INFORME - TALLER II
ANALISIS DE SISTEMAS DINAMICOS

Bustos Mendez Jorge Fernando
Narvaez Arturo Nicolas
Universidad de Nariño

2023

Resumen

En el presente informe busca abordar se plantean múltiples problemas que involucran ecuaciones en diferencia con el objetivo de abordar su resolución a través del método analítico de la Transformada Z. En segundo lugar, se procede a resolver estos problemas empleando métodos computacionales, permitiendo así una validación exhaustiva y la posterior comparación de los resultados obtenidos. Este enfoque integral busca proporcionar una perspectiva completa y precisa de las soluciones, incorporando tanto el análisis analítico como el computacional para garantizar una evaluación rigurosa y completa de los problemas planteados.
Palabras Clave: Ecuacion en Diferencia, Transformada Z, Sistemas Dinámicos, Analitica, Matlab, Simulink.

1. Introducción

En el contexto de este informe dedicado al análisis dinámico, nos sumergimos en la resolución de problemas mediante Ecuaciones en Diferencia y la aplicación clave de la Transformada Z.
En la primera fase, nos centramos en la resolución analítica de ecuaciones, utilizando la Transformada Z para desentrañar las complejidades de sistemas dinámicos. La segunda etapa abraza un enfoque computacional, ampliando nuestra validación y permitiendo una comparación exhaustiva de resultados.
La conjunción de métodos analíticos y computacionales enriquece la comprensión y confiabilidad de las soluciones, ofreciendo una contribución integral al análisis dinámico y validando las soluciones de ecuaciones en diferencia mediante la Transformada Z.

2. Transformada Z - Solución de Ecuaciones en Diferencia

Para este problema se determinará mediante el uso de la Transformada Z la solución analítica de las siguientes ecuaciones en diferencias:

$$y[k+2] + 2y[k+1] + y[k] = 3^k \tag{1}$$

$$y[k+2] - y[k+1] + y[k] = (\frac{b}{0{,}2})^k \tag{2}$$

con condiciones iniciales iguales a 0.

2.1. Primera Ecuación en Diferencia
2.1.1. Desarrollo Analítico

Para el desarrollo analítico haremos el uso de la Transformada Z, en ese caso para la ecuación (1) tendriamos lo siguiente:

$$y[k+2] + 2y[k+1] + y[k] = 3^k$$

$$[z^2Y(z) - z^2y[0] - zy[1]] + 2[zY(z) - zy[0]] + Y(z) = 3^k$$

como las condiciones iniciales son 0, cancelamos algunos términos

$$z^2Y(z) + 2zY(z) + Y(z) = \frac{z}{(z-3)}$$

$$Y(z)[z^2 + 2z + 1] = \frac{z}{(z-3)}$$

$$Y(z) = \frac{z}{(z^2 + 2z + 1)(z-3)}$$

$$Y(z) = z\left(\frac{1}{(z^2 + 2z + 1)(z-3)}\right)$$

aplicaremos el método de fracciones parciales para hallar las constantes:

$$Y(z) = z\left(\frac{A}{(z-3)} + \frac{B}{(z+1)} + \frac{C}{(z+1)^2}\right) \quad (3)$$

Para el caso de la constante A tenemos

$$A = \frac{1}{(z+1)^2} = \frac{1}{(3+1)^2} = \frac{1}{4^2} = \frac{1}{16}$$

Para el caso de la constante C tenemos

$$C = \frac{1}{(z-3)} = \frac{1}{(-1-3)} = \frac{1}{-4} = -\frac{1}{4}$$

Cómo ya tenemos los valores de A y C, para encontrar el valor de la constante B igualamos z a 0 en

$$\frac{1}{(z-3)(z+1)^2} = \frac{A}{(z-3)} + \frac{B}{(z+1)} + \frac{C}{(z+1)^2}$$

obteniendo asi

$$\frac{1}{(-3)(1)^2} = \frac{A}{(-3)} + \frac{B}{(1)} + \frac{C}{(1)^2}$$

$$\frac{1}{-3} = \frac{A}{-3} + B + C$$

despejamos B de la ecuación,

$$B = \frac{1 - A + 3C}{-3}$$

reemplazamos los valores de A y C obtenidos anteriormente,

$$B = \frac{1 - \frac{1}{16} + 3\left(\frac{1}{-4}\right)}{-3} = -\frac{1}{16}$$

ahora que tenmos todas las constantes, sustituimos sus valores en la ecuación (3)

$$Y(z) = z\left(\frac{\frac{1}{16}}{(z-3)} + \frac{-\frac{1}{16}}{(z+1)} + \frac{-\frac{1}{4}}{(z+1)^2}\right)$$

$$Y(z) = z\left(\frac{1}{16}\frac{1}{(z-3)} - \frac{1}{16}\frac{1}{(z+1)} - \frac{1}{4}\frac{1}{(z+1)^2}\right)$$

$$Y(z) = \frac{1}{16}\frac{z}{(z-3)} - \frac{1}{16}\frac{z}{(z+1)} + \frac{1}{4}\frac{-z}{(z+1)^2}$$

aplicamos la inversa para hallar la solución final

$$Y(z) = \frac{1}{16}(3)^k - \frac{1}{16}(-1)^k + \frac{1}{4}k(-1)^k \quad (4)$$

Para combrobar el resultado evaluamos k en 0, por lo tanto, nos deberia dar 0

$$Y(z) = \frac{1}{16}(3)^0 - \frac{1}{16}(-1)^0 + \frac{1}{4}0(-1)^0$$

$$Y(z) = \frac{1}{16} - \frac{1}{16} + 0$$

$$Y(z) = 0$$

2.1.2. Solución Graficada en Matlab

Para realizar la gráfica de la solución haremos uso de la herramienta computacional Matlab, para esto emplearemos los siguientes códigos:

