

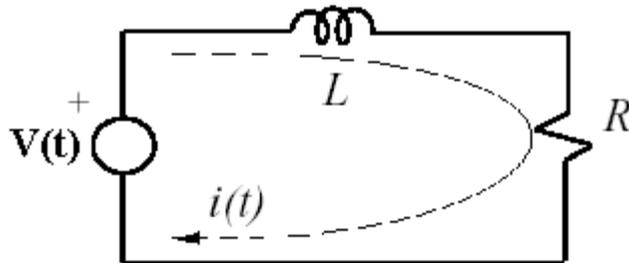
## *Desarrollo Del Algoritmo De La Ecuación Diferencial*

### 4.1. INTRODUCCION.

Al representar la línea mediante un circuito R-L El algoritmo de protección a distancia determina la localización de una falla en función de los valores de la resistencia y reactancia de la línea de transmisión que existe en el momento de su ocurrencia resolviendo una ecuación diferencial.

### 4.2. SOLUCIÓN DEL MODELO MONOFÁSICO DE LA ECUACIÓN DIFERENCIAL.

El algoritmo de la ecuación diferencial consiste en modelar un sistema de potencia en lugar de considerar el modelo de la señal, para el desarrollo del algoritmo se considera el equivalente monofásico de una línea en falla mostrado en la figura 4.1



**Figura 4.1.** Modelo de un circuito serie R-L de una línea de transmisión

$$R I_1(t) + L \frac{d i_1}{dt} = V_1(t) \quad (4.1)$$

Si se considera que la falla ocurre a una distancia determinada, es posible expresar la ecuación diferencial de la siguiente forma.

$$\kappa_R R \mathbf{I}(t) + \kappa_L L \frac{d\lambda}{dt} = \mathbf{V}(t) \quad (4.2)$$

Donde  $\kappa_R$  y  $\kappa_L$  son los valores que representan los estimados en por unidad de la línea de transmisión, las magnitudes de voltaje  $\mathbf{V}(t)$  y corriente  $\mathbf{I}(t)$  son determinadas por las magnitudes medidas por los transformadores de corriente y potencial y los valores  $R$  y  $L$  son conocidos y dependientes de la línea por lo tanto es posible estimar la distancia a la falla resolviendo una ecuación diferencial en términos de los estimados a la longitud del punto fallado.

Al representar la línea mediante en circuito R-L supone que el modelo es exacto para líneas cortas aunque evidentemente el modelo implica otras aproximaciones que son: los TCs y TPs son ideales en el rango de frecuencia de 50 a 300 Hz, la resistencia de falla es pequeña y la línea se considera perfectamente transpuesta.

Para el análisis del problema se consideran un instante  $n$  y  $n+1$  para el caso del instante  $n$  corresponde 1 y para  $n+1$  corresponde 2, esto aplica para la subíndices de los elementos dependientes del tiempo, esto con el fin de facilitar el calculo. Y ahora tenemos la misma ecuación pero para dos instantes diferentes.

$$\kappa_R R \mathbf{I}_1(t) + \kappa_L L \frac{d\lambda_1}{dt} = \mathbf{V}_1(t) \quad (4.3)$$

$$\kappa_R R \mathbf{I}_2(t) + \kappa_L L \frac{d\lambda_2}{dt} = \mathbf{V}_2(t) \quad (4.4)$$

Despejando  $R$  de (4.2) tenemos:

$$\frac{R = \mathbf{V}_2(t) - \kappa_L L \frac{d\lambda_2}{dt}}{\kappa_R \mathbf{I}_2(t)} \quad (4.5)$$

Sustituyendo (4.5) en (4.3) tenemos:

$$\frac{\mathbf{I}_1 \mathbf{V}_2(t) - \mathbf{I}_1 \kappa_L L \frac{d\lambda_2}{dt} + \kappa_L L \frac{d\lambda_1}{dt}}{\mathbf{I}_2(t)} = \mathbf{V}_1(t) \quad (4.6)$$

$$\frac{\mathbf{I}_1 \mathbf{V}_2(t) - \mathbf{I}_1 \kappa_{\ell} \mathbf{L} \frac{d\lambda_2}{dt}}{\mathbf{I}_2(t)} = \mathbf{V}_1(t) - \kappa_{\ell} \mathbf{L} \frac{d\lambda_1}{dt} \quad (4.7)$$

