

**DISEÑO DE UNA INTERFAZ EN EL GUI DE MATLAB PARA EL
DESARROLLO DE LAS TRANSFORMACIONES A DOMINIO S Y
DOMINIO Z**

LUIS GABRIEL ARÉVALO HERRERA

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERIA
INGENIERIA ELECTRÓNICA
NEIVA
2007**

**DISEÑO DE UNA INTERFAZ EN EL GUI DE MATLAB PARA EL
DESARROLLO DE LAS TRANSFORMACIONES A DOMINIO S Y
DOMINIO Z**

LUIS GABRIEL ARÉVALO HERRERA

**Director
AGUSTIN SOTO OTALORA
Ingeniero Electrónico
Especialista en Automatización Industrial**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERIA
INGENIERIA ELECTRÓNICA
NEIVA
2007**

Nota de Aceptación

Director

Firma del Jurado

Firma del Jurado

Neiva, agosto 17 de 2007

A Dios y a mi familia, en especial a mi abuelita Bertha, mi tío Gilberto y Samuel mi padre. Gracias por estar conmigo en los buenos y malos momentos

Luis Gabriel

RESUMEN

El objetivo principal de este trabajo es desarrollar una herramienta en el GUI de Matlab, con el fin de facilitar la comprensión de la respuesta dada por Matlab ante un problema determinado en el tema de las transformaciones de dominio matemático, como lo son la transformada de Laplace, la transformada Z y sus respectivas operaciones inversas.

Se pretende en cierta medida dar conocimiento al usuario del procedimiento empleado para llegar a la solución de estos problemas matemáticos, no solo mediante una elaborada respuesta sino también mediante el uso del menú de ventanas emergentes incluido en el diseño, a través del cual se puede conocer que teoremas son aplicados en los distintos casos. Además, mediante el uso de casillas de edición de texto se crea un ambiente mas agradable a la hora de desarrollar tareas de transformación de dominio matemático.

En la elaboración del software se empleó la herramienta de edición GUIDE del Matlab, ya que constituye una forma sencilla de realizar tareas de programación de un ambiente gráfico. El código base del trabajo se torna un poco extenso debido a que cada módulo diseñado emplea procesos iterativos, los cuales se vuelven recurrentes con el fin de llegar a un calculo determinado

ABSTRACT

The main objective of this work is to develop a tool in the Matlab GUI in order to make easy the understanding of the answer in Matlab for a given problem in the subject of mathematical dominium transformations, like Laplace transformation, Z transformation and their respective inverse operations.

This work pretends to give to the user a knowledge about the proceeding used to find a solution for this kind of mathematical problems, not only by the answer but also by the use of the menu of emergent windows included in the design where you can know about the aplicated theorems in the different cases. Furthermore, by the use of edition boxes it creates a nicer environment to develop works of mathematical dominium transformation.

In the software elaboration it used the GUIDE edition tool of Matlab which constitutes a simple way of making works of programation of a grafic environment. The base code it is a little extensive due to each designed module use iterative processes.

TABLA DE CONTENIDO

	pag
INTRODUCCIÓN	13
1. MARCO TEORICO	14
1.1 TRANSFORMADA DE LAPLACE	14
1.1.1 Continuidad seccional a trazos	14
1.1.2 Funciones de orden exponencial	15
1.1.3 Condiciones para la existencia de la transformada S	15
1.1.4 Transformada de Laplace de funciones elementales	15
1.1.4.1 Función exponencial	15
1.1.4.2 Función escalón	16
1.1.4.3 Función rampa	16
1.1.4.4 Funciones seno y coseno	18
1.1.4.5 Función impulso	18
1.1.5 Propiedades y teoremas	18
1.1.5.1 Propiedad de linealidad	18
1.1.5.2 Teoremas de traslación	18
1.1.5.3 Cambio de escala	18
1.1.5.4 Teorema del valor inicial	18
1.1.5.5 Teorema del valor final	18
1.2 TRANSFORMADA INVERSA DE LAPLACE	19

	pag
1.2.1 Métodos para hallar la transformada inversa S	19
1.2.1.1 Método de expansión en fracciones parciales	19
1.3 TRANSFORMADA Z	22
1.3.1 Transformada z de funciones elementales	23
1.3.1.1 Función escalón unitario	23
1.3.1.2 Función rampa unitaria	23
1.3.1.3 Función polinomial	23
1.3.1.4 Función exponencial	23
1.3.1.5 Función senoidal	23
1.3.2 Propiedades y teoremas importantes	24
1.3.2.1 Multiplicación por a^k	24
1.3.2.2 Teorema de traslación compleja	24
1.3.2.3 Teorema del valor inicial	24
1.3.2.4 Teorema del valor final	24
1.4 TRANSFORMADA Z INVERSA	24
1.4.1 Método de expansión en fracciones parciales	25
1.4.2 Método computacional	26
1.5 PROGRAMACION VISUAL CON MATLAB	27
1.5.1 Breve descripción de ambiente de desarrollo de interfaz de Usuario	27
1.5.2 Cuadro de herramientas	28
1.5.2.1 Push Buttons	28

	pag
1.5.2.2 Radio Buttons	29
1.5.2.3 Check Boxes	29
1.5.2.4 Edit text	29
1.5.2.5 Pop Up menús	29
1.5.2.6 Menús	29
2. PROCEDIMIENTO PARA EL DESARROLLO DEL SOFTWARE	30
2.1 CREACION DE AMBIENTE	30
2.1.1 Modulo de transformada inversa de Laplace	30
2.1.2 Modulo de transformada Z inversa	31
2.1.3 Modulo de transformada de Laplace	32
2.1.4 Modulo de transformada Z	33
2.1.5 Modulo de teoremas	35
2.1.5.1 Transformada Z	35
2.1.5.2 Transformada de Laplace	35
2.2 PROGRAMACION GENERAL	35
2.2.1 Transformada inversa de Laplace	35
2.2.1.1 Raíces reales	35
2.2.1.2 Raíces complejas simples	36
2.2.2 Transformada Z inversa	37
2.2.2.1 Expansión en fracciones parciales	37
2.2.2.2 Método computacional	38

	pag
2.2.3 Transformada de Laplace	38
2.2.4 Transformada Z	39
2.3 VISUALIZACION DE DATOS	39
3. EJEMPLOS DE APLICACIÓN	42
4. CONCLUSIONES	49
BIBLIOGRAFIA	50
ANEXO. Código de programación	51

LISTA DE FIGURAS

	pag
Figura 1. Solución al problema	13
Figura 2. Continuidad seccional a trazos	14
Figura 3. Editor de diseño	27
Figura 4. Archivo .m de un formulario vacío	28
Figura 5. Editor de propiedades	28
Figura 6. Vista general modulo Transformada Inversa de Laplace	30
Figura 7. Vista general modulo Transformada Z Inversa	32
Figura 8. Vista general modulo Transformada de Laplace	33
Figura 9. Vista general modulo Transforma Z	34
Figura 10. Casillas de texto auxiliares	40
Figura 11. Ingreso al sistema, Ejercicio 1	43
Figura 12. Ingreso al sistema Ejercicio 2	44
Figura 13. Ingreso al sistema Ejercicio 3	46
Figura 14. Ingreso al sistema Ejercicio 4	47

LISTA DE TABLAS

	pag
Tabla 1. Descripción de archivos Transformada inversa de Laplace	36
Tabla 2. Descripción de archivos Transformada Z Inversa (Fracciones parciales)	37
Tabla 3. Descripción de archivos Transformada Z inversa (Método computacional)	38
Tabla 4. Descripción de archivos transformada de Laplace	38
Tabla 5. Descripción de archivos transformada Z	39

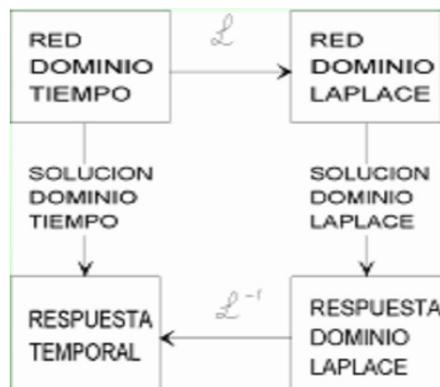
INTRODUCCIÓN

En el análisis de circuitos eléctricos en el dominio del tiempo las relaciones resultantes entre sus variables son del tipo “Ecuaciones diferenciales lineales con coeficientes constantes”.

Al aplicar la transformada de Laplace (tiempo continuo) o la transformada Z (tiempo discreto), las relaciones entre las variables transformadas son del tipo “Ecuaciones algebraicas lineales”, lo que facilita su resolución.

La resolución del problema consiste en hallar una respuesta temporal a dicho problema siguiendo dos posibles para el caso del empleo de la transformada de Laplace (Ver figura 1).

Figura 1. Solución al problema



El principal objetivo del software diseñado bajo la Interfaz Gráfica de Matlab GUI es proveer al usuario de una herramienta eficaz y muy comprensible para llevar a cabo la transformación de dominio matemático requerida, ya sea de manera directa (Transformada de Laplace o Transformada Z) o de manera inversa (Transformada Inversa de Laplace o Transformada Z Inversa).

1. MARCO TEORICO

1.1 TRANSFORMADA DE LAPLACE

La transformada de la Laplace está definida de la siguiente forma:

Sean:

$f(t)$ = Una función del tiempo t tal que $f(t) = 0$ para $t < 0$.

s = Una variable compleja.

L = Un símbolo operativo que indica que la cantidad a la que antecede se va a

transformar mediante la integral de Laplace $\int_0^{\infty} e^{-st} dt$.

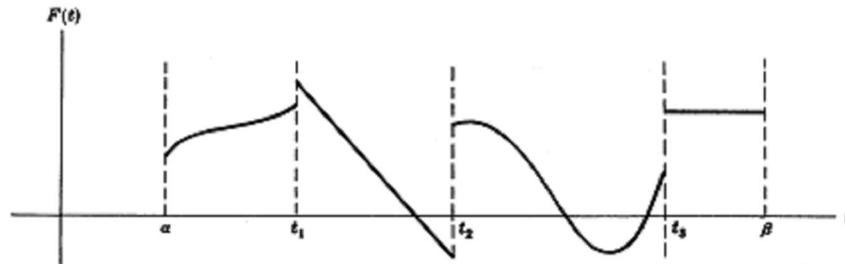
$F(s)$ = transformada de Laplace de $f(t)$.

La transformada de Laplace de $f(t)$ se obtiene mediante:

$$L\{f(t)\} = F(s) = \int_0^{\infty} f(t) e^{-st} dt$$

1.1.1 Continuidad seccional o a trazos. Una función es seccionalmente continua o continua a trazos en un intervalo $\alpha \leq t \leq \beta$ si es posible partir el intervalo en un número infinito de subintervalos de tal manera que la función sea continua en cada uno de ellos y tenga límites a izquierda y derecha (Ver figura 2).

Figura 2. Continuidad seccional a trazos



La función de la figura tiene discontinuidades en t_1 , t_2 y t_3 . Esta función es seccionalmente continua.

1.1.2 Funciones de orden exponencial. Si existen constantes reales $M > 0$ y γ tales que para todo $t > N$.

$$\left| e^{-\gamma t} F(t) \right| < M \quad \text{o} \quad \left| F(t) \right| < M e^{\gamma t}$$

Se dice que $F(t)$ es una función de orden exponencial γ cuando $t \rightarrow \infty$, o simplemente, que es una función de orden exponencial.

Ejemplos:

1. $F(t) = t^3$ es de orden exponencial 3 (por ejemplo) ya que $|t^3| = t^3 < e^{3t}$ para todo $t > 0$.

2. e^3 no es de orden exponencial ya que $\left| e^{-\gamma t} e^3 \right| = e^{3-\gamma t}$ puede hacerse más grande que cualquier constante al hacer crecer t .

1.1.3 Condiciones suficientes para la existencia de la transformada de Laplace. Teorema. Si $f(t)$ es seccionalmente continua en cada intervalo finito $0 \leq t \leq N$ y de orden exponencial γ para $t > N$, entonces existe la transformada de Laplace $F(s)$ para todo $s > \gamma$.

Otras condiciones son:

- $f(t)$ es seccionalmente continua en cualquier intervalo $N_1 \leq t \leq N$ donde $N_1 > 0$,
- $\lim_{t \rightarrow 0} t^n f(t) = 0$ para cualquier n tal que $0 < n < 1$,
- $f(t)$ es de orden exponencial γ para $t > N$.

1.1.4 Transformada de Laplace de funciones elementales

1.1.4.1 Función exponencial. Sea la función exponencial

$$f(t) = \begin{cases} Ke^{-\alpha t}, & \text{para } t \geq 0 \\ 0, & \text{para } t < 0 \end{cases}$$

donde K y α son constantes.

La transformada de Laplace de esta función exponencial se obtiene así:

$$\begin{aligned} L[Ke^{-\alpha t}] &= \int_0^{\infty} Ke^{-\alpha t} e^{-st} dt = K \int_0^{\infty} e^{-(\alpha+s)t} dt = -\frac{Ke^{-(\alpha+s)t}}{\alpha+s} \Big|_0^{\infty} \\ &= -\frac{Ke^{-(\alpha+s)\infty}}{s+\alpha} + \frac{Ke^{-(\alpha+s)0}}{s+\alpha} = \frac{K}{s+\alpha} \end{aligned}$$

1.1.4.2 Función escalón. Sea la función escalón

$$u(t) = \begin{cases} K, & \text{para } t \geq 0 \\ 0, & \text{para } t < 0 \end{cases}$$

en donde K es una constante. Su transformada es:

$$\begin{aligned} L\{u(t)\} &= U(s) = \int_0^{\infty} Ke^{-st} dt \\ &= -\frac{Ke^{-st}}{s} \Big|_0^{\infty} = -\frac{Ke^{-s(\infty)}}{s} + \frac{Ke^{-s(0)}}{s} \\ &= \frac{K}{s} \end{aligned}$$

1.1.4.3 Función rampa. Sea la función rampa

$$r(t) = \begin{cases} Kt, & \text{para } t \geq 0 \\ 0, & \text{para } t < 0 \end{cases}$$

en donde K es la pendiente de la recta t

$$L\{r(t)\} = \int_0^{\infty} Kte^{-st} dt = K \int_0^{\infty} te^{-st} dt$$

Usando la formula $\int xe^{ax} dx = \frac{e^{ax}}{a^2}(ax-1) + c$ tenemos

$$L\{r(t)\} = K \left[\frac{e^{-st}}{(-s)^2} (-st-1) \right]_0^{\infty} = \frac{K}{s^2}$$

1.1.4.4 Funciones seno y coseno. Sea la función seno

$$f(t) = \begin{cases} K \operatorname{sen} \omega t, & \text{para } t \geq 0 \\ 0, & \text{para } t < 0 \end{cases}$$

en donde K y ω son constantes.