1. Solución con el comando plot

En éste código empleamos la función **iztrans** la cuál nos arroja la solución de la función inversa que obtuvimos en la ecuación (4) para asi, con la función **plot** poder graficar la respuesta de la ecuación en diferencia de la ecuación (1).

```
syms z;
k = 0 : 0.01 : 60 ;
fz = z*(((1/16)/(z-3))+((-1/16)/(z+1))+((-1/4)/(z+1)^2))
fk = iztrans (fz,k);
y = fk;
plot(k,y, 'g')
title("Solucion analitica primera ecuacion en diferencia Matlab")
xlabel("Tiempo (s)")
ylabel("Amplitud")
xlim([0 60.5])
ylim([-0.1 270530687572180524835065O368])
```

Figura 1: Código empleado en Matlab para graficar la solución de la primera ecuación en diferencia.

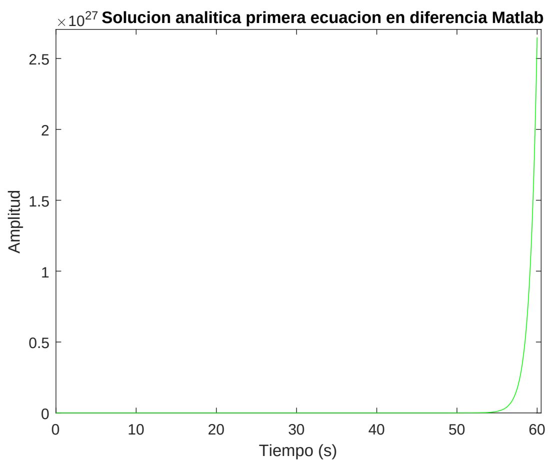

Figura 2: Gráfica primera ecuación en diferencia en Matlab usando la función plot.

2. Solución con el comando step

En éste código empleamos la función **tf** la cuál nos arroja la solución de la función inversa que obtuvimos en la ecuación (4) para asi, con la función **step** poder graficar la respuesta de la ecuación en diferencia de la ecuación (1).

```
n = [1 0];
d = [1 -1 -5 -3];
g = tf(n,d,1)
step(g, 'g')
title("Solucion analitica primera ecuacion en diferencia Matlab")
xlabel("Tiempo")
ylabel("Amplitud")
```

Figura 3: Código empleado en Matlab para graficar la solución de la primera ecuación en diferencia.

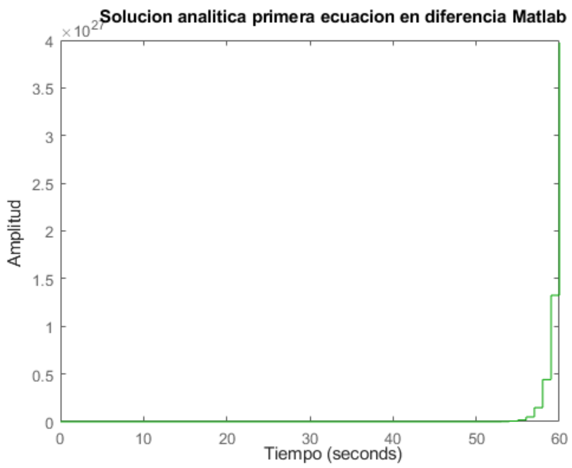

Figura 4: Gráfica primera ecuación en diferencia en Matlab usando la función plot.

2.1.3. *Solución Graficada en Simulink*

Para esta solución hicimos un diagrama de bloques en la herramienta computacional Simulink usando los siguientes elementos:

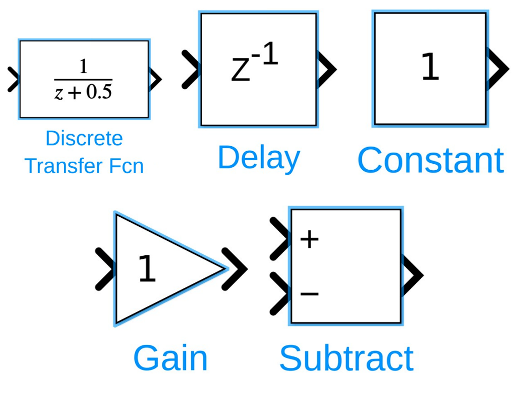

Figura 5: Elementos utilizados para el diagrama de bloques.

Haciendo las respectivas conexiones de los elementos y adecuando los valores según lo dice la ecuación en diferencia, tenemos el siguiente diagrama de bloques hecho en Simulink

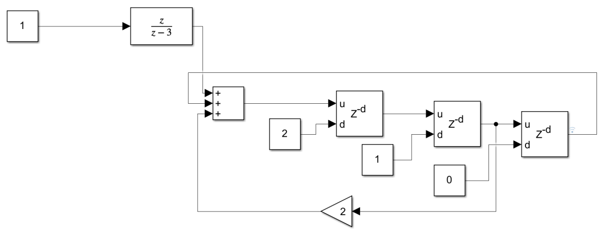

Figura 6: Diagrama de bloques Simulink primera ecuación en diferencia.

el cual al momento de correrlo en el programa nos da la siguiente gráfica:

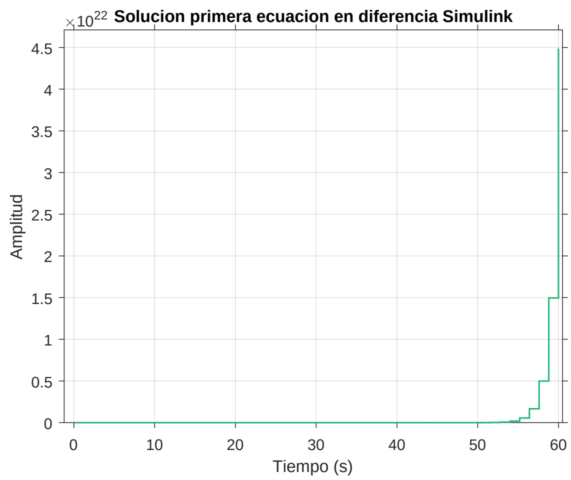

Figura 7: Gráfica primera ecuación en diferencia Simulink.