Ahora se despeja  $\kappa_{\ell} \mathbf{L}$ :

$$\mathbf{I}_1 \mathbf{V}_2(t) - \mathbf{I}_1 \kappa_{\ell} \mathbf{L} \frac{d\lambda_2}{dt} = \mathbf{I}_2(t) \mathbf{V}_1(t) - \mathbf{I}_2(t) \kappa_{\ell} \mathbf{L} \frac{d\lambda_1}{dt} \quad (4.8)$$

$$\kappa_{\ell} \mathbf{L} = \frac{\mathbf{I}_1 \mathbf{V}_2(t) - \mathbf{I}_1 \kappa_{\ell} \mathbf{L} \frac{d\lambda_2}{dt}}{\mathbf{I}_2(t) \mathbf{V}_1(t) - \mathbf{I}_2(t) \kappa_{\ell} \mathbf{L} \frac{d\lambda_1}{dt}} \quad (4.9)$$

Analizando cada elemento de la ecuación, y por medio de la integración trapezoidal para obtener la solución de la ecuación diferencial. [7]

$$\kappa_{\mathbb{R}} \int_t^{t_1} \mathbf{I}(t) + \kappa_{\ell} \mathbf{L} \int_t^{t_1} \frac{d\lambda}{dt} = \int_t^{t_1} \mathbf{V}(t) \quad (4.10)$$

$$\kappa_{\mathbb{R}} \int_{t_1}^{t_2} \mathbf{I}(t) + \kappa_{\ell} \mathbf{L} \int_{t_1}^{t_2} \frac{d\lambda}{dt} = \int_{t_1}^{t_2} \mathbf{V}(t) \quad (4.11)$$

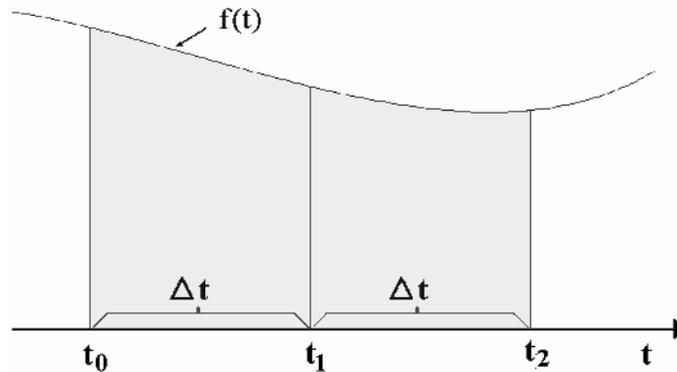


Figura 4.2. Análisis de una función para la solución de la ecuación diferencial

$$\kappa_{\mathbb{R}} \int_t^{t_1} \mathbf{I}(t) dt + \kappa_{\ell} \mathbf{L} \{ \lambda(t_1) - \lambda(t) \} = \int_t^{t_1} \mathbf{V}(t) dt \quad (4.12)$$

$$\int_t^{t_1} \mathbf{V}(t) dt = \left( \frac{t_1 - t}{2} \right) \{ v(t) + v(t_1) \} \implies \int_t^{t_1} \mathbf{V}(t) dt = \left( \frac{t_1 - t}{2} \right) (v + v_1) \quad (4.13)$$

$$\int_{t_1}^{t_2} \mathbf{V}(t)dt = (t/2)(v(t_1)+v(t_2)) \Rightarrow \int_{t_1}^{t_2} \mathbf{V}_2 = (t/2)(v_1+v_2) \quad (4.14)$$

$$\int_t^{t_1} d\lambda = \lambda(t_1) - \lambda(t) \Rightarrow \int_t^{t_1} d\lambda_1/dt = \lambda_1 - \lambda \quad (4.15)$$

$$d\lambda = \lambda(t_2) - \lambda(t_1) \Rightarrow \int_{t_1}^{t_2} d\lambda_2/dt = \lambda_2 - \lambda_1 \quad (4.16)$$

$$\int_t^{t_1} \mathbf{I}(t)dt = (t/2)(\lambda(t) + \lambda(t_1)) \Rightarrow \int_t^{t_1} \mathbf{I}_1 = (t/2)(\lambda_1 + \lambda) \quad (4.17)$$

$$\int_t^{t_1} \mathbf{I}(t)dt = (t/2)(\lambda(t_2) + \lambda(t_1)) \Rightarrow \int_t^{t_1} \mathbf{I}_2 = (t/2)(\lambda_2 + \lambda_1) \quad (4.18)$$

$$\mathcal{L}_I L = \frac{(t/2)(\lambda_1 + \lambda)(t/2)(v_2 + v_1) - (t/2)(\lambda_2 + \lambda_1)(t/2)(v_1 + v)}{(t/2)(\lambda_1 + \lambda)(\lambda_2 - \lambda_1) - (t/2)(\lambda_2 + \lambda_1)(\lambda_1 - \lambda)} \quad (4.19)$$

$$\mathcal{L}_I L = \frac{t}{2} \frac{(\lambda_1 + \lambda)(v_2 + v_1) - (\lambda_2 + \lambda_1)(v_1 + v)}{(\lambda_1 + \lambda)(\lambda_2 - \lambda_1) - (\lambda_2 + \lambda_1)(\lambda_1 - \lambda)} \quad (4.20)$$