Usando la formula de integración

$$\int e^{ax} \operatorname{sen} bx \, dx = \frac{e^{ax}}{a^2 + b^2} (a \operatorname{sen} bx - b \cos bx) + c$$

tenemos que la transformada es

$$\begin{aligned} L\{K \operatorname{sen} \omega t\} &= \int_0^{\infty} K e^{-st} \operatorname{sen} \omega t \, dt \\ &= \lim_{P \rightarrow \infty} K \int_0^P e^{-st} \operatorname{sen} \omega t \, dt \\ &= \lim_{P \rightarrow \infty} \left. \frac{K e^{-st} (-s \operatorname{sen} \omega t - a \cos \omega t)}{s^2 + \omega^2} \right|_0^P \\ &= \lim_{P \rightarrow \infty} \left\{ K \left[\frac{\omega}{s^2 + \omega^2} - \frac{e^{-sP} (-s \operatorname{sen} \omega P + \omega \cos \omega P)}{s^2 + \omega^2} \right] \right\} \\ &= \frac{K\omega}{s^2 + \omega^2} \quad \text{si } s > 0. \end{aligned}$$

De igual forma se puede verificar que

$$L\{K \cos \omega t\} = \frac{Ks}{s^2 + \omega^2}$$

1.1.4.5 Función impulso. Sea la función impulso definida así:

$$\delta(t) = \begin{cases} \infty & \text{para } t = 0 \\ 0 & \text{para } t \neq 0 \end{cases}$$

La transformada de Laplace de la función impulso es

$$L\{\delta(t)\} = 1$$

1.1.5 Propiedades y teoremas

1.1.5.1 Propiedad de linealidad. La transformada de Laplace es una transformación lineal, es decir, cumple con las siguientes expresiones:

- a. $L\{af(t)\} = aL\{f(t)\} = aF(s)$ donde a es un escalar,
b. $L\{\alpha f(t) + \beta g(t)\} = \alpha L\{f(t)\} + \beta L\{g(t)\} = \alpha F(s) + \beta G(s)$

1.1.5.2 Teoremas de translación

- Primer Teorema: Si a es un número real cualquiera, entonces:

$$L\{e^{\pm at} f(t)\} = F(s \mp a) \text{ donde } F(s) = L\{f(t)\}$$

- Segundo Teorema: si $\alpha > 0$ y $\beta > 0$ entonces

$$L\{f(t-\beta)u(t-\alpha)\} = (\beta - \alpha)e^{-\alpha s} L\{u(t)\} + e^{-\alpha s} L\{f(t)\}$$

1.1.5.3 Cambio de escala

$$L\{f(\alpha t)\} = \frac{1}{\alpha} F\left(\frac{s}{\alpha}\right)$$

1.1.5.4 Teorema del valor inicial. Si $f(t)$ y $f'(t)$ se pueden transformar por el método de Laplace y si existe el $\lim_{s \rightarrow \infty} sF(s)$, entonces

$$f(0) = \lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} sF(s)$$

1.1.5.5 Teorema del valor final. Este teorema nos permita calcular el límite de la señal $f(t)$ cuando $t \rightarrow \infty$ a partir de su transformada de Laplace mediante la relación $f(\infty) = \lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$

$$f(\infty) = \lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$$

Para utilizar le teorema, todos los polos de $sF(s)$ deben estar en el lado izquierdo del plano s.

1.2 TRANSFORMADA INVERSA DE LAPLACE

Si la transformada inversa de Laplace de una función $f(t)$ es $F(s)$, es decir, si $L\{f(t)\}=F(s)$, entonces $f(t)$ se denomina una transformada inversa de Laplace de $F(s)$ y se expresa

$$f(t) = \mathcal{L}^{-1}\{F(s)\}$$

donde \mathcal{L}^{-1} se llama el operador transformada inversa de Laplace.

1.2.1 Métodos para hallar la transformada inversa de Laplace. Existen varios métodos para determinar transformadas inversas de Laplace, algunos de los cuales son:

- Método de las fracciones parciales
- Método de las series
- Método de las ecuaciones diferenciales
- Método de la fórmula de inversión compleja también denominada formula integral de Bromwich.

Para el diseño de este software se empleo el método de las fracciones parciales.

1.2.1.1 Método de expansión en fracciones parciales. El proceso de efectuar un desarrollo en fracciones parciales se analiza en tres casos. En cada uno se supondrá que

$$F(s) = \frac{N(s)}{D(s)} = \frac{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}$$

donde el grado del polinomio D(s) debe ser mayor que el grado del polinomio N(s).

- CASO I: Raíces reales simples. Si se supone que F(s) puede escribirse como

$$F(s) = \frac{N(s)}{(s+\alpha)D_1(s)}$$

donde el número real $-\alpha$ no es una raíz de $D_1(s)$, entonces podemos escribir $F(s)$ en la forma

$$F(s) = \frac{K}{s+\alpha} + F_1(s)$$

Para determinar K , se multiplican ambos miembros de esta ecuación por $s+\alpha$, los que nos lleva a lo siguiente:

$$(s+\alpha)F(s) = K + (s+\alpha)F_1(s)$$
$$\frac{N(s)}{D_1(s)} = K + (s+\alpha)F_1(s)$$

Haciendo $s = -\alpha$, obtenemos una fórmula para K ; esto es,

$$K = (s+\alpha)F(s)|_{s=-\alpha} = \frac{N(s)}{D_1(s)}|_{s=-\alpha} = \frac{N(-\alpha)}{D_1(-\alpha)}$$

El problema de encontrar el desarrollo de fracciones parciales de $F(s)$, se reduce a encontrar el desarrollo en fracciones parciales de $F_1(s)$. Obviamente $F_1(s) = F(s) - K/(s+\alpha)$. Si $F_1(s)$ tiene un polo real simple, podemos repetir el proceso anterior.

-CASO II: Raíces complejas simples. Ya que las raíces complejas de un polinomio con coeficientes reales siempre aparecen en pares conjugados podemos suponer que $F(s)$ tiene la forma

$$F(s) = \frac{N(s)}{(s+\alpha+j\beta)(s+\alpha-j\beta)D_1(s)}$$

Donde $-\alpha-j\beta$ y $-\alpha+j\beta$ no son raíces de $D_1(s)$. Entonces podemos escribir $F(s)$ como

$$F(s) = \frac{K}{s+\alpha+j\beta} + \frac{K^*}{s+\alpha-j\beta} + F_1(s)$$

donde se determina K y su complejo conjugado K^* .

Una vez que se han determinado K y K^* , pueden combinarse los dos términos correspondientes de la siguiente forma: Supongamos que $K = x + jy$ entonces $K^* = x - jy$.

Luego de un proceso matemático simple podemos determinar que

$$\frac{K}{s+\alpha+j\beta} + \frac{K^*}{s+\alpha-j\beta} = \frac{x+jy}{s+\alpha+j\beta} + \frac{x-jy}{s+\alpha-j\beta} = 2x \frac{(s+\alpha)}{(s+\alpha)^2 + \beta^2} + 2jy \frac{\beta}{(s+\alpha)^2 + \beta^2}$$

Esta ecuación es empleada en el proceso cuando la función tiene raíces complejas. De esta forma podemos obtener fracciones parciales sin números imaginarios y así obtener una transformada inversa de Laplace mucho más fácil de identificar.

-CASO III: Raíces reales múltiples. Supongamos que $F(s)$ tiene la forma

$$F(s) = \frac{M(s)}{(s+s_0)^n D_1(s)}$$

donde $-s_0$ no es una raíz de $D_1(s)$ y en general es compleja. Entonces podemos escribir $F(s)$ como

$$F(s) = \frac{K_1}{s+s_0} + \frac{K_2}{(s+s_0)^2} + \dots + \frac{K_{n-1}}{(s+s_0)^{n-1}} + \frac{K_n}{(s+s_0)^n} + F_1(s)$$

Multiplicando los dos miembros de esta ecuación por $(s+s_0)^n$ se tiene como resultado:

$$(s+s_0)^n F(s) = K_1(s+s_0)^{n-1} + K_2(s+s_0)^{n-2} + \dots + K_{n-1}(s+s_0) + K_n + (s+s_0)^n F_1(s)$$

Haciendo $s = -s_0$, se obtiene

$$K_n = (s+s_0)^n F(s) \Big|_{s=-s_0}$$

Para encontrar K_{n-1} después de la multiplicación por $(s+s_0)^n$ se derivan ambos miembros con respecto a S. Entonces

$$\begin{aligned} & \frac{d}{ds} [(s+s_0)^n F(s)] = \\ & = (n-1)K_1(s+s_0)^{n-2} + (n-2)K_2(s+s_0)^{n-3} + \dots \\ & \quad + 2K_{n-2}(s+s_0) + K_{n-1} + 0 + (s+s_0)^n \frac{dF_1(s)}{ds} + n(s+s_0)^{n-1} F(s) \end{aligned}$$

Haciendo $s = -s_0$ en esta ecuación, se obtiene

$$K_{n-1} = \frac{d}{ds} [(s+s_0)^n F(s)]_{s=-s_0}$$

Repitiendo este proceso,

$$2K_{n-2} = \frac{d^2}{ds^2} [(s+s_0)^n F(s)]_{s=-s_0}$$

por lo que,

$$K_{n-2} = \frac{1}{2} \frac{d^2}{ds^2} [(s+s_0)^n F(s)]_{s=-s_0}$$

1.3 TRANSFORMADA Z

El método de la transformada z es un método muy útil cuando se trabaja con sistemas en tiempo discreto.

Al considerar la transformada z de una función del tiempo $x(t)$, solo se toman en cuenta los valores muestreados de $x(t)$, esto es, $x(0), x(T), x(2T), \dots$, donde T es el periodo de muestreo.

La transformada z de una función del tiempo $x(t)$, donde t es positivo, o de la secuencia de valores $x(kT)$, donde k adopta valores de cero o positivos y T es el periodo de muestreo, se define mediante la siguiente ecuación:

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

1.3.1 Transformada z de funciones elementales

1.3.1.1 Función escalón unitario. Si la función escalón unitario está definida como

$$\begin{cases} 1(t), 0 \leq t \\ 0, t < 0 \end{cases}$$

de esta forma se tiene que.

$$X(z) = \sum_0^{\infty} z^{-k} = 1 + z^{-1} + z^{-2} + z^{-3} + \dots = \frac{z}{z-1}$$

1.3.1.2 Función rampa unitaria. Observe que $x(kT) = kT, k=0,1,2,\dots$

La transformada z de la función rampa unitaria se puede escribir como

$$X(z) = T \sum_0^{\infty} k z^{-k} = T(z^{-1} + 2z^{-2} + 3z^{-3} + \dots) = \frac{Tz}{(z-1)^2}$$

1.3.1.3 Función polinomial a^k

$$\begin{cases} a^k, k=0,1,2,\dots \\ 0, k < 0 \end{cases}$$

donde a es una constante. Su transformada Z está dada por

$$X(z) = \sum_0^{\infty} a^k z^{-k} = 1 + a z^{-1} + a^2 z^{-2} + a^3 z^{-3} + \dots = \frac{z}{z-a}$$

1.3.1.4 Función exponencial. Puesto que $x(kT) = e^{-akT}, k=0,1,2,\dots$ se tiene que

$$X(z) = \sum_0^{\infty} e^{-akT} z^{-k} = 1 + e^{-aT} z^{-1} + e^{-2aT} z^{-2} + e^{-3aT} z^{-3} + \dots = \frac{z}{z - e^{-aT}}$$

1.3.1.5 Función senoidal. Si tenemos en cuenta que $\text{sen} \omega t = \frac{1}{2j}(e^{j\omega t} - e^{-j\omega t})$ y la

transformada z de la función exponencial es $Z[e^{-at}] = \frac{1}{1 - e^{-aT} z^{-1}}$ podemos encontrar que:

$$\begin{aligned} \mathcal{Z}[\sin \omega T] &= \frac{1}{2j} \left(\frac{1}{1 - e^{j\omega T} z^{-1}} - \frac{1}{1 - e^{-j\omega T} z^{-1}} \right) \\ &= \frac{1}{2j} \frac{(e^{j\omega T} - e^{-j\omega T}) z^{-1}}{1 - (e^{j\omega T} + e^{-j\omega T}) z^{-1} + z^{-2}} = \frac{z \sin \omega T}{z^2 - 2z \cos \omega T + 1} \end{aligned}$$

1.3.2 Propiedades y teoremas importantes

1.3.2.1 Multiplicación por a^k . Si $X(z)$ es la transformada z de $x(k)$, entonces la transformada z de $a^k x(k)$ está dada por $X(a^{-1}z)$:

$$\mathcal{Z}[a^k x(k)] = X(a^{-1}z)$$

1.3.2.2 Teorema de traslación compleja. Si $x(t)$ tiene la transformada z $X(z)$, entonces la transformada z de $e^{-at}x(t)$ está dada por $X(ze^{-aT})$.

$$\mathcal{Z}[e^{-at} x(t)] = \sum_{k=0}^{\infty} x(kT) e^{-akT} z^{-k} = \sum_{k=0}^{\infty} x(kT) (ze^{aT})^{-k} = X(ze^{aT})$$

1.3.2.3 Teorema del valor inicial. Si $x(t)$ tiene la transformada $zX(z)$ y si el $\lim_{z \rightarrow \infty} X(z)$ existe, entonces el valor inicial $x(0)$ de $x(t)$ o $x(k)$ está dado por

$$x(0) = \lim_{z \rightarrow \infty} X(z)$$

1.3.2.4 Teorema del valor final. Si $x(k)$, donde $x(k)=0$ para $k < 0$, tiene la transformada $zX(z)$ y que todos los polos de $X(z)$ están dentro del círculo unitario, con la posible excepción de un solo polo en $z=1$. Entonces el valor final de $x(k)$, esto es, el valor de $x(k)$ a medida que k tiende a infinito, puede darse mediante

$$\lim_{k \rightarrow \infty} x(k) = \lim_{z \rightarrow 1} \left[(1 - z^{-1}) X(z) \right]$$

1.4 TRANSFORMADA Z INVERSA

Cuando $X(z)$, la transformada z de $x(kT)$ o $x(k)$, esta dada, la operación que determina la $x(kT)$ o $x(k)$ correspondiente se denomina *transformación z inversa*.