2.2. *Segunda Ecuación en Diferencia*

2.2.1. *Desarrollo Analítico*

Para el desarrollo analítico haremos el uso de la Transformada Z, en este caso para la ecuación (2) tenemos:

$$y[k+2] - y[k+1] + y[k] = (\frac{b}{0{,}2})^k$$

siendo $\mathbf{b = 6}$, ya que el penúltimo digito del código **221160068** es **6**, por tanto se tiene

$$(\frac{b}{0{,}2})^k = (\frac{6}{0{,}2})^k = 30^k$$

entonces la nueva ecuación a resolver es

$$y[k+2] - y[k+1] + y[k] = 30^k \tag{5}$$

$$[z^2Y(z) - z^2y[0] - zy[1]] - [zY(z) - zy[0]] + Y(z) = 30^k$$

ahora, como las condiciones iniciales son 0 como en la primera ecuación en diferencia, cancelamos algunos términos

$$z^2Y(z) - zY(z) + Y(z) = \frac{z}{(z-30)}$$

$$Y(z)[z^2 - z + 1] = \frac{z}{(z-30)}$$

$$Y(z) = \frac{z}{(z^2 - z + 1)(z-30)}$$

$$Y(z) = z\left(\frac{1}{(z^2 - z + 1)(z-30)}\right)$$

aplicaremos el método de fracciones parciales para hallar las constantes:

$$\frac{1}{(z-30)(z^2 - z + 1)} = \frac{Az + B}{(z^2 - z + 1)} + \frac{C}{(z-30)}$$

Dividimos los terminos

$$\frac{Az + B}{(z^2 - z + 1)} + \frac{C}{(z-30)}$$

entre

$$(z-30)(z^2 - z + 1)$$

obteniendo asi

$$1 = (Az + B)(z - 30) + C(z^2 - z + 1)$$
$$1 = Az^2 - 30Az + Bz - 30B + Cz^2 - Cz + C$$

Igualamos todos los términos que tengan el mismo grado, obteniendo asi el siguiente sistema de ecuaciones

$$0 = Az^2 + Cz^2 \tag{6}$$

$$0 = -30Az + Bz - Cz \tag{7}$$

$$1 = -30B + C \tag{8}$$

de la ecuación (6) despejamos A

$$A = -C \tag{9}$$

de la ecuación (8) despejamos B

$$B = \frac{C-1}{30} \tag{10}$$

reemplazamos (9) y (10) en la ecuación (7)

$$-30(-C) + \left(\frac{C-1}{30}\right) - C = 0$$

$$30C + \frac{C-1}{30} - C = 0$$

$$30C + \frac{C}{30} - \frac{1}{30} - C = 0$$

$$900C + C - 1 - 30C = 0$$

$$871C = 1$$

$$C = \frac{1}{871}$$

ya que tenemos el valor de C, lo reemplazamos en (9)

$$A = -C$$

$$A = -\left(\frac{1}{871}\right)$$

$$A = -\frac{1}{871}$$

por último, para hallar el valor de B, sustituimos los valores de A y C en (10)

$$B = \frac{C-10}{30}$$

$$B = \frac{\frac{1}{871} - 1}{30}$$

$$B = -\frac{29}{871}$$

ahora que tenmos todas las constantes, sustituimos sus valores en la siguiente ecuación

$$Y(z) = z\left(\frac{Az+B}{(z^2-z+1)} + \frac{C}{(z-30)}\right)$$

$$Y(z) = z\left(\frac{-\frac{1}{871}z - \frac{29}{871}}{(z^2-z+1)} + \frac{\frac{1}{871}}{(z-30)}\right)$$

en este problema no aplicamos la inversa para hallar la solución final ya que se nos dificultó en cierta medida, por tanto, hicimos el uso de Matlab para encontrar la respuesta final que se ve a continuación:

$$\frac{30^k}{871} - \frac{60\,(-1)^k \cos\left(\frac{2\,\pi\,k}{3}\right)}{871} - \frac{59\,(-1)^k\,\sqrt{3}\,\left(-\frac{1}{2}-\frac{\sqrt{3}\,\mathrm{i}}{2}\right)^{k-1}\mathrm{i}}{2613} + \frac{59\,(-1)^k\,\sqrt{3}\,\left(-\frac{1}{2}+\frac{\sqrt{3}\,\mathrm{i}}{2}\right)^{k-1}\mathrm{i}}{2613}$$

Figura 8: Solución inversa en Matlab

2.2.2. *Solución Graficada en Matlab*

Para realizar la gráfica de la solución haremos uso de la herramienta computacional Matlab, para esto emplearemos los siguientes códigos:

1. Solución con el comando plot

En éste código empleamos la función **iztrans** la cuál nos arroja la solución de la función inversa que obtuvimos en la **Figura 8** para asi, con la función **plot** poder graficar la respuesta de la ecuación en diferencia de la ecuación (5).

```
syms z;
k = 0 : 0.01 : 90 ;
fz = z*(((((-1/871)*z)+(-29/871))/(z^2-z+1))+((1/871/(z-30))))
fk = iztrans (fz,k);
y = fk;
plot(k,y, 'm')
title("Solucion analitica segunda ecuacion en diferencia Matlab")
xlabel("Tiempo (s)")
ylabel("Amplitud")
xlim([0 90.5])
ylim([0 120262577456273463384730461589612545695945493259979929592534
```

Figura 9: Código empleado en Matlab para graficar la solución de la primera ecuación en diferencia.