$\mathcal{L}_I L = \mathcal{L}_L$  como  $\mathcal{L}_R R = \mathcal{L}_R$  en por unidad de la longitud de la línea de transmisión con una base de 100% de la zona a proteger. A continuación se escribe la expresión en términos de una señal de tiempo discreto, en tiempos de muestra  $k+2, k+1, k$ . [13]

$$\mathcal{L}_L = \frac{t}{2} \frac{(\lambda_{k+1} + \lambda_k)(v_{k+2} + v_{k+1}) - (\lambda_{k+2} + \lambda_{k+1})(v_{k+1} + v_k)}{(\lambda_{k+1} + \lambda_k)(\lambda_{k+2} - \lambda_{k+1}) - (\lambda_{k+2} + \lambda_{k+1})(\lambda_{k+1} - \lambda_k)} \quad (4.21)$$

$$\mathcal{L}_R R \mathbf{I}_1(t) + \mathcal{L}_I L d\lambda_1/dt = \mathbf{V}_1(t) \quad (4.22)$$

$$L = \mathbf{V}_1(t) - \mathcal{L}_R R \mathbf{I}_1(t) \quad (4.23)$$

$$\kappa_{\ell} \mathbf{d}\lambda_1/\mathbf{d}\mathbf{t}$$

$$\kappa_{\mathbb{R}} \mathbf{I}_2(\mathbf{t}) + \kappa_{\ell}(\mathbf{V}_1(\mathbf{t}) - \kappa_{\mathbb{R}} \mathbf{I}_1(\mathbf{t})) \frac{\mathbf{d}\lambda_2/\mathbf{d}\mathbf{t}}{\kappa_{\ell} \mathbf{d}\lambda_1/\mathbf{d}\mathbf{t}} = \mathbf{V}_2(\mathbf{t}) \quad (4.24)$$

$$\frac{\kappa_{\mathbb{R}} - \mathbf{V}_1(\mathbf{t}) \mathbf{d}\lambda_2/\mathbf{d}\mathbf{t} - \mathbf{V}_2(\mathbf{t}) \mathbf{d}\lambda_1/\mathbf{d}\mathbf{t}}{\mathbf{I}_1(\mathbf{t}) \mathbf{d}\lambda_2/\mathbf{d}\mathbf{t} - \mathbf{I}_2(\mathbf{t}) \kappa_{\ell} \mathbf{d}\lambda_1/\mathbf{d}\mathbf{t}} \quad (4.25)$$

$$\kappa_{\mathbb{R}} = \frac{(\nu_{k+1} + \nu_k) (\lambda_{k+2} - \lambda_{k+1}) - (\nu_{k+2} - \nu_{k+1}) (\lambda_{k+1} - \lambda_k)}{(\lambda_{k+1} + \lambda_k) (\lambda_{k+2} - \lambda_{k+1}) - (\lambda_{k+2} + \lambda_{k+1}) (\lambda_{k+1} - \lambda_k)} \quad (4.26)$$

### 4.3. SOFTWARE MATLAB

MATLAB es un ambiente de cómputo, de alta ejecución numérica y de visualización. MATLAB integra el análisis numérico, calculo de matrices, procesamiento de señales, y graficación, en un ambiente sencillo de utilizar, donde los problemas y sus soluciones son expresados justamente como están escritas; a diferencia de la programación tradicional

El nombre de MATLAB surge de *matrix laboratory*, dado que el elemento básico de dato es una matriz, la cual no requiere de dimensionamiento, lo que permite resolver problemas en una fracción de tiempo, del que nos tomaría al escribir un programa en cualquier lenguaje como, Pascal, Fortran, C, etc. [9]