Un método obvio para encontrar la transformada z inversa es referirse a una tabla de transformadas z . Sin embargo, a menos que uno se refiera a una tabla de

transformadas z muy extensa, no sería capaz de encontrar la transformada z inversa de una función de z complicada.

Existen otros cuatro métodos para obtener la transformada z inversa que no implican el uso de tablas:

- Método de la división directa
- Método computacional
- Método de expansión en fracciones parciales
- Método de la integral de inversión

El software ha sido diseñado para el modulo de Transformada Z Inversa de tal forma que permita al usuario emplear dos métodos distintos: Método de expansión en fracciones parciales y Método computacional.

1.4.1 Método de expansión en fracciones parciales. Es idéntico al que se utiliza en la transformada de Laplace, y requiere que todos los términos de la expansión en fracciones parciales se puedan reconocer fácilmente en la tabla de pares de transformadas z . Si $X(z)$ tiene uno o más ceros en el origen ($z = 0$), entonces $\frac{X(z)}{z}$ o $X(z)$ se expande en la suma de términos sencillos de primer o segundo orden mediante la expansión en fracciones parciales, y se emplea una tabla de transformadas z para encontrar la función del tiempo correspondiente para cada uno de los términos expandidos.

Un procedimiento de uso muy común para los casos donde todos los polos son diferentes y hay por lo menos un cero en el origen es dividir ambos miembros de $X(z)$ entre z y entonces expandir $X(z)/z$ en fracciones parciales. Una vez que $X(z)/z$ se ha expandido, ésta será de la forma:

$$\frac{X(z)}{z} = \frac{a_1}{z-p_1} + \frac{a_2}{z-p_2} + \dots + \frac{a_n}{z-p_n}$$

El coeficiente a_i se puede determinar multiplicando ambos miembros de esta última ecuación por $z-p_i$ y haciendo que $z=p_i$. Esto dará como resultado que todos los términos del segundo miembro sean cero excepto el término a_i , en el cual el factor que está multiplicando $z-p_i$, ha sido cancelado por el denominador. Por lo tanto, se tiene:

$$a_i = \left[(z-p_i) \frac{X(z)}{z} \right]_{z=p_i}$$

Hay que observar que dicha forma de determinar a_i es válida sólo para polos simples.

Si $X(z)/z$ involucra un polo múltiple, por ejemplo, un polo doble en $z = p_i$ y no tiene mas polos, entonces $X(z)/z$ tendrá la forma

$$\frac{X(z)}{z} = \frac{c_1}{(z - p_i)^2} + \frac{c_2}{z - p_i}$$

Los coeficientes c_1 y c_2 se determinan a partir de:

$$c_1 = \left[(z - p_i)^2 \frac{X(z)}{z} \right]_{z=p_i}$$

$$c_2 = \left\{ \frac{d}{dz} \left[(z - p_i)^2 \frac{X(z)}{z} \right] \right\}_{z=p_i}$$

1.4.2 Método computacional. En este método, la transformada z inversa se obtiene utilizando la función delta de Kronecker $\delta_0(kT)$, donde

$$\delta_0(kT) = \begin{cases} 1 & \text{para } k = 0 \\ 0 & \text{para } k \neq 0 \end{cases}$$

Suponiendo que $u(k)$, la entrada al sistema $G(z)$ es la entrada Delta de Kronecker, la transformada z de la entrada delta de Kronecker es $U(z) = 1$. Lo vemos con un ejemplo, con el enfoque de la ecuación en diferencias para la ecuación.

$$G(z) = \frac{X(z)}{U(z)} = \frac{.4673z - .3393}{z^2 - 1.5327z + .6607}$$

Dicha ecuación se puede escribir como

$$(z^2 - 1.5327z + .6607) X(z) = (.4673z - .3393) U(z)$$

Esta ecuación se puede convertir en una ecuación en diferencias como sigue:

$$x(k+2) - 1.5327 x(k+1) + .6607 x(k) = .4673 u(k+1) - .3393 u(k)$$

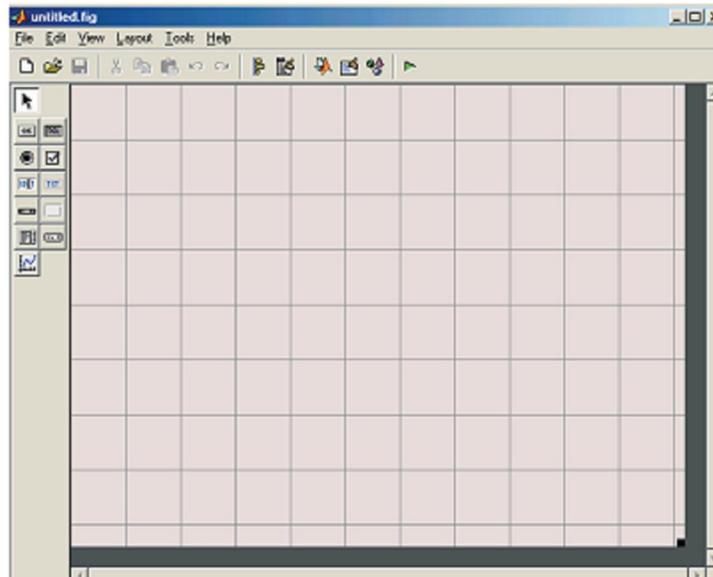
[$u(k)$ es la entrada Delta de Kronecker]. Encontrar la transformada z inversa se convierte ahora en el problema de resolver la ecuación en diferencias para $u(k)$

1.5 PROGRAMACIÓN VISUAL CON MATLAB

1.5.1 Breve descripción de ambiente de desarrollo de interfaz de usuario. El ambiente de desarrollo de Interfaz de Usuario en Matlab (GUIDE) contiene un conjunto de herramientas para crear interfaces graficas muy parecidas a las aplicaciones Windows. Estas herramientas simplifican el proceso de creación y de programación.

Cuando se abre un GUI con GUIDE, Matlab despliega el editor de diseño (Layout), (Ver figura 3).

Figura 3. Editor de diseño

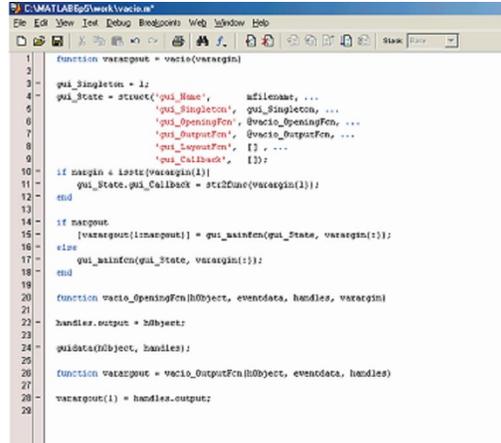


En el área de diseño, también conocida como formulario se pueden colocar los elementos que se encuentran en las herramientas como un botón de comando (push button), menús despegables (popupmenu), ejes (axes), etc.

Cuando se guarda un diseño grafico creado en la ventana GUIDE, Matlab crea automáticamente el archivo "m" correspondiente, el cual tendrá el mismo nombre que el creado en GUIDE, el cual tiene la extensión fig.

El archivo .m contiene los *callback* correspondientes a cada componente (Ver figura 4)

Figura 4. Archivo .m de un formulario vacío

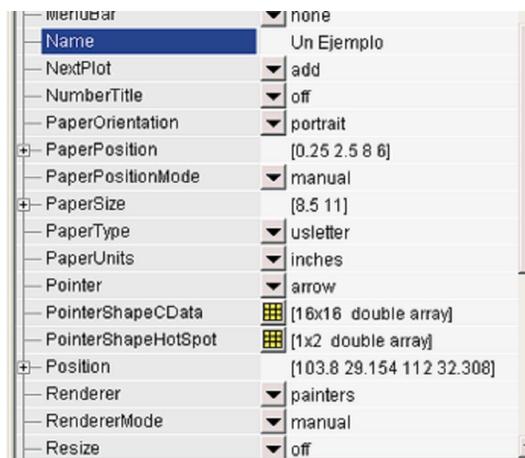


```
1 function hargout = vacio(varargin)
2
3
4 gui_Singleton = 1;
5 gui_State = struct('gui_Name',       sfileaschar(...
6                   'gui_Singleton',  gui_Singleton, ...
7                   'gui_OpeningFcn', @vacio_OpeningFcn, ...
8                   'gui_OutputFcn',  @vacio_OutputFcn, ...
9                   'gui_Callback',   []);
10
11 if nargin & ~isempty(varargin{1})
12     gui_State.gui_Callback = str2func(varargin{1});
13 end
14
15 if nargin
16     [hargout, [errmsg;errmsg]] = gui_mainfcn(gui_State, varargin{:});
17 else
18     gui_mainfcn(gui_State, varargin{:});
19 end
20
21 function vacio_OpeningFcn(hObject, eventdata, handles, varargin)
22
23 handles.output = hObject;
24
25 guidata(hObject, handles);
26
27 function hargout = vacio_OutputFcn(hObject, eventdata, handles)
28
29 hargout(1) = handles.output;
```

1.5.2 Cuadro de herramientas. El cuadro de herramientas contiene los controles que se pueden utilizar en una interfaz grafica de usuario. Estos componentes son objetos uicontrols de Matlab y son programables a través de los callback (las funciones que se ejecutan cuando el usuario activa algún componente en la GUI). Los componentes del cuadro de herramientas que son de interés son:

1.5.2.1 Push Buttons. Los botones de comando generan una acción cuando son presionados. Cuando se hace clic en un botón de comando, da la apariencia de ser presionado, cuando se deja de hacer clic tiene apariencia normal (no presionado). En el editor de propiedades se puede cambiar la apariencia y texto predefinido, en este caso solo es de interés el tamaño y el texto que tiene por *default* (Ver figura 5).

Figura 5. Editor de propiedades



1.5.2.2 Radio Buttons. *Radio button* es similar a *check boxes*, con la diferencia que los *radio buttons* se tienen que excluir mutuamente de entre un grupo de estos botones. Para activar un radio button se hace clic sobre la opción que se desea seleccionar. Estos botones tienen dos estados, seleccionado y no seleccionado y se observa en la propiedad Value, si esta en Max, el botón esta seleccionado, si su valor es cero, no esta seleccionado. En el siguiente texto, se supone que se tienen cuatro radio buttons los cuales serán excluidos.

```
function radiobutton1_Callback(hObject,eventdata,handles)  
off=[handles.radiobutton2,handles.radiobutton3,handles.radiobutton4];  
set(off,Value,0)  
%continuaelcallback
```

Posteriormente se hace algo similar para cuando se seleccione alguno de los otros *radio buttons*, cambiando la numeración de los botones.

1.5.2.3 Check Boxes. Esta opción es similar a la anterior, pero no se necesita excluir mutuamente los objetos checkboxes utilizados.

1.5.2.4 Edit text. Estos controles son campos en los cuales los usuarios pueden modificar las cadenas de texto. Use el editor de texto cuando desea que ingresen texto o datos.

1.5.2.5 Pop up menús. Despliegue de opciones cuando se hace clic en la flecha que tiene a un lado. Para agregar elementos a la lista, se selecciona el botón que se encuentra aun lado de la opción *string* del cuadro de propiedades y se introduce uno a uno cada elemento; dicho botón esta entre *string* y lo que se pondrá. Para programar este elemento, es necesario tener presente las bifurcaciones o condicionales, en especial la instrucción *switch*.

1.5.2.6 Menús. En su barra de herramientas se tienen varias opciones, las opciones de interés son nuevo menú, nuevo elemento y eliminar, el cual es el ultimo. La instrucción *callback* se emplea para hacer llamada a su correspondiente *callback*.

2. PROCEDIMIENTO PARA EL DESARROLLO DEL SOFTWARE

2.1 CREACIÓN DE AMBIENTE

En este aspecto el software diseñado se divide en cinco módulos claramente identificables:

2.1.1 Modulo de transformada inversa de Laplace. Este modulo está conformado inicialmente por dos cuadros de edición de texto, en los cuales se pide al usuario que ingrese el numerador y el denominador de la función la cual será objeto de la Transformada Inversa de Laplace (Ver figura 6). Dicha función deberá ser introducida como vector, tanto el numerador como el denominador.