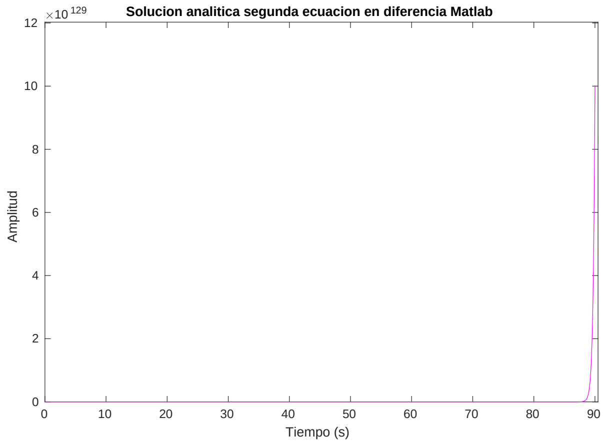

Figura 10: Gráfica segunda ecuación en diferencia en Matlab usando la función plot.

2. Solución con el comando step

En éste código empleamos la función **tf** la cuál nos arroja la solución de la función inversa que obtuvimos en la **Figura 8** para asi, con la función **step** poder graficar la respuesta de la ecuación en diferencia de la ecuación (5).

```
n = [1 0];
d = [1 -31 31 -30];
g = tf(n,d,1)
step(g, 'm')
title("Solucion analitica segunda ecuacion en diferencia Matlab")
xlabel("Tiempo")
ylabel("Amplitud")
```

Figura 11: Código empleado en Matlab para graficar la solución de la segunda ecuación en diferencia.

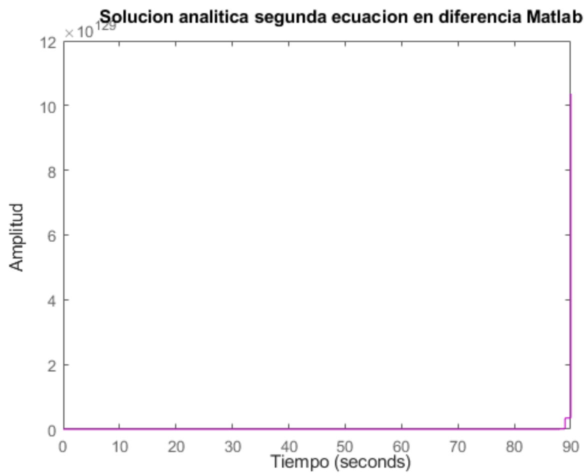

Figura 12: Gráfica primera ecuación en diferencia en Matlab usando la función plot.

2.2.3. *Solución Graficada en Simulink*

En esta solución hicimos un diagrama de bloques en la herramienta computacional Simulink usando los mismos elementos de la primera ecuación en diferencia:

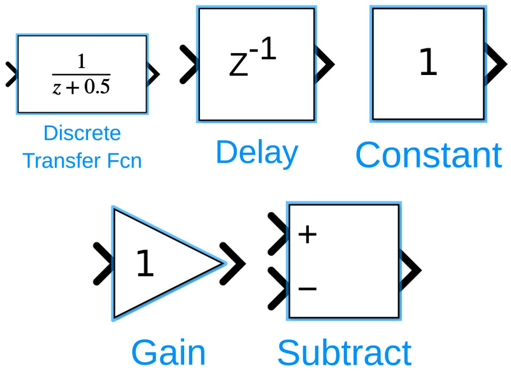

Figura 13: Elementos utilizados para el diagrama de bloques.

Haciendo las respectivas conexiones de los elementos y adecuando los valores según lo dice la ecuación en diferencia, tenemos el siguiente diagrama de bloques hecho en Simulink

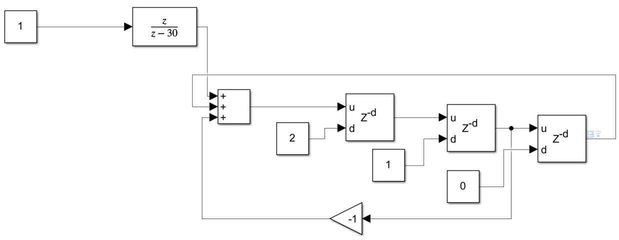

Figura 14: Diagrama de bloques Simulink segunda ecuación en diferencia.

el cual al momento de correrlo en el programa nos da la siguiente gráfica:

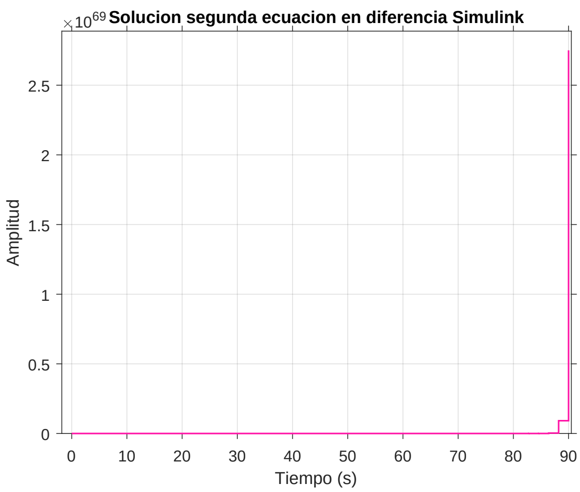

Figura 15: Gráfica segunda ecuación en diferencia Simulink.