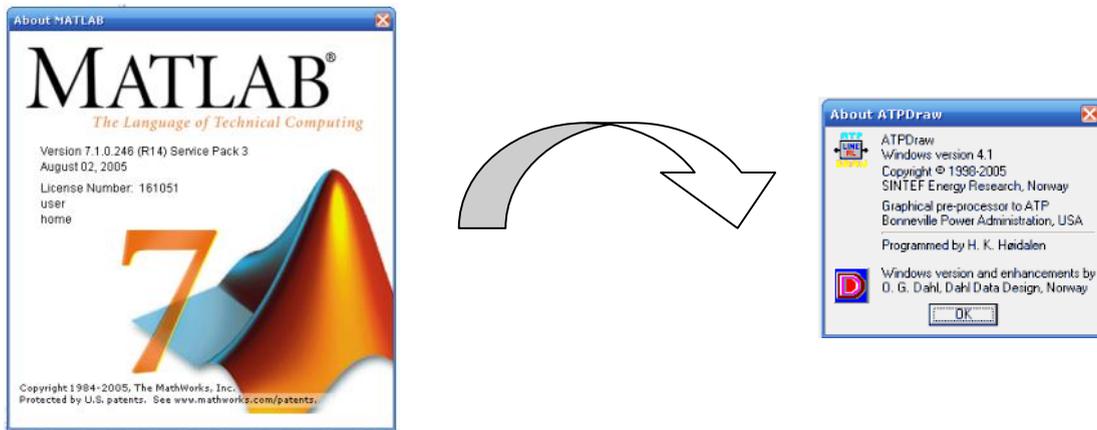
MATLAB, también cuenta con varias familias de soluciones para aplicaciones específicas llamadas *toolboxes*, que son colecciones de funciones utilizadas para resolver alguna clase particular de problema. Las áreas en donde los *toolboxes* están disponibles incluyen el procesamiento de señales, diseño de sistemas de control, la simulación de sistemas dinámicos, la identificación de sistemas, redes neuronales y otros.

#### 4.3.1. Conversión .pl4 a .mat para análisis numérico del algoritmo en Matlab.

El algoritmo de la ecuación diferencial en Matlab es básicamente un análisis numérico de suma, resta, multiplicación y división de matices obtenidas de la solución del

modelo de la ecuación diferencial, estos parámetros son obtenidos de la simulación del modelo de la línea de transmisión en el programa ATP Draw.

En programa ATP Draw las graficas obtenidas de la simulación, tienen una extensión que funciona únicamente con el ejecutable *PLOTXY* de ATP Draw que viene incluido en la instalación de ATP Draw. Las graficas obtenidas en la simulación tiene la extensión *.pl4* mientras que las graficas que genera Matlab tienen la extensión *.mat*.



En las siguientes líneas de comandos en la ventana de Matlab se muestra como importar los archivos desde la dirección de ATP Draw hasta la carpeta de Matlab ya con la extensión *.mat*.

```

Command Window
Warning: MATLAB Toolbox Path Cache is out of date and is not being used.
Type 'help toolbox_path_cache' for more info.

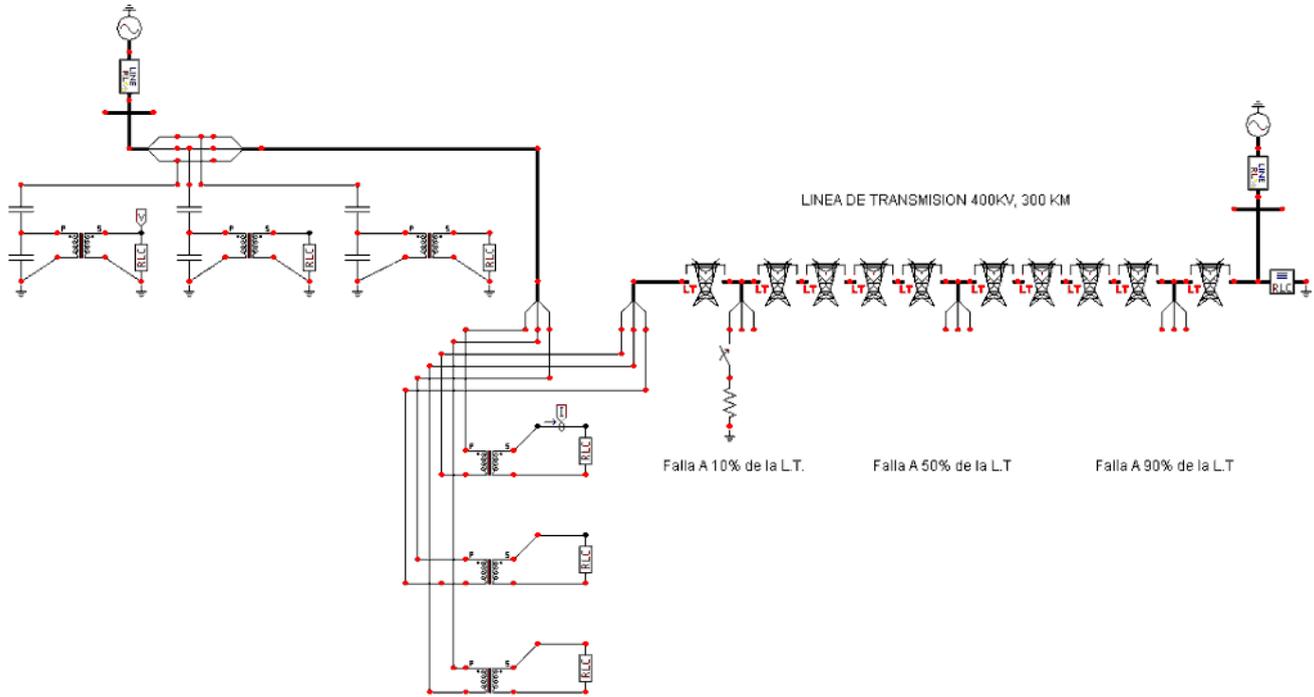
To get started, select MATLAB Help or Demos from the Help menu.

>> !copy c:\atpgnu\T100.pl4 "C:\Archivos de programa\MATLAB71\work"
!P142mat4.exe T100.pl4 T100.mat;
    1 archivos copiados.
>> !copy c:\atpgnu\T10.pl4 "C:\Archivos de programa\MATLAB71\work"
!P142mat4.exe T10.pl4 T10.mat;
    1 archivos copiados.
>>

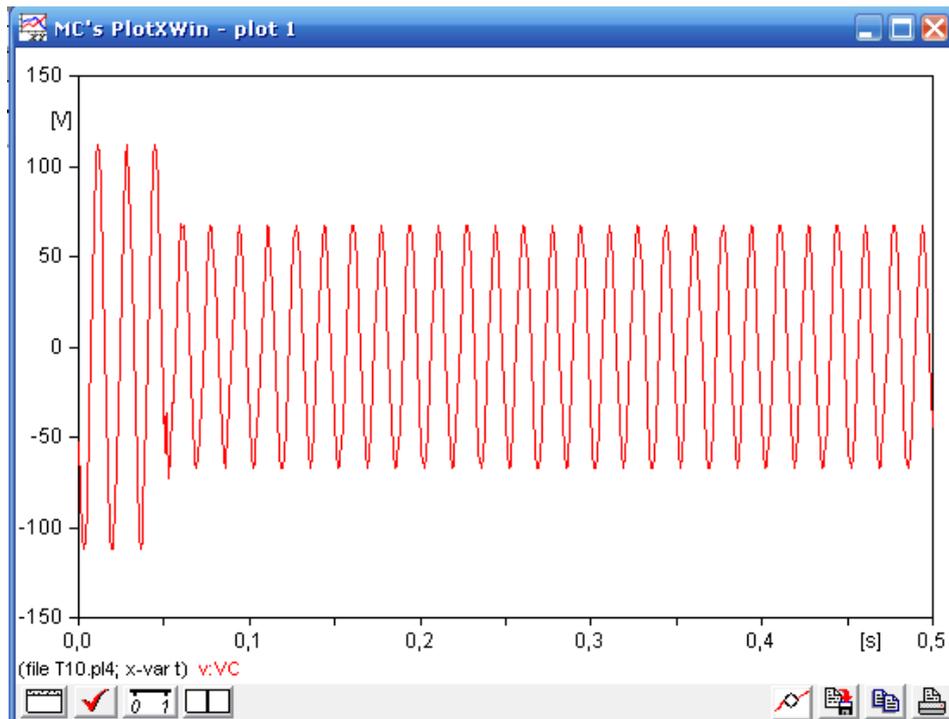
```