Figura 6. Vista general modulo Transformada Inversa de Laplace

The screenshot shows a software window titled "Transformacion de Dominio Matematico" with a "Teoremas" menu. A dropdown menu is set to "Transformada Inversa de Laplace". The main interface is divided into several sections:

- Input Section:** Two columns for "Ingrese el Numerador como vector" and "Ingrese el Denominador como vector". The numerator input contains "1" and the denominator input contains "1 2". Below these are two more input fields for "Introduzca r" (containing "1") and "Introduzca p" (containing "-2"), with an "Introducir datos" button below them.
- Case Section:** A box labeled "CASO I" with the text "Raices reales simples".
- Output Section:** A box titled "Los archivos generados son los siguientes:" containing two sub-sections:
 - Resultado.mat:** "Este archivo almacena dos variables: -SISTEMA. Representa el sistema introducido en funcion de transferencia -RESULTADO. Representa la transformada inversa de laplace del sistema".
 - Fparciales.mat:** "Este archivo almacena dos variables: -FPARCIALESS. Representa las fracciones parciales del sistema introducido -FPARCIALEST. Representa la transformada inversa de laplace de cada fraccion parcial del sistema".

Cuando la función es introducida en los cuadros de edición de texto, se habilitará el control tipo *pushbutton* "Ejecutar acción". Cuando el usuario da clic en dicho control, se inicia el proceso mediante el cual se calcula la Transformada Inversa de Laplace.

Ya que la programación de una Interfaz Gráfica en Matlab (GUI) se basa en el control de eventos, es posible que el usuario solicite el comienzo del proceso sin haber introducido la función. Al producirse este evento, el sistema deberá dar aviso al usuario solicitándole que ingrese la función para iniciar el proceso.

Al dar inicio al proceso de forma correcta (ingresando la función previamente), el software determina que tipo de raíces posee el denominador de la función. Si las raíces son reales, se harán visibles dos nuevas casillas de edición de texto donde se le solicitará al usuario que introduzca los valores de " r " y " p " de las fracciones parciales los cuales son presentados previamente en casillas de texto estático. Por ultimo el software resuelve el sistema y presenta las respuestas en el *workspace* de Matlab. Al mismo tiempo genera dos archivos los cuales contienen dichas respuestas. La descripción del contenido de cada archivo se presenta en el formulario del modulo actual enmarcada en dos *frames* junto con el nombre de cada archivo generado.

Para reiniciar el sistema y realizar una nueva operación, el usuario deberá seleccionar nuevamente la opción "Transformada Inversa de Laplace"

2.1.2 Modulo de transformada z inversa. El modulo de Transformada Z Inversa es presentado ante el usuario con dos casillas de edición de texto donde el usuario debe introducir el numerador y el denominador como vector y en función de "Z", del sistema que será objeto del proceso de Transformada Inversa Z (Ver figura 7).

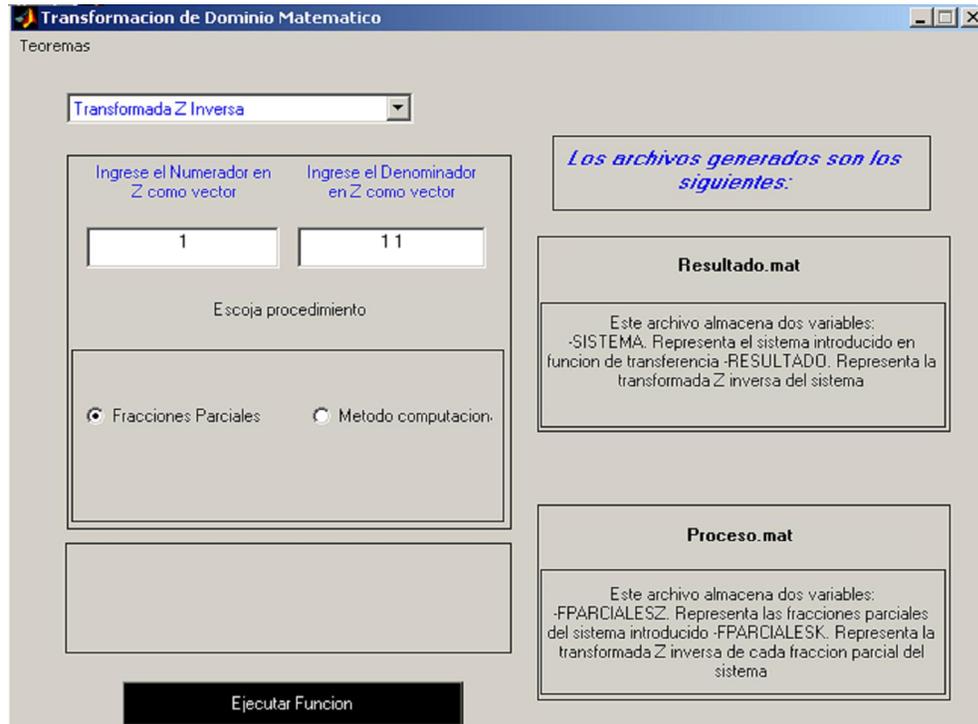
Seguidamente el usuario deberá escoger el procedimiento que le sea conveniente. Para tal fin se presentan 2 controles del tipo *radio buttons* etiquetados con los procedimientos *Fracciones Parciales* y *Método Computacional*.

Luego de que la función se ingresa al sistema y se escoge el procedimiento, el usuario podrá dar inicio al proceso presionando el control tipo *push button* etiquetado como "Ejecutar Función". Si el *push button* es presionado sin que la función haya sido introducida o sin que el usuario haya escogido el procedimiento a seguir, el sistema dará indicación del problema y el usuario podrá corregir dicha falla.

Al dar inicio al proceso de la forma correcta ya mencionada, el sistema calculará la Transformada Inversa Z y presentará las respuestas en el *workspace* de Matlab. Al mismo tiempo genera dos archivos los cuales contienen dichas respuestas. La descripción del contenido de cada archivo se presenta en el formulario del modulo actual enmarcada en dos *frames* junto con el nombre de cada archivo generado.

Para reiniciar el sistema y realizar una nueva operación, el usuario deberá seleccionar nuevamente la opción "Transformada Z Inversa"

Figura 7. Vista general modulo Transformada Z Inversa



2.1.3 Modulo de transformada de Laplace. El modulo de Transformada de Laplace mas allá de ser una herramienta para el calculo de transformadas de Laplace se presenta al usuario como un modulo didáctico el cual genera una respuesta simbólica de tal manera que el usuario pueda observar la forma en que se aplican los distintos teoremas (Ver figura 8).

En principio el modulo consta de un control del tipo *toggle button* el cual se encuentra etiquetado como "*Funciones predeterminadas*". Al ser presionado, se hace visible un grupo de opciones las cuales corresponden a funciones en el tiempo, para las cuales el sistema está habilitado para encontrar su transformada de Laplace. Así mismo se hace visible un control del tipo *popup menu* etiquetado como "*Relación entre funciones*". Como su nombre lo muestra, dicho control le indica al usuario la forma en que pueden ser relacionadas las funciones predeterminadas; ya sean suma, resta o multiplicación de funciones.

Para este modulo el software ha sido diseñado para que el usuario este habilitado para seleccionar hasta dos funciones, relacionadas todas entre sí de la misma forma, ya sea suma, resta o multiplicación.

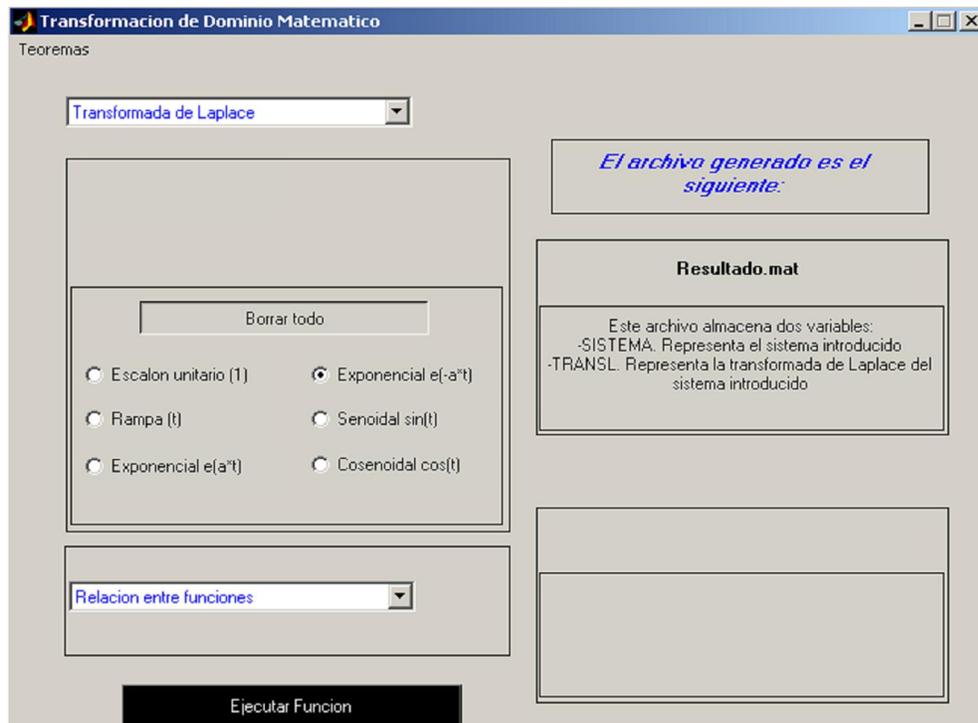
Para iniciar el proceso de forma correcta, el usuario deberá escoger al menos una función y luego presionar el control del tipo *pushbutton* etiquetado como "*Ejecutar*

Función". De lo contrario el software emitirá un mensaje informando al usuario del error. Si el usuario no selecciona ninguna relación entre funciones del *popup menu*, el software por defecto las relacionara como suma de funciones.

Al dar inicio al proceso de la forma correcta ya mencionada, el sistema obtendrá la Transformada de Laplace y presentará las respuestas en el *workspace* de Matlab. Al mismo tiempo genera un archivo el cual contienen dichas respuestas. La descripción del contenido del archivo se presenta en el formulario del modulo actual enmarcada en un *frame* junto con el nombre del archivo generado.

Para reiniciar el sistema y realizar una nueva operación, el usuario deberá seleccionar nuevamente la opción "Transformada de Laplace"

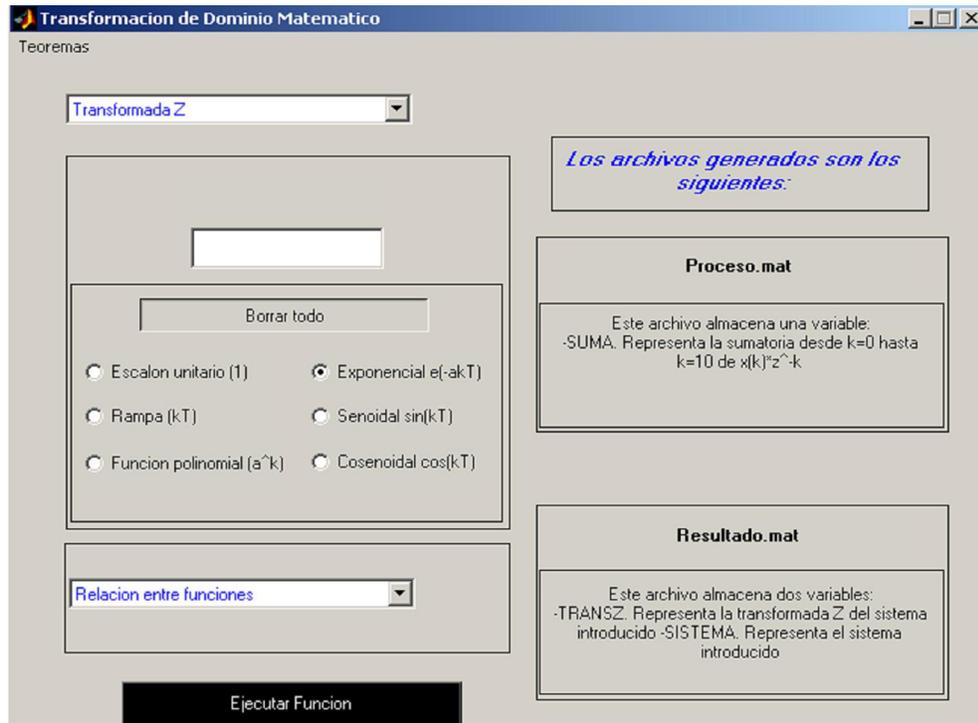
Figura 8. Vista general modulo Transformada de Laplace



2.1.4 Modulo de transformada z. El modulo de Transformada Z se presenta al usuario inicialmente compuesto por un cuadro de edición de texto y un control del tipo *toggle button* etiquetado como "*Funciones Predeterminadas*"(ver figura 9). El usuario podrá introducir cualquier función en la casilla de edición de texto en términos de " $k \cdot T$ " como se lo indica el software y luego iniciar el proceso oprimiendo el *push button* "*Ejecutar Función*". Al presionar *toggle button* el usuario habrá tomado la determinación de resolver un sistema basado en las funciones predeterminadas, por lo cual se hará visible un grupo de opciones las cuales

corresponden a funciones en tiempo discreto y un control del tipo *popupmenu* etiquetado como “*Relación entre funciones*”. De la misma manera el software quedará inhabilitado para resolver cualquier función que sea introducida en la casilla de edición de texto.

Figura 9. Vista general modulo Transforma Z



El diseño del software para este modulo le da la posibilidad al usuario de seleccionar hasta 6 funciones relacionadas todas entre si de la misma manera, ya sea suma, resta o multiplicación de funciones.

Para iniciar el proceso de forma correcta se debe hacer lo siguiente:

- Si se trabaja en el modo de casilla de edición de texto, el usuario deberá introducir la función antes de presionar el control “*Ejecutar Función*”, de lo contrario el sistema emitirá un mensaje de advertencia.

- Si se trabaja en el modo de Funciones Predeterminadas, el usuario deberá escoger al menos una función antes de presionar el control “*Ejecutar Función*”, de lo contrario el sistema emitirá un mensaje de advertencia.