2.3. Desarrollo mediante algoritmo en un script de Matlab

Para este caso se hizo un código en Matlab que permitió resolver las ecuaciones en diferencia propuestas e indicar sus respuestas gráficamente, para ello se usó un ciclo for como se muestra a continuación:

```
clear all; close all;
y(1)=0;
y(2)=0;
y(3)=0;
for k= 1:60
    y(k+2)=(3^k)-(2*y(k+1))-y(k);
end
bar(y)
grid on
```

Figura 16: Código para la primera ecuación en diferencia desarrollado en Matlab.

```
clear all; close all;
y(1)=0;
y(2)=0;
y(3)=0;
for k= 1:90
    y(k+2)=(30^k)+y(k+1)-y(k);
end
bar(y)
grid on
```

Figura 17: Código para la segunda ecuación en diferencia desarrollado en Matlab.

obteniendo asi, las siguientes graficas:

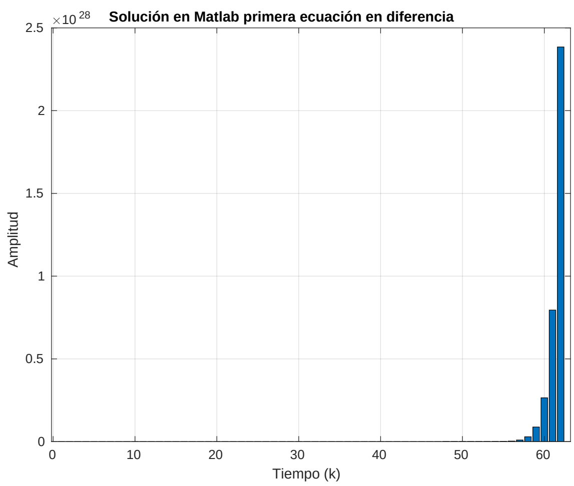

Figura 18: Gráfica primera ecuación en diferencia desarrollada en Matlab.

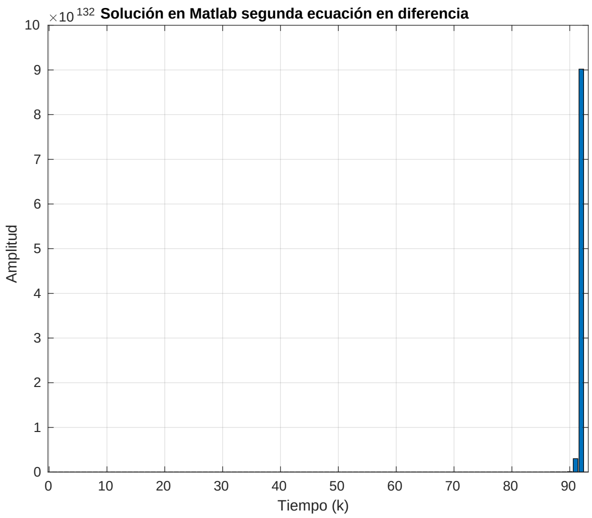

Figura 19: Gráfica segunda ecuación en diferencia desarrolado en Matlab.

2.4. Análisis y Conclusiones

En este laboratorio, nos enbarcamos en la resolución analítica de ecuaciones en diferencia mediante la transformada Z, apoyándonos en MATLAB para enfrentar las partes más desafiantes del proceso. Este enfoque se revela como esencial para ingenieros electrónicos, ya que las habilidades analíticas son fundamentales en el diseño y análisis de sistemas de control, que serán de suma importancia en futuras aplicaciones.
Durante la resolución de la segunda ecuación en diferencia, MATLAB desempeñó un papel crucial al abordar números complejos en la respuesta final. La capacidad de MATLAB para trabajar con componentes imaginarios permitió una interpretación clara y precisa de la solución. Esta experiencia subraya la importancia de las herramientas computacionales en situaciones donde las soluciones involucran números complejos, destacando la versatilidad y utilidad de MATLAB en el contexto de la ingeniería electrónica.
La validación de los resultados mediante tres métodos diferentes - el análitico, simulación en Simulink y un script de MATLAB - es destacable. Logramos comprobar exitosamente que los tres métodos producían respuestas coherentes, aunque con niveles de complejidad y susceptibilidad a errores diferentes. La eficiencia del método analítico se destacó como el más sencillo, seguido por la solución simulada en Simulink y, por último, el script de MATLAB. Esta comparación entre métodos resalta la importancia de elegir la herramienta más adecuada para abordar problemas específicos, considerando la complejidad del sistema y minimizando posibles errores en el proceso de resolución. El trabajo colaborativo y la combinación de habilidades analíticas y computacionales en nuestro grupo fueron fundamentales para el éxito de este laboratorio.

3. Convolución Discreta y Filtro Digital

Vamos a usar el algoritmo de la convolución discreta o una secuencia de ponderación para abordar el proceso de filtración de una señal con ruido. En este ejercicio, estableceremos una frecuencia de muestreo de 1000 Hz o un intervalo de muestreo de 0.001 segundos. Utilizaremos una señal generada en Matlab de la siguiente manera:

```
Ts=0.001; Fs=1/Ts;
for k = 1:100
x(k) = 10*sin(2*pi*60*Ts*k);
r(k) = 2.5*sin(2*pi*400*Ts*k);
t(k) = Ts*k;
xr(k) = x(k)+r(k);
end
plot(t,xr,'b')
```

Figura 20: Código para generar una señal en Matlab.