**Figura 4.3.** Código en Matlab para importar graficas de *.pl4* a *.mat*,

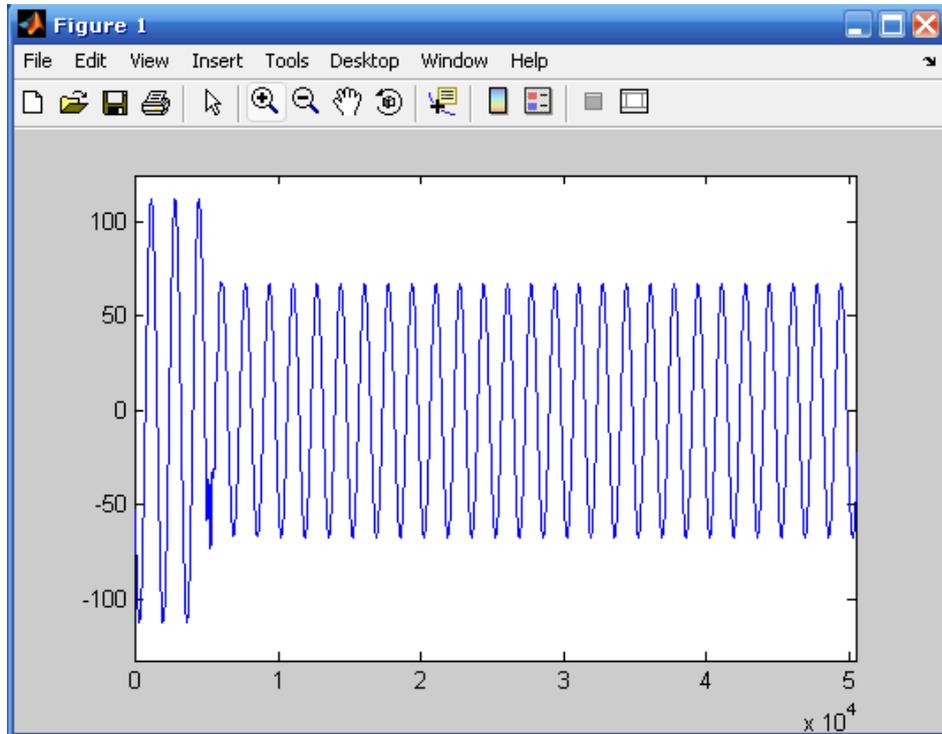
La simulación de un corto circuito es realizada del siguiente modelo que incluye un punto de medición de corriente y voltaje además una línea de 300 Km. En secciones de 30 Km. para simulaciones posteriores.