Al dar inicio al proceso de la forma correcta ya mencionada, el sistema obtendrá la Transformada de Z y presentará las respuestas en el *workspace* de Matlab. Al mismo tiempo generará dos archivos los cuales contienen dichas respuestas. La

descripción del contenido de cada archivo se presenta en el formulario del modulo actual enmarcada en un *frame* junto con el nombre de cada archivo generado.

Para reiniciar el sistema y realizar una nueva operación, el usuario deberá seleccionar nuevamente la opción “Transformada Z”

2.1.5 Modulo de teoremas. Este modulo es básicamente una ayuda didáctica para el usuario. Está presente en la esquina superior izquierda del formulario como menú con la etiqueta *Teoremas*. Dicho menú contiene 2 opciones:

2.1.5.1 Transformada_Z: Cuando el usuario llama al menú y escoge la opción Transformada_Z se despliega un nuevo formulario el cual se presenta con un control tipo *popup menú* y una casilla de texto estática. El *popup menú* contiene una lista de las principales propiedades y teoremas de la Transformada Z. Cada propiedad o teorema será desplegada sobre el cuadro de texto estático una vez el usuario la seleccione.

2.1.5.2 Transformada_de_Laplace: Cuando el usuario llama al menú y escoge la opción Transformada_de_Laplace se despliega un nuevo formulario el cual se presenta con un control tipo *popup menú* y una casilla de texto estática. El *popup menú* contiene una lista de las principales propiedades y teoremas de la Transformada de Laplace. Cada propiedad o teorema será desplegada sobre el cuadro de texto estático una vez el usuario la seleccione.

2.2 PROGRAMACIÓN GENERAL

Una vez que el usuario ha ingresado la función objeto en cualquiera de los módulos del software y ejecuta el inicio del proceso interno en busca de una solución mediante la interacción con el control tipo *Push Button* etiquetado como “Ejecutar Función” se describe el siguiente procedimiento:

2.2.1 Transformada inversa de Laplace. El procedimiento empleado para hallar la transformada inversa de Laplace es Expansión en Fracciones Parciales para tal fin se llevan a cabo el análisis de las raíces del denominador con el fin de determinar si se trata de raíces reales o raíces complejas. Para cada caso se realiza lo siguiente:

2.2.1.1 Raíces reales. Si se determina que las raíces del denominador de la función objeto de análisis son reales se iniciará un proceso iterativo, el cual busca la conformación de las fracciones parciales de dicha función con la ayuda del comando “*residue*” de Matlab.

Una vez realizado este proceso se encuentra la transformada inversa de Laplace de cada una de las fracciones parciales encontradas con ayuda del comando

“*ilaplace*” de Matlab y a su vez se conforma una solución total sumando cada una de las transformadas inversas de cada fracción parcial.

Finalmente, el resultado de este proceso se guarda en dos archivos (ver tabla 1).

Los archivos son almacenados en el directorio de trabajo actual de Matlab y con ayuda del comando “*fprintf*” se muestra en el *workspace* el contenido de las distintas variables.

2.2.1.2 Raíces complejas simples. Si se determina que las raíces del denominador de la función objeto de análisis son complejas y simples se iniciará un proceso iterativo el cual busca la conformación de las fracciones parciales de dicha función con la ayuda del comando “*residue*” de Matlab.

Seguidamente y ya que las raíces complejas siempre se dan en pares conjugados, se procede a encontrar los valores de X , Y , α y β que permitan la conformación de fracciones parciales que no contengan números imaginarios y así hallar de manera sencilla su transformada inversa de Laplace.

Como siguiente paso se halla la transformada inversa de Laplace de cada una de las fracciones parciales encontradas. Este proceso se logra con la ayuda del comando “*ilaplace*” de Matlab y a su vez se conforma una solución total sumando cada una de las transformadas inversas de cada fracción parcial.

Finalmente, el resultado obtenido de este proceso se guarda en dos archivos (ver tabla 1).

Tabla 1. Descripción de archivos Transformada inversa de Laplace

Archivo	Contenido	Descripción
Fparciales.mat	FparcialesS	Variable que contiene la expansión en fracciones parciales de la función objeto
	FparcialesT	Variable que contiene la transformada inversa de Laplace de cada una de las fracciones parciales
Resultado.mat	Resultado	Variable que contiene la transformada inversa de Laplace de la función objeto
	Sistema	Variable que contiene la representación en terminos de "S" de la función objeto

Los archivos son almacenados en el directorio de trabajo actual de Matlab y con ayuda del comando “*fprintf*” se muestra en el *workspace* el contenido de las distintas variables.

2.2.2 Transformada z inversa. El software ha sido diseñado para dar la posibilidad al usuario de escoger entre dos diferentes métodos para calcular la transformada Z Inversa. Estos métodos son los siguientes:

2.2.2.1 Expansión en fracciones parciales. El método de expansión en fracciones parciales que se presenta aquí es idéntico al método de expansión en fracciones parciales que se utiliza en la transformada inversa de Laplace. El método requiere que todos los términos de la expansión se puedan reconocer fácilmente en la tabla de pares de transformada z.

Inicialmente se arman las fracciones parciales mediante un proceso iterativo y con ayuda del comando *"residue"* de Matlab. Ya que dichas fracciones corresponden a la expansión de $X(z)/z$, se procede a multiplicar cada fracción por "z" y a obtener su transformada z inversa con la ayuda del comando *"iztrans"* de Matlab.

En el proceso de calcular la transformada Z inversa y de la misma manera que en la transformada inversa de Laplace, se lleva a cabo un análisis del denominador con el fin de determinar la naturaleza de sus raíces. Si en dicho análisis se encuentran raíces imaginarias, se procederá a calcular los valores respectivos de α , β que permitan la conformación de fracciones parciales que no contengan números imaginarios y así hallar de manera sencilla su transformada Z inversa.

Finalmente, el resultado obtenido de este proceso se guarda en dos archivos (ver tabla 2).

Tabla 2. Descripción de archivos Transformada Z Inversa (Fracciones parciales)

Archivo	Contenido	Descripción
Proceso.mat	FparcialesZ	Variable que contiene la expansión en fracciones parciales de la función objeto
	FparcialesK	Variable que contiene la transformada Z inversa de cada una de las fracciones parciales
Resultado.mat	Resultado	Variable que contiene la transformada Z inversa de la función objeto
	Sistema	Variable que contiene la representación en términos de "Z" de la función objeto

Los archivos son almacenados en el directorio de trabajo actual de Matlab y con ayuda del comando *"fprintf"* se muestra en el *workspace* el contenido de las distintas variables.

2.2.2.2 Método computacional. Si el sistema introducido es de la forma $Y(z)/X(z)$, se toma como entrada $X(z)$ al sistema la transformada z de la delta de Kronecker, la cual es igual a la unidad, de esta forma empleando el comando "filter" de Matlab se calcula la respuesta del sistema ante dicha entrada, lo que para este caso corresponde a la transformada Z inversa del sistema inicial.

Finalmente, el resultado de este proceso se guarda en un archivo (ver tabla 3).

Tabla 3. Descripción de archivos Transformada Z inversa (Método computacional)

Archivo	Contenido	Descripción
Resultado.mat	Rkronecker	Variable que representa la respuesta del sistema ante una entrada delta de Kronecker
	Sistema	Variable que contiene la representación en términos de "Z" de la función objeto

Los archivos son almacenados en el directorio de trabajo actual de Matlab y con ayuda del comando "fprintf" se muestra en el *workspace* el contenido de las distintas variables.

2.2.3 Transformada de Laplace. Ya que la finalidad del modulo de transformada de Laplace es servir como herramienta didáctica de tal forma que ayude al usuario a identificar las distintas propiedades existentes a través del análisis de la respuesta la cual está dada en forma simbólica para facilitar esta tarea.

Debido a todo lo anterior el software se encarga de tomar la función seleccionada por el usuario en el menú de *Funciones Predeterminadas* y obtener su transformada de Laplace con ayuda del comando "laplace" de Matlab.

El cálculo de la transformada de Laplace se guarda en un archivo (ver tabla 4).

Tabla 4. Descripción de archivos transformada de Laplace

Archivo	Contenido	Descripción
Resultado.mat	TransL	Variable que representa la transformada de Laplace del sistema introducido
	Sistema	Variable que contiene la representación en términos de "S" de la función objeto

Los archivos son almacenados en el directorio de trabajo actual de Matlab y con ayuda del comando "fprintf" se muestra en el *workspace* el contenido de las distintas variables.

2.2.4 Transformada z. En este caso el software ha sido diseñado para permitir al usuario editar una función para ser introducida al software, así como también posibilita la selección de una función del menú predeterminado. De esta manera se lleva a cabo el siguiente procedimiento:

Cuando el usuario interactúa con el control principal tipo *Push Button* etiquetado como *Ejecutar Función* se determina si la opción de menú predeterminado está activada. De ser así, el software hallará la transformada Z de la función o funciones que el usuario ha escogido y no tomará en cuenta ninguna función que se encuentre en la casilla de edición de texto. Si la opción de menú predeterminado no está activada, el software hallará la transformada Z de la función que se encuentre en la casilla de edición de texto.

Finalmente, el resultado de este proceso se guarda en dos archivos (ver tabla 5).

Tabla 5. Descripción de archivos transformada Z

Archivo	Contenido	Descripción
Proceso.mat	Suma	Variable que representa la sumatoria desde k=0 hasta k=10 de x(k)*z ^{-k}
Resultado.mat	TransZ	Variable que contiene la transformada Z inversa de la función objeto
	Sistema	Variable que contiene la representación del sistema introducido

Por último con ayuda del comando *"fprintf"* se muestra en el workspace el contenido de las variables ya mencionadas.

2.3 VISUALIZACION DE DATOS

El software ha sido diseñado de tal manera que emplea el procedimiento de expansión en fracciones parciales, tanto para calcular la transformada inversa de Laplace como para la transformada inversa Z.

Si se toma como ejemplo la transformada de Laplace, la expansión en fracciones parciales para el caso de un sistema con raíces simples debe resultar como sigue

$$\frac{R(s)}{A(s)} = \frac{R(1)}{s-P(1)} + \frac{R(2)}{s-P(2)} + \dots + \frac{R(n)}{s-P(n)} + K(s)$$

Para armar dichas fracciones se hace uso de los ciclos *for*. Matlab presenta un inconveniente cuando se quieren realizar divisiones empleando los ciclos *for* en las que estén presentes los datos simbólicos, ya sea en el numerador o en el

denominador. En este caso se encuentra el dato simbólico “s” en los denominadores.

Gráficamente el problema es el siguiente:

$$-\frac{1}{2} \frac{s}{s^2+1} = -\frac{1}{2} \frac{s}{s^2 + \frac{1125899906842621}{1125899906842624}}$$

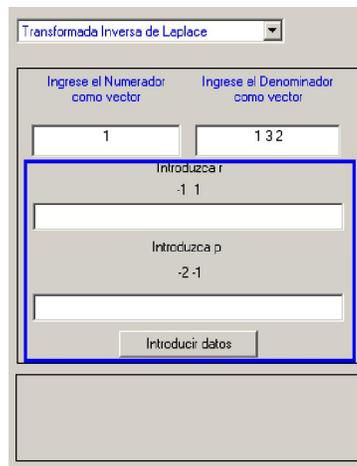
El miembro izquierdo de la ecuación representa una fracción parcial de un sistema en particular, y el lado derecho representa la versión obtenida durante la ejecución del programa donde se puede observar que el cociente planteado es igual a 1.

A medida que el ciclo *for* continúa “armando” las demás fracciones el problema se acentúa como se puede ver

$$\frac{1}{2} \frac{1}{s^2+1} = \frac{4503599627370507}{9007199254740992} \frac{1}{s^2 + \frac{1125899906842621}{1125899906842624}}$$

donde el nuevo cociente planteado es igual a $\frac{1}{2}$.

El ciclo *for* correspondiente emplea los valores *r*, *p* y *k* entregados por el comando *residue* de Matlab para armar las fracciones parciales. Se pudo determinar mediante ensayo y error que para el caso en que la función posee raíces reales, el hecho de tomar los valores de *r*, *p* y *k* de una casilla de edición de texto (ingresados por el usuario) soluciona el inconveniente. Por ésta razón se incluye en el diseño del software dichas casillas para los casos I y III de la transformada Inversa de Laplace (ver figura 10).



Sin embargo se determinó que la inclusión de dichas casillas era inútil si la función posee raíces complejas. Debido a esto, en estos casos la respuesta que se obtiene puede llegar a ser de gran tamaño, por lo cual se optó definitivamente por presentar los datos en el *workspace* de Matlab.