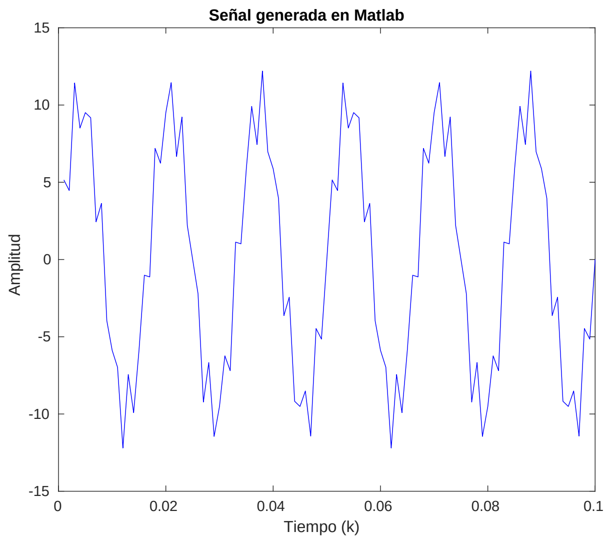

Figura 21: Gráfica señal generada con Matlab.

3.1. Diseño de un filtro

Para crear un filtro que permita el paso de frecuencias hasta 100 Hz, con una frecuencia de corte en 150 Hz y diseñado para tener 16 coeficientes, empleamos la función *designfilt* de la siguiente manera:

```
filtro = designfilt('lowpassfir' ...
    ,'FilterOrder',15, ...
    'PassbandFrequency',100, ...
    'StopbandFrequency',150, ...
    'DesignMethod','ls', ...
    'SampleRate',1000);
hold on;
```

Figura 22: Función designfilt para diseñar un filtro.

Los 16 coeficientes de este filtro FIR con un orden de 15($N = 15$) se pueden extraer de *filtro.Coefficients* en Matlab. Es importante destacar que la respuesta de un filtro FIR se calcula utilizando la entrada actual $x(k)$ y las entradas previas $x(k-1), x(k-2), ..., x(k-15)$, cada una multiplicada por los coeficientes del filtro bn. Matemáticamente, esta respuesta se expresa de la siguiente manera:

$$y(k) = b_0 x(k) + b_1 x(k-1) + ... + b_{N-1} x(k-N+1) \tag{11}$$

$$y(k) = \sum_{n=0}^{N-1} b_n x(k-n) \tag{12}$$

Los resultados descritos en las ecuaciones (11) y (12) son fácilmente obtenibles en Matlab utilizando la función *filter*, y su representación espectral puede ser adquirida mediante el uso de *freqz*.

```
y = filter(filtro,xr);
hold on;
plot(t,y,'r')
figure(2);
freqz(filtro.Coefficients,1,[],Fs);
```

Figura 23: Función filter y freqz para obtener los resultados del filtro digital.

En el gráfico, notaremos una señal filtrada que podría estar retrasada en unas 16 muestras en comparación con la señal de entrada ruidosa $xr(k)$. Este retraso se debe a la fase que introduce un filtro de esta magnitud

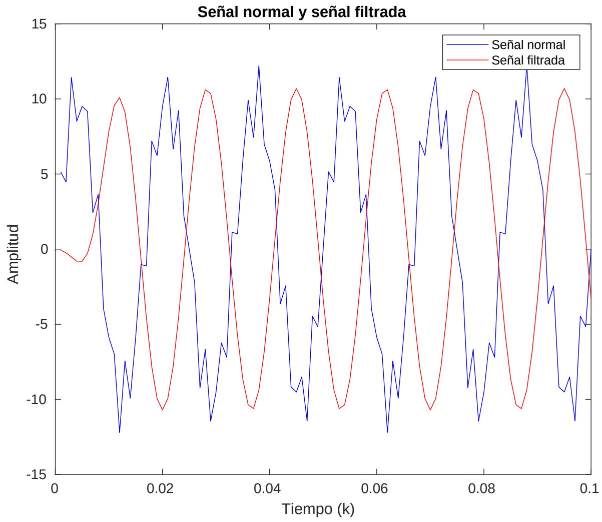

Figura 24: Gráfica señal con ruido y señal filtrada.

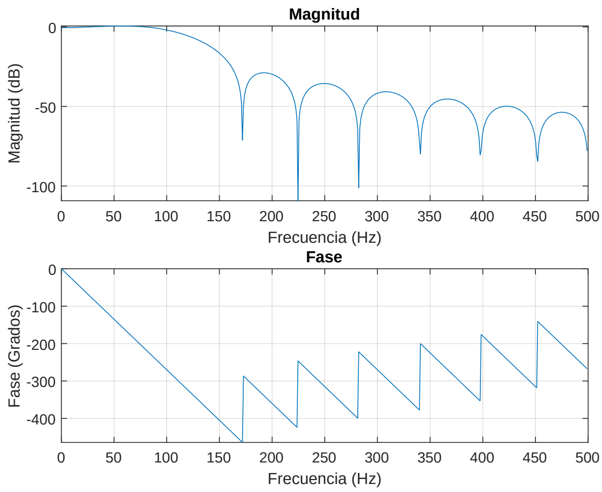

Figura 25: Gráficas de Magnitud y Fase .