**Figura 4.4.** Modelo de la línea de transmisión con unidades medición de corriente y voltaje para análisis con carga.



**Figura 4.5.** Voltaje obtenido de la línea de transmisión en la simulación en ATP Draw.



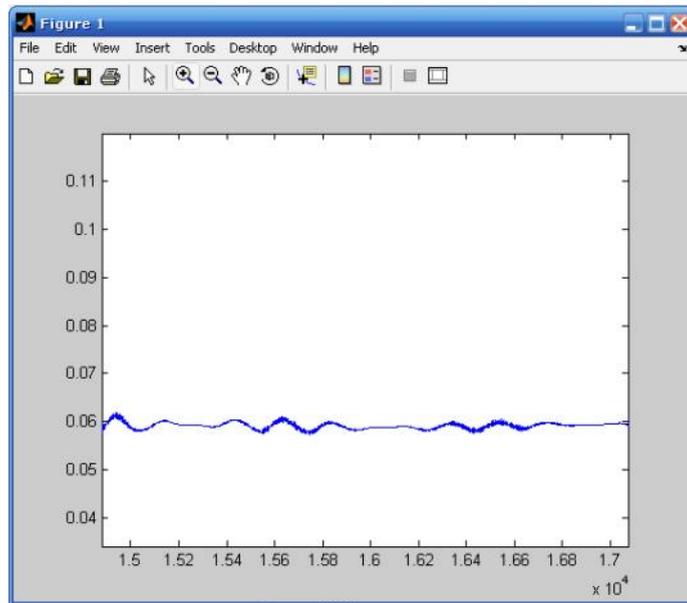
**Figura 4.6.** Voltaje obtenido de la línea de transmisión en el programa Matlab.

#### 4.4. DESARROLLO DEL ALGORITMO DE LA ECUACIÓN DIFERENCIAL EN MATLAB.

Para la solución y obtención de los parámetros base a utilizar para tener el valor en por unidad del estimado al punto de falla utilizamos la solución de la ecuación para la solución de  $k_l$  y obtener el valor de referencia para la longitud total de la línea de transmisión.

```
%-----
% Desarrollo Del Algoritmo De La Ecuación Diferencial
%-----
%Para calcular el valor base para  $k_l$  tenemos
>> !copy c:\atpgnu\T100.pl4 "C:\Archivos de programa\MATLAB71\work"
!PI42mat4.exe T100.pl4 T100.mat;
```

```
!copy c:\atpgnu\T10.pl4 "C:\Archivos de programa\MATLAB71\work"
!PI42mat4.exe T10.pl4 T10.mat;
clear all
load 'T100.mat';
load 'T10.mat';
load 'filtrobutterworth.mat';
%Calculo De Parámetros Base Para El Coeficiente kl De La Ec. Diferencial
k=1;
n=80001;
IIk=iIcCc(k:n-2);
IIk2=iIcCc(k+2:n);
VVk1=vVcc(k+1:n-1);
IIk1=iIcCc(k+1:n-1);
VVk2=vVcc(k+2:n);
VVk=vVcc(k:n-2);
NB=0.000005*((VVk2+VVk1).*(IIk1+IIk)-(IIk2+IIk1).*(VVk1+VVk));
DB=(IIk1+IIk).*(IIk2-IIk1)-(IIk2+IIk1).*(IIk1-IIk);
KB=NB./DB;
KBF=filter(Hd,KB);
plot (KBF);
%-----
%-----
```