3. EJEMPLOS DE APLICACIÓN

3.1 EJEMPLO 1¹

Obtener la transformada z inversa de

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

MATLAB

```
>> syms z
```

```
>> sys = (z^2+z+2)/[(z-1)*( z^2-z+1)];
```

```
>> pretty(iztrans(sys))
```

```

      /      / / 1 \n      \
      | ----- | |-----|      ||
      | \      | \_alpha/      / 1 \n||
-2 charfcn[0](n) + 4 - 1/3 | ) | -4 ----- + 5 |-----| ||
      | /      \   _alpha      \_alpha/ /|
      | ----- |
      \_alpha = %1      /

      2
%1 := RootOf(_Z  - _Z + 1)

```

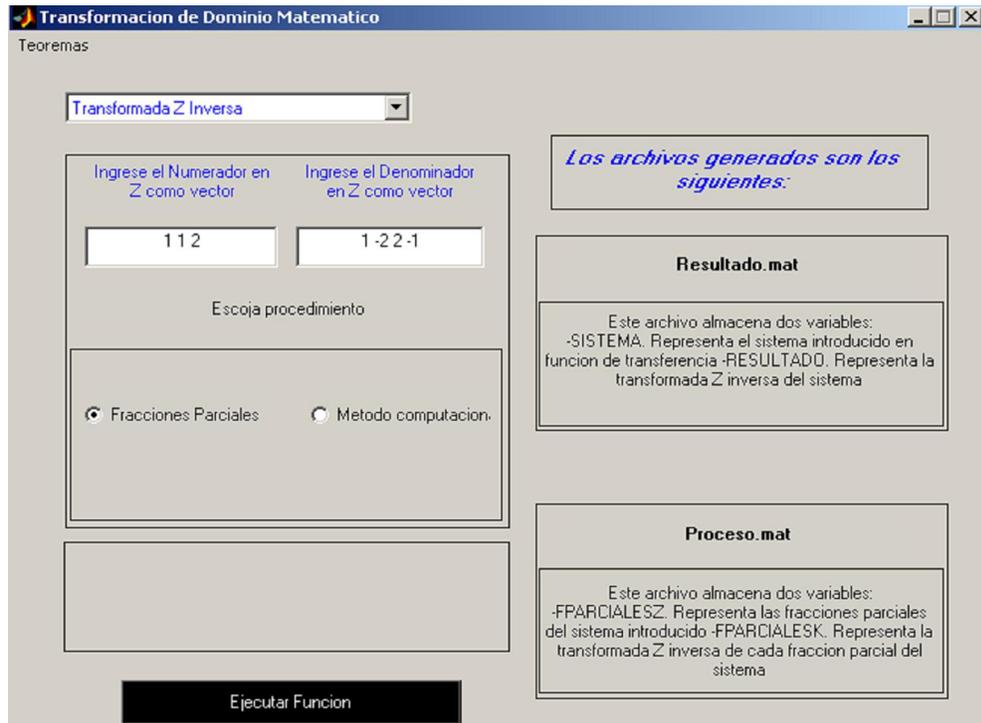
Como se puede observar, Matlab presenta inconvenientes en el momento de resolver el problema y la respuesta que entrega no es del todo comprensible.

- TRANSFORMACION DE DOMINIO

La función en cuestión deberá ser introducida como vector (ver figura11).

¹ KATSUHIKO, Ogata. Sistemas de Control en tiempo discreto, 1996. p.48

Figura 11. Ingreso al sistema, Ejercicio 1



Obteniéndose la siguiente respuesta:

LAS FRACCIONES PARCIALES CALCULADAS PARA EL SISTEMA SON LAS SIGUIENTES:

$$\begin{bmatrix} 4 & z - 1/2 & 4503599627370541 & 1 &] \\ [----- & -3 ----- & ----- & ----- &] \\ [z - 1 & 2 & 9007199254740992 & 2 &] \\ [& z - z + 1 & & z - z + 1 &] \end{bmatrix}$$

LA TRANSFORMADA Z INVERSA DE LAS FRACCIONES PARCIALES ES LA SIGUIENTE:

$$\begin{bmatrix} & & 325019307148085 & &] \\ [4 & -3 \cos(1/3 k \pi) & ----- & \sin(1/3 k \pi) &] \\ [& & 562949953421312 & &] \end{bmatrix}$$

LA TRANSFORMADA Z INVERSA DEL SISTEMA INGRESADO ES LA SIGUIENTE:

$$4 - 3 \cos(1/3 k \pi) + \frac{325019307148085}{562949953421312} \sin(1/3 k \pi)$$

3.2 EJEMPLO 2²

Hallar la transformada inversa de Laplace de:

$$V(s) = \frac{10s^2 + 15s - 5}{s^3 + 2s^2 + 5s}$$

MATLAB

```
>> syms s
```

```
>> sys = (10*s^2+15*s-5)/(s^3+2*s^2+5*s);
```

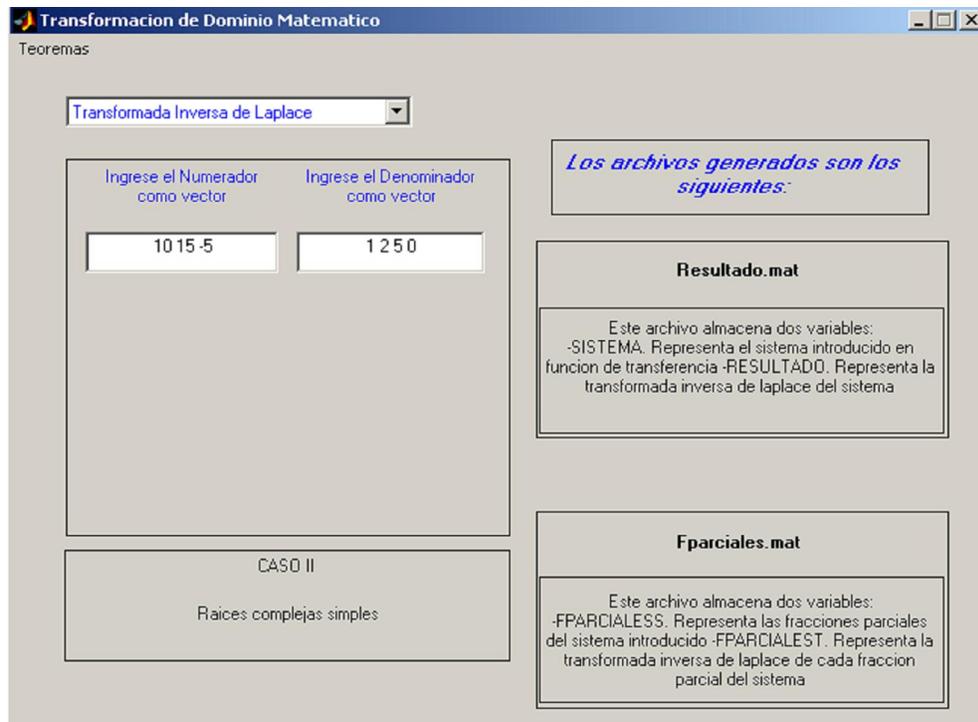
```
>> pretty(ilaplace(sys))
```

$$-1 + 11 \exp(-t) \cos(2 t) + 3 \exp(-t) \sin(2 t)$$

TRANSFORMACIÓN DE DOMINIO

De la misma forma el sistema debe ser ingresado en forma de vector (ver figura 12)

Figura 12. Ingreso al sistema Ejercicio 2



² TRONCOSO HERNÁNDEZ, Ricardo. Archivo SS_2_USCO.pdf. p. 108.

Obteniéndose la siguiente respuesta:

LAS FRACCIONES PARCIALES CALCULADAS PARA EL SISTEMA SON LAS SIGUIENTES:

$$\begin{bmatrix} s + 1 & 6 \\ -1/s & 11 \end{bmatrix} \begin{bmatrix} \frac{1}{(s + 1)^2 + 4} & \frac{1}{(s + 1)^2 + 4} \end{bmatrix}$$

LA TRANSFORMADA INVERSA DE LAPLACE DE LAS FRACCIONES PARCIALES ES LA SIGUIENTE:

$$[-1 \quad 11 \exp(-t) \cos(2 t) \quad 3 \exp(-t) \sin(2 t)]$$

LA TRANSFORMADA INVERSA DE LAPLACE DEL SISTEMA INGRESADO ES LA SIGUIENTE:

$$-1 + 11 \exp(-t) \cos(2 t) + 3 \exp(-t) \sin(2 t)$$

Como se puede observar, este sistema es una excepción a la regla, ya que no presenta problema alguno de visualización a pesar de que su denominador tiene un par de raíces imaginarias.

3.3 EJERCICIO 3

Obtener la transformada de Laplace de $e^{-2t} \cos 4t$

MATLAB

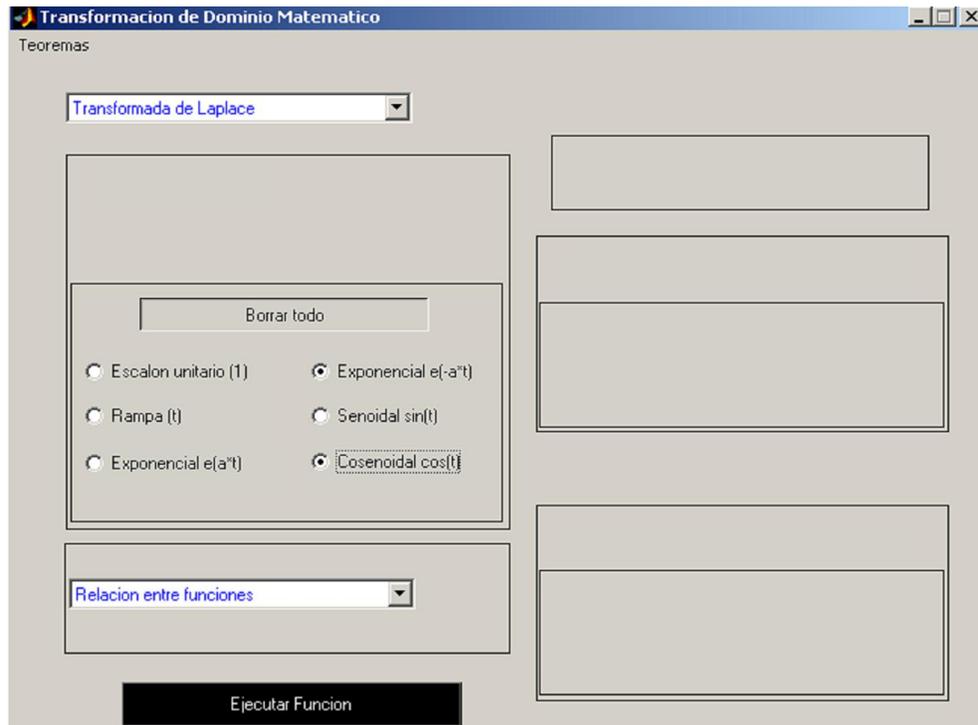
```
>> syms t
>> sys = exp(-2*t)*cos(4*t);
>> pretty(laplace(sys))
```

$$\frac{s + 2}{(s + 2)^2 + 16}$$

TRANSFORMACIÓN DE DOMINIO

Se deben introducir las funciones predeterminadas como producto (ver figura 13)

Figura 13. Ingreso al sistema Ejercicio 3



Obteniendo la siguiente respuesta:

EL SISTEMA INTRODUCIO FUE EL SIGUIENTE:

Sistema =

$\exp(-a*t) * \cos(a*t)$

LA TRANSFORMADA DE LAPLACE DEL SISTEMA INTRODUCIDO ES EL SIGUIENTE:

$$\frac{s + a}{(s + a)^2 + a^2}$$

En esta respuesta se plasma la filosofía didáctica del modulo ya que de su análisis podemos determinar la traslación que tiene la transformada de Laplace del coseno debido a la acción de la multiplicación del exponencial.

3.4 EJERCICIO 4

Obtener la transformada z de $e^{-at} \cos \omega t$

MATLAB

```
>> syms t a w
```

```
>> sys = exp(-a*t)*cos(w*t);
```

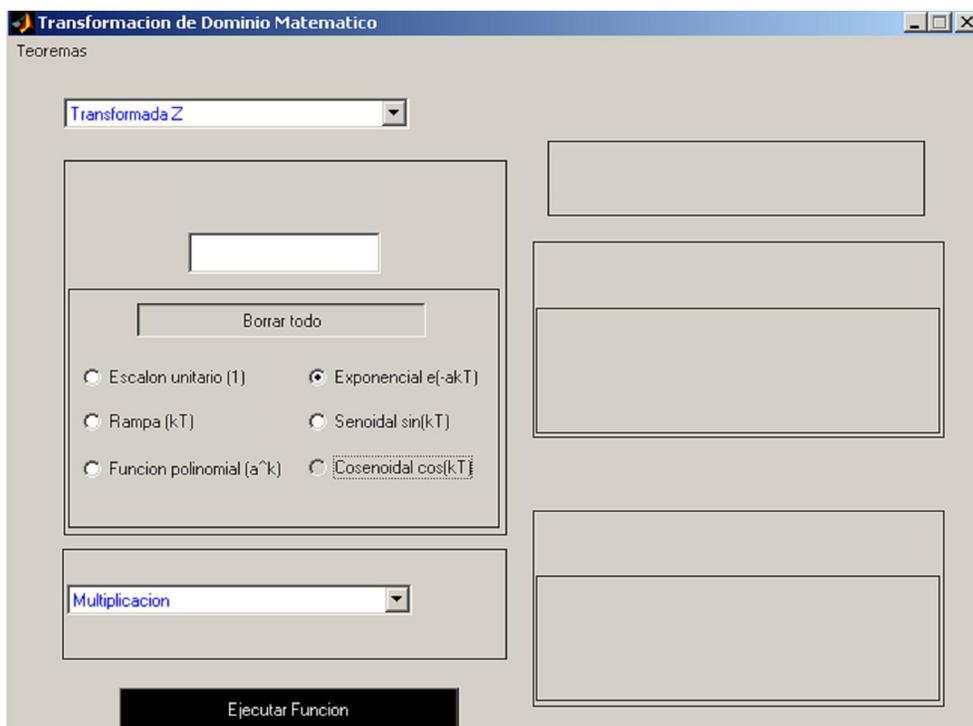
```
>> pretty(ztrans(sys))
```

$$\frac{\exp(-a t) (z - \cos(t)) z}{z^2 - 2 z \cos(t) + 1}$$

TRANSFORMACIÓN DE DOMINIO

De la misma forma y ya que se trata de una función simbólica, hacemos uso del menú predeterminado (ver figura 14).