3.2. Convolución Discreta

Ahora, exploraremos la respuesta obtenida en el problema anterior a través de la función *filter* en Matlab mediante un algoritmo propio de convolución discreta, para esto, resolveremos con ayuda del modelo de lenguaje GPT-3.5, proporcionado por OpenAI [4] la suma de convoluciones para comprender mejor el proceso

$$y(k) = \sum_{n=0}^{N-1} h(k) \cdot x(k-n) \tag{13}$$

Con la siguiente función pudimos graficar la convolución discreta de la sumatoria anterior

```
y_convolucion = zeros(size(xr));

for n = 16:length(xr)
    y_convolucion(n) = sum(xr(n:-1:n-15) .* coeficientes_filtro);
end
```

Figura 26: Convolución discreta Matlab.

dando como solución la grafica que se muestra a continuación:

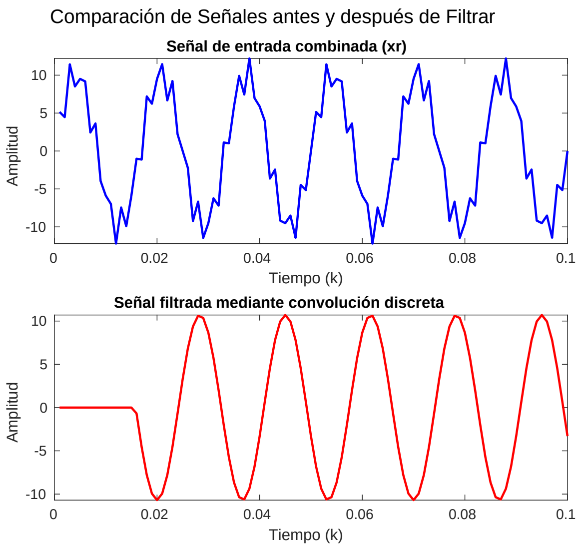

Figura 27: Convolución discreta Matlab.

Tambien se empleó una función para obtener la respuesta en el impulso de cada señal

```
respuesta_impulso = impz(filtro);
y_convolucion = conv(xr, coeficientes_filtro, 'full');
```

Figura 28: Código para generar la respuesta al impulso del filtro.

obteniendo asi la siguiente figura

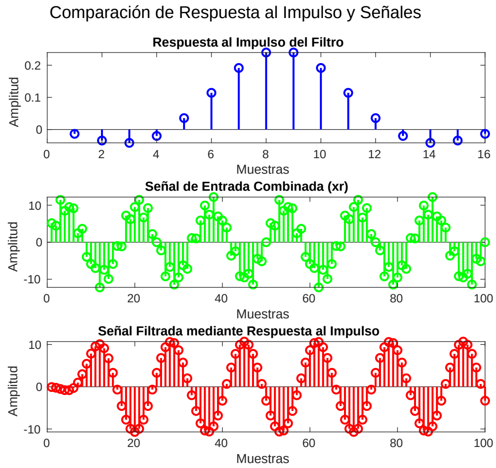

Figura 29: Código para generar la respuesta al impulso del filtro.

La secuencia $h(k)$ es identificada como la respuesta al impulso del filtro y se encuentra restringida al conjunto finito de los 16 coeficientes obtenidos durante el diseño.

```
figure(3);
stem(filtro.Coefficients)
```

Figura 30: Código para generar la respuesta al impulso del filtro.

La secuencia $x(k)$ comprende 100 muestras obtenidas al principio de este ejercicio. El objetivo es resolver la convolución y comparar su rendimiento con la salida proporcionada por Matlab utilizando la función *filter*. Se espera obtener una señal filtrada de 100 valores con un retraso uniforme de 16 muestras con respecto a la señal original con ruido.

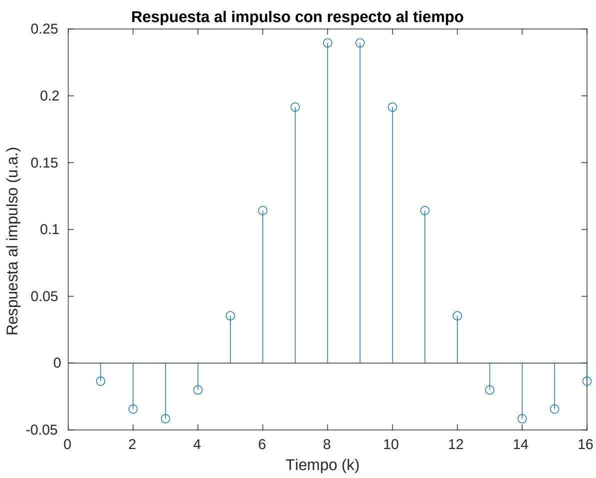

Figura 31: Gráfica impulso del filtro con respecto al tiempo k.

3.3. *Solución mediante diagrama de bloques en Simulink*

Una manera diferente de llevar a cabo el procedimiento de filtración es mediante la implementación de un diagrama de bloques que emplea la transformada Z. Para lograrlo, es necesario representar el filtro FIR en términos de la variable Z utilizando la ecuación (11), es decir

$$y(k) = b_0 X(z) + b_1 X(z) Z^{-1} + ... + b_{N-1} X(z) Z^{-N+1} \tag{14}$$

$$y(k) = \sum_{n=0}^{N-1} b_n Z^n X(z) \tag{15}$$

Esta fórmula puede ser modelada en Simulink utilizando un sumador con 16 entradas asociadas a $x(k); x(k-1); x(k-2), ...x(k-15)$ y 15 bloques de retraso. Para llevar a cabo la simulación, se ajustará el periodo de muestreo constante a $0{,}001s$ *Fixed-step* en los parámetros de configuración del *Solver* en Simulink.

