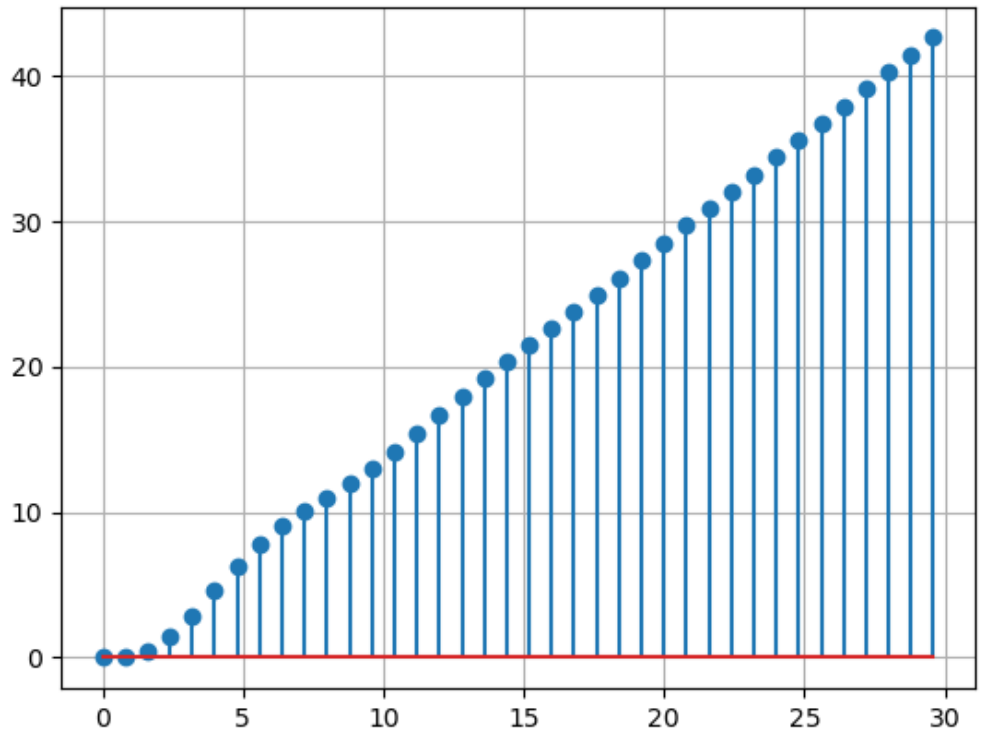


Figura 13. Respuesta a la rampa del sistema discreto de segundo orden.



CONCLUSIONES

En la actualidad, los sistemas de control se implementan principalmente en hardware digital, lo que ha llevado a una creciente necesidad de discretizar los controladores diseñados en el dominio del tiempo continuo. Este proceso de discretización se realiza generalmente al finalizar la fase de diseño de los sistemas, utilizando técnicas estándar que permiten la conversión de sistemas continuos a su versión digital. Esta transformación es fundamental para implementar los controladores en sistemas computacionales, ya que el hardware digital opera de forma discreta y no continua. Esta práctica es común en el campo de la ingeniería de control, donde los sistemas continuos deben ser adaptados a plataformas digitales para su implementación efectiva. Además, con frecuencia los sistemas continuos son diseñados para ser posteriormente discretizados y utilizados como filtros digitales. En la materia de control digital, es esencial realizar un análisis exhaustivo de la respuesta de los sistemas digitales, ya que su comportamiento puede diferir significativamente de los sistemas continuos debido a la discretización. Este análisis debe incluir la evaluación de la respuesta del sistema ante diferentes entradas, como las señales escalón y rampa, y su comportamiento dinámico, que está influenciado por el tiempo de muestreo. Un aspecto crucial a tener en cuenta es que la precisión de un sistema digital

depende en gran medida del tiempo de muestreo utilizado. Un tiempo de muestreo demasiado grande puede provocar una pérdida significativa de información y reducir la precisión de la respuesta del sistema, lo que afecta su desempeño. Por el contrario, un tiempo de muestreo demasiado pequeño puede generar un mayor costo computacional sin aportar beneficios sustanciales en la precisión. Por tanto, encontrar un equilibrio adecuado en el tiempo de muestreo es fundamental para garantizar que los sistemas digitales ofrezcan un rendimiento óptimo. Este análisis es clave tanto para el diseño como para la implementación de sistemas de control y filtros digitales en la práctica actual.