Figura 14. Ingreso al sistema Ejercicio 4



Obteniendo la siguiente respuesta:

LA SUMATORIA DESDE K=0 HASTA K=10 DE X(K)*Z^-K ES LA SIGUIENTE

$$\begin{aligned}
 & 1 + \frac{\exp(-a T) \cos(T)}{z} + \frac{\exp(-2 a T) \cos(2 T)}{z^2} + \frac{\exp(-3 a T) \cos(3 T)}{z^3} \\
 & + \frac{\exp(-4 a T) \cos(4 T)}{z^4} + \frac{\exp(-5 a T) \cos(5 T)}{z^5} + \frac{\exp(-6 a T) \cos(6 T)}{z^6} \\
 & + \frac{\exp(-7 a T) \cos(7 T)}{z^7} + \frac{\exp(-8 a T) \cos(8 T)}{z^8} + \frac{\exp(-9 a T) \cos(9 T)}{z^9} \\
 & + \frac{\exp(-10 a T) \cos(10 T)}{z^{10}}
 \end{aligned}$$

LA TRANSFORMADA Z DEL SISTEMA INGRESADO ES LA SIGUIENTE:

$$\frac{(z - \cos(T) \exp(-a T)) z}{z^2 - 2 z \cos(T) \exp(-a T) + \exp(-a T)^2}$$

4. CONCLUSIONES

- Diseñar una aplicación de interfaz grafica en este caso de Matlab, implica un amplio conocimiento previo acerca del manejo de los distintos controles con los que puede interactuar el usuario. A pesar de esto y con la ayuda de conocimiento básicos en el tema, se ha logrado la consecución de un software que resuelve sin inconvenientes cualquier función que sea introducida por el usuario en sus distintos módulos.
- Aunque la funcionalidad del software es la adecuada, esta versión está propensa a mejoras tanto en programación como en el desarrollo de su ambiente, ya que se trata de una versión "beta". Por lo anterior, queda abierta la puerta a cualquier actualización que pueda sufrir la aplicación.
- La intencionalidad al desarrollar una herramienta que permita la transformación de dominio matemático, no es la de mejorar y mucho menos reemplazar la labor hecha mediante los comandos de Matlab directamente. De hecho lo que se pretende es dar a conocer al usuario del proceso que se debe seguir para lograr la solución al problema y también proporcionar una visual de los teoremas empleados para lograr cada solución; como es el caso del módulo de la transformada de Laplace.
- Ya que el tema base de este trabajo es de un volumen bastante amplio, se hace complicado realizar una síntesis que pueda ser plasmada en el desarrollo de una aplicación en GUI. Sin embargo, la aplicación da una luz de los aspectos más importantes que se pueden encontrar en el ámbito de la transformada de Laplace y la transformada Z

BIBLIOGRAFÍA

KATSUHIKO Ogata. Ingeniería de control moderna: Prentice Hall, 3 ed

KATSUHIKO Ogata. Problemas de ingeniería de control utilizando Matlab: Prentice Hall, 1999

KATSUHIKO Ogata. Sistemas de control en tiempo discreto: Pearson Educación 2 ed, 1996

RAMÍREZ, Luis Alberto. Introducción a la programación visual con Matlab. Pdf

TROCOSO, Ricardo. Sistemas de tiempo continuo. Especialización en Automatización Industrial. Pdf

ANEXO. Código de programación

```
function General_OpeningFcn(hObject, eventdata, handles, varargin)
set(hObject,'Name','Transformacion de Dominio Matematico');

function varargout = General_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function popupmenu1_Callback(hObject, eventdata, handles) % MENU PRINCIPAL
switch str{val}; % MONITOREA LA SELECCION QUE HA HECHO EL USUARIO
case 'Transformada de Laplace'
handles.current_data = 1;
case 'Transformada Inversa de Laplace'
handles.current_data = 2;
case 'Transformada Z'
handles.current_data = 3;
case 'Transformada Z Inversa'
handles.current_data = 4;
end
guidata(hObject,handles)

function pushbutton1_Callback(hObject, eventdata, handles) % BOTON PRINCIPAL DE EJECUCION DE TAREA

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if handles.current_data == 2 % DEFINE SI HA SIDO LLAMADO EL PROCEDIMIENTO DE
TRANSFORMADA INVERSA DE LAPLACE
set(hObject,'Visible','off');
syms s
handles.Num = [str2num(get(handles.edit12,'String'))]; % CAPTURA LA FUNCION
handles.Den = [str2num(get(handles.edit13,'String'))]; % INGRESADA POR EL USUARIO
handles.H = roots(handles.Den);
handles.Im = imag(handles.H);
LongitudN = length(handles.Num);
LongitudD = length(handles.Den);
if LongitudN== 0 | LongitudD == 0 % ENVIA MENSAJE DE ERROR SI SE EJECUTA UNA ACCION SIN
INGRESAR LA FUNCION
set(hObject,'Visible','on');
Respuesta=errorldg('Sistema Incorrecto',' Transformada Inversa de Laplace ');
switch Respuesta
case 'OK'
end
else
nada = 0;
Immin = min(handles.Im);
if abs(Immin) < 0.01
handles.Im = 0;
end
if handles.Im == 0 % DETERMINA SI EL NUMERADOR DE LA FUNCION POSEE RAICES
REALES
[r,p,k]=residue(handles.Num,handles.Den);
rtrans = r';
ptrans = p';
set(handles.pushbutton3,'Visible','on','String','Introducir datos');
set(handles.text19,'Visible','on','String','Introduzca r')
set(handles.text20,'Visible','on','String','Introduzca p')
else % AL NO SER RAICES REALES, INICIA PROCEDIMIENTO PARA RAICES
COMPLEJAS
set(handles.text24,'Visible','on');
set(handles.text25,'Visible','on');
set(handles.text24,'String','CASO II'); % ESCRIBE EN PANTALLA EL CASO II, RAICES IMAGINARIAS
set(handles.text25,'String','Raices complejas simples');
set(handles.text27,'String','Calculando.....');
```

```

syms s
ITRANS = 0;
Real = 0;
tam = size(handles.H);
Posreal = 0;
for n=1:tam(1)
    temp = handles.Im(n);
    if temp == 0
        Real= Real +1;
        Posreal = n;
    end
end
Imaginario = tam(1) - Real;
for i=1:tam(1) % CICLO DE ARMADO DE LAS FRACCIONES PARCIALES
    nume = polyval(handles.Num,handles.H(i));
    dense = deconv(handles.Den,[1 -handles.H(i)]);
    dene = polyval(dense,handles.H(i));
    Numer(i) = nume/dene;
    Denumer(i)= (s-handles.H(i));
    Sys(i)=Numer(i)/Denumer(i);
    syst(i) = factor(ilaplace(Sys(i)));
end
if Posreal-tam(1) == 0 % CALCULO DE X Y ALFA Y BETA
    for i=1:2:(Posreal-1)
        X(i) = real(Numer(i));
        if abs(X(i))<0.01
            X(i)=0;
        end
        Y(i) = abs(imag(Numer(i)));
        if Y(i)<0.01
            Y(i)=0;
        end
        ALFA(i) = real(-handles.H(i));
        BETA(i) = abs(imag(handles.H(i)));
    end
else
    for i=(Posreal+1):2:tam(1)
        X(i) = real(Numer(i));
        if abs(X(i))<0.01
            X(i)=0;
        end
        Y(i) = abs(imag(Numer(i)));
        if Y(i)<0.01
            Y(i)=0;
        end
        ALFA(i) = real(-handles.H(i));
        if abs(ALFA(i))<0.001
            ALFA(i)=0;
        end
        BETA(i) = abs(imag(handles.H(i)));
        if abs(BETA(i))<0.001
            BETA(i)=0;
        end
    end
end
end

for i =1:2:Imaginario % CALCULO DE FRACCIONES PARCIALES SIN NUMEROS IMAGINARIOS
    Sys(i+Posreal) = 2*X(Posreal+i)*[(s+ALFA(Posreal+i))/((s+ALFA(Posreal+i))^2+BETA(Posreal+i)^2)];
    syst(i+Posreal) = factor(ilaplace(Sys(i+Posreal)));
    Temporal = [1/((s+ALFA(Posreal+i))^2+BETA(Posreal+i)^2)];
    syst(i+Real+1) = 2*Y(Posreal+i)*BETA(Posreal+i)*ilaplace(Temporal);
    Sys(i+Real+1) = 2*Y(Posreal+i)*BETA(Posreal+i)*Temporal;
end
end
for i = 1:tam(1)
    ITRANS = ITRANS + syst(i);
end
end

```

```

Resultado = ITRANS;
FparcialesS = Sys;
FparcialesT = syst;
Sistema = tf(handles.Num,handles.Den);
save('Resultado','Resultado','Sistema') % GENERACION DE ARCHIVOS
save('Fparciales','FparcialesS','FparcialesT')
set(handles.text27,'String','Los archivos generados son los siguientes:');
set(handles.text31,'String','Resultado.mat');
set(handles.text28,'String',[' ','Este archivo almacena dos variables:',' ','-SISTEMA. Representa el sistema
introducido en funcion de transferencia',...
'-RESULTADO. Representa la transformada inversa de laplace del sistema'])
set(handles.text10,'String','Fparciales.mat');
set(handles.text29,'String',[' ','Este archivo almacena dos variables:',' ','-FPARCIALESS. Representa las
fracciones parciales del sistema introducido',...
'-FPARCIALEST. Representa la transformada inversa de laplace de cada fraccion parcial del sistema'])
clc
fprintf('EL SISTEMA INGRESADO FUE EL SIGUIENTE: \n'); % PRESENTACION EN PANTALLA DE RESULTADOS
Sistema
fprintf(' \n');
fprintf('LAS FRACCIONES PARCIALES CALCULADAS PARA EL SISTEMA SON LAS SIGUIENTES: \n');
pretty(FparcialesS)
fprintf(' \n');
fprintf('LA TRANSFORMADA INVERSA DE LAPLACE DE LAS FRACCIONES PARCIALES ES LA SIGUIENTE: \n');
pretty(FparcialesT)
fprintf(' \n');
fprintf('LA TRANSFORMADA INVERSA DE LAPLACE DEL SISTEMA INGRESADO ES LA SIGUIENTE: \n');
pretty(Resultado)
end
end
end
%%%%%%%%%%
if handles.current_data == 3 % Transformada Z
syms k T t a
if handles.Vtoogle == 1
if handles.Func==0
Respuesta=errordlg('Escoja al menos una funcion',' Transformada Z ');
switch Respuesta
case 'OK'
end
else
set(handles.text27,'String','Calculando....');
Suma = 0;
syms k T z

for k=0:10
if handles.Func == 1
Suma = Suma + z^-k;
else
Suma = Suma + eval(handles.Func)*z^-k;
end
end
if handles.Func == 1
TransZ = z/(z-1);
else
TransZ = factor(ztrans(handles.Func));
end
end
clc
fprintf('EL SISTEMA INGRESADO FUE EL SIGUIENTE: \n');
Sistema
fprintf(' \n');
fprintf('LA SUMATORIA DESDE K=0 HASTA K=10 DE X(K)*Z^-K ES LA SIGUIENTE \n');
pretty(Suma)
fprintf(' \n');
fprintf('LA TRANSFORMADA Z DEL SISTEMA INGRESADO ES LA SIGUIENTE: \n');
pretty(TransZ)
end
end
else

```

```

syms k T z
Suma = 0;

dato = char(get(handles.edit12,'String'));
Ldato = length(dato);
if Ldato == 0
    Respuesta=errordlg('No ha ingresado la funcion',' Transformada Z ');
    switch Respuesta
        case 'OK'
            end
    else
        set(handles.text27,'String','Calculando....');
        dato = sym(dato);
        for k=0:10
            Suma = Suma + eval(dato)*z^k;
        end
        Sistema = dato;
        TransZ = factor(ztrans(dato));
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if handles.current_data == 4 % DETERMINA SI SE HA LLAMADO EL PROCEDIMIENTO DE
TRANSFORMADA INVERSA Z
    VRampa = get(handles.Rampa,'Value');
    VSenoidal = get(handles.Senoidal,'Value');
    if VRampa==0 & VSenoidal==0
        Respuesta=warndlg('Seleccione un procedimiento','Transformada Inversa Z');
        switch Respuesta
            case 'OK'
                end
        else
            if handles.current_data == 4 & handles.Procedimiento==1 % VERIFICA SI SE HA ESCOGIDO EL PROCEDIMIENTO
DE FRACCIONES PARCIALES
                handles.Num = [str2num(get(handles.edit12,'String'))];
                handles.Den = [str2num(get(handles.edit13,'String'))];
                handles.H = roots(handles.Den);
                Npr = handles.Num;
                Npr(1,1) = 1;
                Respuesta = questdlg('¿ Sistema correcto?', ' Transformada Z Inversa ');
                switch Respuesta
                    case 'No'
                        % NO HAGA NADA
                    case 'Cancel'
                    case 'Yes'
                        set(handles.text27,'String','Calculando....');
                        syms z k Ddirac
                        ITRANS = 0;
                        Real = 0;
                        tam = length(handles.H);
                        Posreal = 0;
                        Im = imag(handles.H);
                        for n=1:tam
                            temp =Im(n);
                            if temp == 0
                                Real= Real +1;
                                Posreal = n;
                            end
                        end
                        Imaginario = tam - Real;
                        for i=1:tam
                            nume = polyval(handles.Num,handles.H(i));
                            dense = deconv(handles.Den,[1 -handles.H(i)]);
                            dene = polyval(dense,handles.H(i));
                            Numer(i) = nume/dene;
                            Denumer(i)= (z-handles.H(i));
                            Sys(i)=Numer(i)/Denumer(i);