Para el diseño del diagrama usamos los siguientes elementos:

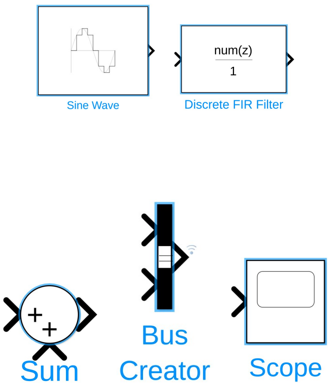

Figura 32: Elementos utilizados para el diagrama de bloques.

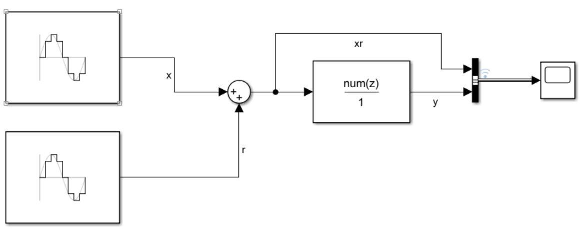

Figura 33: Diagrama de bloques de la ecuación modelada.

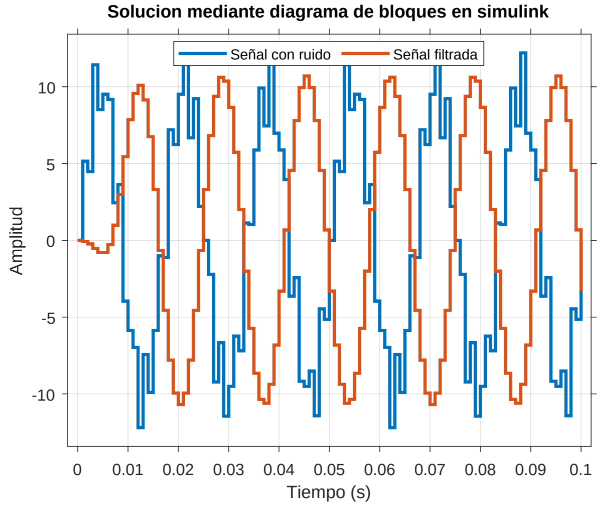

Figura 34: Gráfica obtenida mediante el diagrama de bloques en Simulink.

3.4. *Análisis y Conclusiones*

A lo largo de este laboratorio, exploramos minuciosamente la aplicación de la transformada Z en la convolución discreta y el diseño de filtros digitales, destacando la trascendental importancia de la discretización de señales en el ámbito de la ingeniería electrónica. Al incorporar la transformada Z en este contexto, no solo logramos entender y analizar eficazmente señales discretas, una habilidad fundamental en la ingeniería moderna, sino que también reconocimos cómo la discretización simplifica no solo el análisis y diseño de sistemas, sino también su implementación práctica en entornos digitales.
La utilización de filtros digitales, especialmente los FIR (Finite Impulse Response), dentro del entorno de MATLAB, emergió como un componente esencial de nuestro aprendizaje. No solo nos capacitamos para utilizar y simular estos filtros, sino que también profundizamos en el diseño de los mismos, asegurando que cumplieran con requisitos específicos de respuesta en frecuencia y comportamiento. Esta experiencia práctica no solo refuerza nuestra comprensión teórica, sino que también demuestra la aplicabilidad directa de estos conceptos en situaciones del mundo real, resaltando su relevancia en la implementación efectiva de sistemas electrónicos modernos.
Al expandir nuestras conclusiones sobre la transformada Z y los tiempos discretos, reconocemos que estas herramientas no solo son esenciales para la resolución analítica de ecuaciones en diferencia, como abordamos en la primera parte del laboratorio, sino que también son fundamentales para el diseño y análisis de sistemas en tiempo discreto. La transformada Z, al permitirnos llevar a cabo análisis en el dominio Z, nos ofrece una ventana única

para comprender y manipular señales digitales. En este contexto, los tiempos discretos no solo simplifican la representación y manipulación de señales, sino que también son cruciales para el diseño eficaz de sistemas discretos y la implementación de algoritmos digitales.
Este laboratorio no solo nos proporcionó conocimientos detallados sobre la aplicación de la transformada Z y la convolución discreta, sino que también resaltó la trascendental importancia de la discretización de señales y la aplicación de filtros digitales en el ámbito de la ingeniería electrónica. Estas herramientas teóricas no solo son fundamentales, sino que también son altamente aplicables en la implementación exitosa de sistemas electrónicos modernos, consolidando su papel esencial en la ingeniería electrónica actual.
En conclusión, este laboratorio no solo consolidó nuestra comprensión de la transformada Z, la convolución discreta y los filtros digitales, sino que también resaltó la trascendental importancia de la discretización de señales y la aplicación de herramientas computacionales en la ingeniería electrónica. Estas habilidades no solo son esenciales en el análisis teórico, sino también altamente aplicables en la implementación exitosa de sistemas electrónicos modernos, destacando su papel esencial en el campo de la ingeniería electrónica.

Referencias

[1] C. T. Chen .ᴬnalog and Digital Control System Design. ",2da Ed. Saunders College Publishing, 2002.

[2] K. Ogata. Ïngeniería de Control Moderna. ",4a Ed. México: Prentice Hall Panamericana, 2003.

[3] A. Poularikas. "The Transforms and Applications Handbook. ",2a Ed. Alabama: CRC Press, 2000.

[4] OpenAI. GPT-3.5 (Generative Pretrained Transformer 3.5),www.openai.com.

[5] The MathWorks, Çontrol System Toolbox for use with Matlab",www.matworws.com.

[6] Getting Started with Simulink,,www.matworws.com.