REFERENCIAS BIBLIOGRÁFICAS

Ádám, T., Dadvandipour, S., & Futás, J. (2003). Influence of discretization method on the digital control system performance. *Acta Montanistica Slovaca Ročník*, *8*, 4.

Alexander Cortés Osorio, J., Baldomiro Cano Garzón, H., & Andrés Chaves Osorio, J. (n.d.). FUNDAMENTOS Y APLICACIÓN DEL MUESTREO EN SEÑALES UBICADAS EN LAS BANDAS ALTAS DEL ESPECTRO. *Scientia et Technica Año XIV*, *39*. Retrieved May 19, 2024, from http://prof.usb.ve/tperez/docencia/2422/contenido/muestreo/

Analysis of Transient Response of First & Second Order System using EXPEYES. (n.d.). Retrieved January 13, 2025, from https://www.researchgate.net/publication/353447390_Analysis_of_Transient_Response_of_First_Second_Order_System_using_EXPEYES

c2d. (n.d.). Retrieved January 13, 2025, from https://la.mathworks.com/help/control/ref/dynamicsystem.c2d.html

Chen, T., & Francis, B. A. (1995). Optimal Sampled-Data Control Systems. *Optimal Sampled-Data Control Systems*. https://doi.org/10.1007/978-1-4471-3037-6

control.matlab.c2d — Python Control Systems Library 0.10.1 documentation. (n.d.). Retrieved January 13, 2025, from https://pythoncontrol.readthedocs.io/en/0.10.1/generated/control.matlab.c2d.html#control.matlab.c2d

Diaz, A. E. (2002). Identificación con Modelos Discretos para Sistemas Lineales. Modelo Matemático y Aplicaciones. *Ingeniería*. https://www.academia.edu/50810606/Identificaci%C3%B3n_con_Modelos_Discretos_para_Sistemas_Lineales_Modelo_Matem%C3%A1tico_y_Aplicaciones

Esparza Arellano María Elena, & Avalos Briseño J. Benito. (2003). Reconocimiento de voz. *Conciencia Tecnológica*, (22). https://www.redalyc.org/articulo.oa?id=94402206

Franklin, G. F., Powell, J. D., & Workman, M. L. (1998). *Digital control of dynamic systems*. 742.

Hsu, H. P., Mata Hernández, G., & Alatorre Miguel, E. (2013). *Señales y sistemas*.

Nagahara, M., & Yamamoto, Y. (2013). Optimal discretization of analog filters via sampled-data H ∞ control theory. *Proceedings of the IEEE International Conference on Control Applications*, 527–532. https://doi.org/10.1109/CCA.2013.6662803

Pastor, L., Juan, S. F., Díaz De León, L., Cornelio, Y. M., & Charles, R. H. (2004). Cuantificación del Error en las Mediciones Debido a la Frecuencia de Muestreo The Error's Quantification in the Measurements Due to Sampling Frequency. *CIC-IPN*, *8*, 86–105.

Pérez, A. S. C., Rodríguez, G. Á., & Gómez, R. P. (2018). DISEÑO DE UN FILTRO DIGITAL PASA BAJAS DE PRIMER Y SEGUNDO ORDEN A PARTIR DE CIRCUITO RC. *Pistas Educativas*, *38*(120). https://pistaseducativas.celaya.tecnm.mx/index.php/pistas/article/view/602

Rabbath, C. A., & Léchevin, N. (2014). Control Systems. In *Discrete-Time Control System Design with Applications* (pp. 1–11). Springer New York. https://doi.org/10.1007/978-1-4614-9290-0_1

Vadhavkar, P. R., Watkins, J. M., & Paarmann, L. D. (2007). Mapping a controller from the s-domain to z-domain using Magnitude Invariance Method (MIM). *3rd Annual GRASP Symposium*, 199–200.