```

```

        if Sys(i) == (1/z) | Sys(i) == (-1/z)
            PruebaS = z*Sys(i);
            Syst(i) = PruebaS*Ddirac;
        else
            Syst(i) = iztrans(z*Sys(i),k);
        end
    end
end
if Posreal-tam(1) == 0
    for i=1:2:(Posreal-1)
        X(i) = real(Numera(i));
        if abs(X(i))<0.01
            X(i)=0;
        end
        Y(i) = abs(imag(Numera(i)));
        if Y(i)<0.01
            Y(i)=0;
        end
        ALFA(i) = real(-handles.H(i));
        BETA(i) = abs(imag(handles.H(i)));
    end
end
end
for i =1:2:Imaginario
    if Posreal-tam(1) == 0
        Sys(i) = 2*X(i)*[(z+ALFA(i))/(z^2+2*ALFA(i)*z+ALFA(i)^2+BETA(i)^2)];
        MaT(i) = 0.5*(log(ALFA(i)^2+BETA(i)^2));
        EMat(i) = exp(MaT(i));
        WT(i) = acos(-(ALFA(i)/EMat(i)));
        Syst(i) = 2*X(i)*(EMat(i))^k*cos(k*WT(i));
        Sys(i+1) = 2*Y(i)*BETA(i)*[1/(z^2+2*ALFA(i)*z+ALFA(i)^2+BETA(i)^2)];
        Syst(i+1) = [(2*Y(i)*BETA(i))/(sin(WT(i)*EMat(i)))]*(EMat(i))^k*sin(k*WT(i));
    end
end
SYST = 0;
for i = 1:tam(1)
    SYST = SYST + Syst(i);
end
FparcialesZ = Sys;
FparcialesK = Syst;
Sistema = tf(handles.Num,handles.Den,'Variable','z');
Resultado = SYST;
save('Proceso','FparcialesZ','FparcialesK')
save('Resultado','Resultado','Sistema')
set(handles.text27,'String','Los archivos generados son los siguientes:');
set(handles.text31,'String','Resultado.mat');
set(handles.text28,'String',[' ','Este archivo almacena dos variables:',' ','...
'-SISTEMA. Representa el sistema introducido en funcion de transferencia',...
' -RESULTADO. Representa la transformada Z inversa del sistema'])
set(handles.text10,'String','Proceso.mat');
set(handles.text29,'String',[' ','Este archivo almacena dos variables:',' ','...
'-FPARCIALESZ. Representa las fracciones parciales del sistema introducido',...
' -FPARCIALESK. Representa la transformada Z inversa de cada fraccion parcial del sistema'])

clc
fprintf('EL SISTEMA INGRESADO FUE EL SIGUIENTE: \n');
Sistema
fprintf(' \n');
fprintf('LAS FRACCIONES PARCIALES CALCULADAS PARA EL SISTEMA SON LAS SIGUIENTES: \n');
pretty(FparcialesZ)
fprintf(' \n');
fprintf('LA TRANSFORMADA Z INVERSA DE LAS FRACCIONES PARCIALES ES LA SIGUIENTE: \n');
pretty(FparcialesK)
fprintf(' \n');
fprintf('LA TRANSFORMADA Z INVERSA DEL SISTEMA INGRESADO ES LA SIGUIENTE: \n');
pretty(Resultado)
end
end

```

```

if handles.current_data ==4 & handles.Procedimiento ==2
    Num = [str2num(get(handles.edit12,'String'))];
    Den = [str2num(get(handles.edit13,'String'))];

    Respuesta = questdlg('¿Sistema correcto?', ' Transformada Z Inversa ');
    switch Respuesta
        case 'No'
            % NO HAGAN NADA
        case 'Cancel'
        case 'Yes'
            x = [1 zeros(1,40)];
            y = filter(Num,Den,x);
            k=0:40;
            RKronecker = y;
            Sistema = tf(Num,Den,'Variable','z');
            save('Resultado','RKronecker','Sistema')
            set(handles.text27,'String','El archivo generado es el siguiente:');
            set(handles.text31,'String','Resultado.mat');
            set(handles.text28,'String',[' ','Este archivo almacena dos variables:',' ','...
            '-RKRONECKER. Representa la respuesta del sistema ante una entrada delta de Kronecker',...
            '-SISTEMA. Representa el sistema introducido en fucion de transferencia'])
        clc
        fprintf('EL SISTEMA INGRESADO FUE EL SIGUIENTE: \n');
        Sistema
        fprintf(' \n');
        fprintf('LAS RESPUESTA DEL SISTEMA ANTE UNA ENTRADA DELTA \n');
        fprintf('DE KRONECKER ES LA SIGUIENTE: \n');
        RKronecker
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if handles.current_data == 1 % SE DETERMINA SI SE HA LLAMADO EL PROCEDIMIENTO DE
TRANSFORMA DE LAPLACE
    syms a k T t
    TransL = laplace(handles.Final);
    Sistema = handles.Final;
    save('Resultado','Sistema','TransL');
    set(handles.text27,'String','El archivo generado es el siguiente:');
    set(handles.text31,'String','Resultado.mat');
    set(handles.text28,'String',[' ','Este archivo almacena dos variables:',' ','-SISTEMA. Representa el sistema
introducido',...
    '-TRANSL. Representa la transformada de Laplace del sistema introducido'])
    clc
    fprintf('EL SISTEMA INTRODUCIO FUE EL SIGUIENTE: \n');
    Sistema
    fprintf(' \n');
    fprintf('LA TRANSFORMADA DE LAPLACE DEL SISTEMA INTRODUCIDO ES EL SIGUIENTE: \n');
    pretty(TransL)
end
guidata(hObject,handles)

function pushbutton3_Callback(hObject, eventdata, handles) % SEGUNDO BOTON DE EJECUCION DE
PROCEDIMIENTO(AUXILIAR)
if handles.current_data==2
    ITRANS = 0;
    r = [str2num(get(handles.edit14,'String'))]; % CAPTURA LOS VALORES DE R Y P INGRESADOS POR EL
USUARIO
    r = r';
    Lr = length(r);
    p = [str2num(get(handles.edit15,'String'))];
    p = p';
    Lp = length(p);
    tam = size(r);
    syms s

```

```

if Lr == 0 | Lp == 0
    Respuesta = warndlg('No ha introducido los datos','Transformada Inversa de Laplace');
    switch Respuesta
        case 'OK'
            end
        else
            n=1;
            u=0;
            for i=1:tam(1,1)-1                % IDENTIFICA EXISTENCIA DE RAICES MULTIPLES
                if p(i+1)==p(i)
                    n=n+1;
                end
            end
            if n>1
                set(handles.text24,'Visible','on');
                set(handles.text25,'Visible','on');
                set(handles.text24,'String','CASO III');
                set(handles.text25,'String','Raices reales multiples');
                set(handles.text27,'String','Calculando.....');
                for i=1:n
                    u = u+1;
                    Numfp(1,i) = r(i);
                    Denfp(1,i) = (s-p(i))^u;
                    sys(1,i) = Numfp(1,i)/Denfp(1,i);
                    syst(1,i) = ilaplace(sys(1,i));
                    ITRANS = ITRANS + syst(1,i);
                end
                FparcialesS = sys;
                FparcialesT = syst;
                Resultado = ITRANS;
                Sistema = tf(handles.Num,handles.Den);
                save('Fparciales','FparcialesS','FparcialesT')
                save('Resultado','Resultado','Sistema')
                set(handles.text27,'String','Los archivos generados son los siguientes:');
                set(handles.text31,'String','Resultado.mat');
                set(handles.text28,'String',[' ','Este archivo almacena dos variables:',' ','...
                    '-SISTEMA. Representa el sistema introducido en funcion de transferencia','...
                    '-RESULTADO. Representa la transformada inversa de laplace del sistema'])
                set(handles.text10,'String','Fparciales.mat');
                set(handles.text29,'String',[' ','Este archivo almacena dos variables:',' ','...
                    '-FPARCIALESS. Representa las fracciones parciales del sistema introducido','...
                    '-FPARCIALEST. Representa la transformada inversa de laplace de cada fraccion parcial del sistema'])
            else
                set(handles.text24,'Visible','on');
                set(handles.text25,'Visible','on');
                set(handles.text24,'String','CASO I');
                set(handles.text25,'String','Raices reales simples');
                set(handles.text27,'String','Calculando.....');
            end
            clc
            fprintf('EL SISTEMA INGRESADO FUE EL SIGUIENTE: \n');
            Sistema
            fprintf(' \n');
            fprintf('LAS FRACCIONES PARCIALES CALCULADAS PARA EL SISTEMA SON LAS SIGUIENTES: \n');
            pretty(FparcialesS)
            fprintf(' \n');
            fprintf('LA TRANSFORMADA INVERSA DE LAPLACE DE LAS FRACCIONES PARCIALES ES LA SIGUIENTE: \n');
            pretty(FparcialesT)
            fprintf(' \n');
            fprintf('LA TRANSFORMADA INVERSA DE LAPLACE DEL SISTEMA INGRESADO ES LA SIGUIENTE: \n');
            pretty(Resultado)
        end
    end
end

function togglebutton1_Callback(hObject, eventdata, handles) % BOTON DE SELECCION DE FUNCIONES
PREDETERMINADAS

```

```
handles.Vtoogle = get(hObject,'Value');
if handles.Vtoogle == 0
    handles.Func = 0;
    if handles.current_data == 3
        set(handles.text17,'String','Introduzca la funcion en terminos de "k*T"');
    end
else
    off=[handles.Escalon, handles.Rampa, handles.Cosenoidal, handles.Senoidal,handles.Exponencial,
handles.Polinomial];
    set(off,'Value',0);
    if handles.current_data == 1
        set(handles.pushbutton1,'Visible','off');
        set(hObject,'String','Borrar todo','Visible','on');
        set(handles.Exponencial,'String','Exponencial e(-a*t)','Visible','on');
    else
        set(handles.text17,'String','');
        set(hObject,'String','Borrar todo');
        set(handles.Exponencial,'String','Exponencial e(-akT)','Visible','on');
    end
end
guidata(hObject,handles)

function Escalon_Callback(hObject, eventdata, handles)    % TAREA A REALIZAR ANTE LLAMADA A FUNCION
ESCALON
set(handles.popupmenu2,'Value',1,'Visible','on');
Value = get(hObject,'Value');
handles.Opera = get(handles.popupmenu2,'Value');
syms a k T t
if Value == 1
    handles.Cont = handles.Cont + 1;
    if handles.Cont == 2
        set(handles.popupmenu2,'Value',1,'Visible','off');
    end
    if handles.current_data == 1
        set(handles.pushbutton1,'Visible','on');
        set(handles.popupmenu2,'Visible','on');

        if handles.Opera == 1 | handles.Opera == 2
            handles.Final = handles.Final + 1;
        end
        if handles.Opera == 3
            handles.Final = handles.Final - 1;
        end
        if handles.Opera == 4
            handles.Final = handles.Final * 1;
        end
    end
end
guidata(hObject,handles)

function Rampa_Callback(hObject, eventdata, handles)

if handles.current_data == 4
    set(handles.Senoidal,'Value',0);
    handles.Procedimiento=1;
else
    Value = get(hObject,'Value');
    handles.Opera = get(handles.popupmenu2,'Value');
    syms a k T t
    if Value == 1
        handles.Cont = handles.Cont + 1;
        if handles.Cont ==2
            set(handles.popupmenu2,'Value',1,'Visible','off');
        end
        if handles.Opera == 1 | handles.Opera == 2
            handles.Final = handles.Final + t;
        end
    end
end
```

```
        if handles.Opera == 3
            handles.Final = handles.Final - t;
        end
        if handles.Opera == 4
            handles.Final = handles.Final * t;
        end
    end
end

guidata(hObject,handles)

function Polinomial_Callback(hObject, eventdata, handles)
Value = get(hObject,'Value');
handles.Opera = get(handles.popupmenu2,'Value');
syms a k T t
if Value == 1
    handles.Cont = handles.Cont + 1;
    if handles.current_data == 1
        set(handles.pushbutton1,'Visible','on');
        set(handles.popupmenu2,'Visible','on');
        if handles.Opera == 1 | handles.Opera == 2
            handles.Final = handles.Final + exp(a*t);
        end
        if handles.Opera == 3
            handles.Final = handles.Final - exp(a*t);
        end
        if handles.Opera == 4
            handles.Final = handles.Final * exp(a*t);
        end
    end
    guidata(hObject,handles)
function Exponencial_Callback(hObject, eventdata, handles)
Value = get(hObject,'Value');
handles.Opera = get(handles.popupmenu2,'Value');
syms a k T t
if Value == 1
    handles.Cont = handles.Cont + 1;
    if handles.Cont == 2
        set(handles.popupmenu2,'Value',1,'Visible','off');
    end
    if handles.current_data == 1
        set(handles.pushbutton1,'Visible','on');
        set(handles.popupmenu2,'Visible','on');
        if handles.Opera == 1 | handles.Opera == 2
            handles.Final = handles.Final + exp(-a*t);
        end
        if handles.Opera == 3
            handles.Final = handles.Final - exp(-a*t);
        end
        if handles.Opera == 4
            handles.Final = handles.Final * exp(-a*t);
        end
    end
    guidata(hObject,handles)

function Senoidal_Callback(hObject, eventdata, handles)
if handles.current_data ==4
    set(handles.Rampa,'Value',0);
    handles.Procedimiento=2;
else
    Value = get(hObject,'Value');
    handles.Opera = get(handles.popupmenu2,'Value');
    syms a k T t
    if Value == 1
        handles.Cont = handles.Cont + 1;
        if handles.Cont == 2
            set(handles.popupmenu2,'Value',1,'Visible','off');
```

```
end
if handles.Opera == 1 | handles.Opera == 2
    handles.Final = handles.Final + sin(a*t);
end
if handles.Opera == 3
    handles.Final = handles.Final - sin(a*t);
end
if handles.Opera == 4
    handles.Final = handles.Final * sin(a*t);
end
end
guidata(hObject,handles)

function Cosenoidal_Callback(hObject, eventdata, handles)
Value = get(hObject,'Value');
handles.Opera = get(handles.popupmenu2,'Value');
if Value == 1
    if handles.current_data == 1
        if handles.Opera == 1 | handles.Opera == 2
            handles.Final = handles.Final + cos(a*t);
        end
        if handles.Opera == 3
            handles.Final = handles.Final - cos(a*t);
        end
        if handles.Opera == 4
            handles.Final = handles.Final * cos(a*t);
        end
    end
end
guidata(hObject,handles)
```