**Figura 4.7.** Valor base obtenido del la ecuación diferencial para el coeficiente kl.

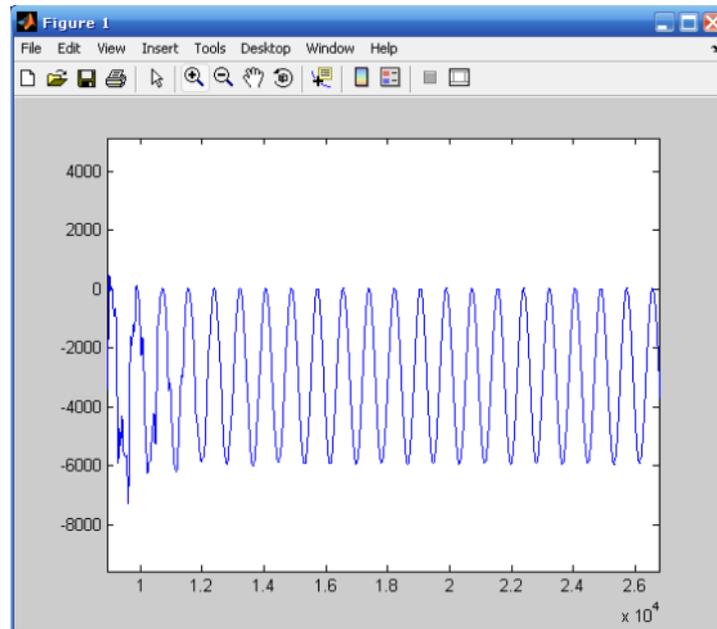
Para la solución y obtención de los parámetros base que utilizamos para tener el valor en por unidad del estimado al punto de falla para kr utilizamos la solución del la ecuación diferencial obtener el valor de referencia para la longitud total de la línea de transmisión pero para este caso del elemento resistivo.

```
%-----
% Desarrollo Del Algoritmo De La Ecuación Diferencial
%-----
```

```

>> !copy c:\atpgnu\T100.pl4 "C:\Archivos de programa\MATLAB71\work"
!PI42mat4.exe T100.pl4 T100.mat;
!copy c:\atpgnu\T10.pl4 "C:\Archivos de programa\MATLAB71\work"
!PI42mat4.exe T10.pl4 T10.mat;
clear all
load 'T100.mat';
load 'T10.mat';
load 'filtrobutterworth.mat';
% Elementos para el calculo De Parámetros Base Para El Coeficiente Kr De La Ec. Diferencial
k=1;
n=80001;
IIk=ilcCc(k:n-2);
IIk2=ilcCc(k+2:n);
VVk1=vVcc(k+1:n-1);
IIk1=iIcCc(k+1:n-1);
VVk2=vVcc(k+2:n);
VVk=vVcc(k:n-2);
% Calculo de parametros base kr
NBR=((VVk1+VVk).*(IIk2-IIk1))-((IIk1-IIk).*(VVk2-VVk1));
DBR=(IIk1+IIk).*(IIk2-IIk1)-(IIk2+IIk1).*(IIk1-IIk);
KBR=NBR./DBR;
KBRF=filter(Hd,KBR);
plot (KBRF);
%-----

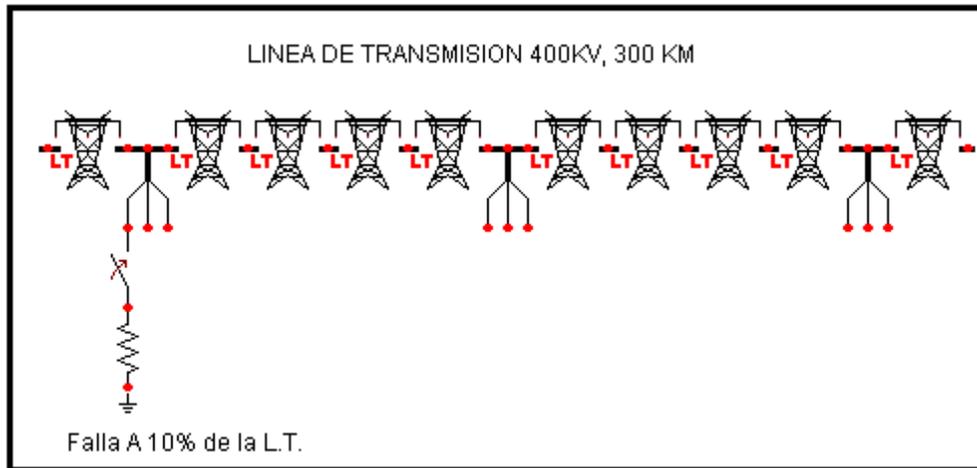
```



**Figura 4.8.** Valor base obtenido de la ecuación diferencial para el coeficiente  $kr$ .

#### 4.4.1. Desarrollo del Algoritmo de la Ecuación Diferencial con una línea de transmisión con carga y sin filtrado.

Ya obtenidos los valores base para la referencia de la línea de transmisión se realiza el cálculos para una falla al 10 % de la longitud de la línea sin filtro ante una falla monofásica.



**Figura 4.9.** Modelo del corto circuito al 10% de la línea de transmisión.

Línea de comandos en Matlab para una falla monofásica sin filtro de Butterworth, este análisis es realizado con carga.

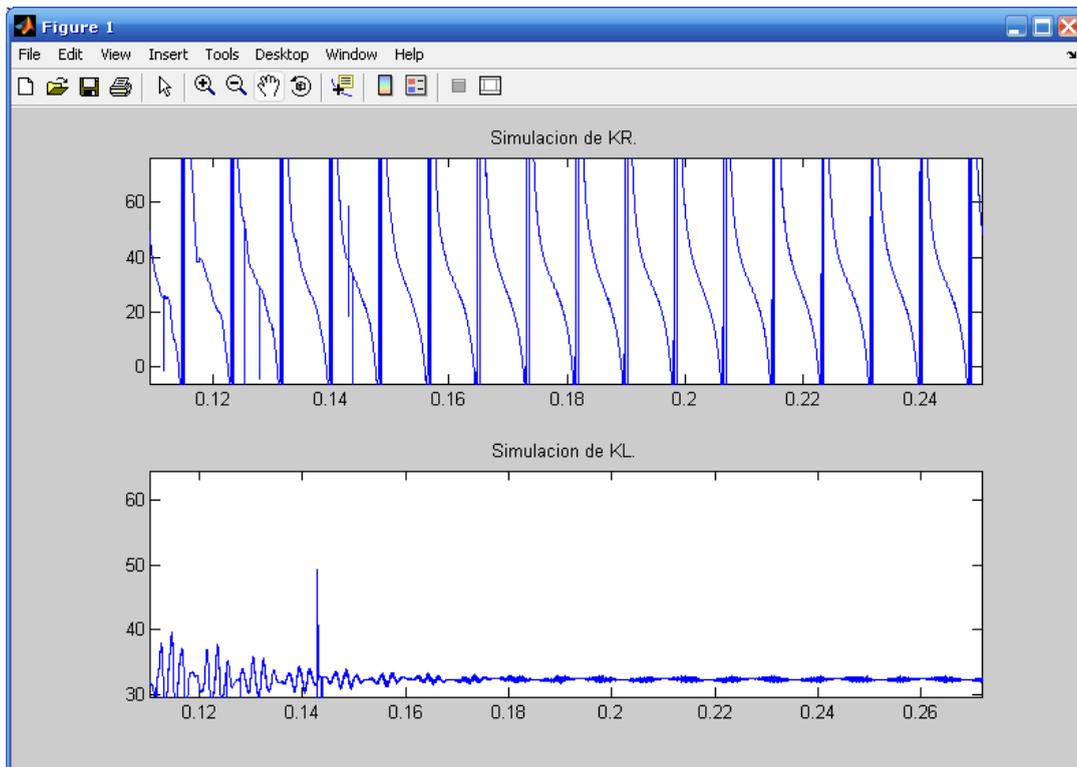
```
%-----
% Desarrollo Del Algoritmo De La Ecuación Diferencial
%-----
>> !copy c:\atpgnu\T100.pl4 "C:\Archivos de programa\MATLAB71\work"
>> !Pl42mat4.exe T100.pl4 T100.mat;
>> !copy c:\atpgnu\T10.pl4 "C:\Archivos de programa\MATLAB71\work"
>> !Pl42mat4.exe T10.pl4 T10.mat;
>> clear all
>> load 'T100.mat';
load 'T10.mat';
%Calculo de parámetros base
k=1;
n=30001;
T=t(k:n-2);
Ik=iIcCc(k:n-2);
Ik2=iIcCc(k+2:n);
VVk1=vVcc(k+1:n-1);
Ik1=iIcCc(k+1:n-1);
```

```

VVk2=vVcc(k+2:n);
VVk=vVcc(k:n-2);
NB=0.000005*((VVk2+VVk1).*(IIk1+IIk)-(IIk2+IIk1).*(VVk1+VVk));
DB=(IIk1+IIk).*(IIk2-IIk1)-(IIk2+IIk1).*(IIk1-IIk);
KB=NB./DB;
Ik=iIcC(k:n-2);
Ik1=iIcC(k+1:n-1);
Ik2=iIcC(k+2:n);
Vk=vVc(k:n-2);
Vk1=vVc(k+1:n-1);
Vk2=vVc(k+2:n);
N=0.000005*((Vk2+Vk1).*(Ik1+Ik)-(Ik2+Ik1).*(Vk1+Vk));
D=(Ik1+Ik).*(Ik2-Ik1)-(Ik2+Ik1).*(Ik1-Ik);
K=N./D;
KL=300*K./KB;
NBR=((VVk1+VVk).*(IIk2-IIk1)-(IIk1-IIk).*(VVk2-VVk1));
DBR=(IIk1+IIk).*(IIk2-IIk1)-(IIk2+IIk1).*(IIk1-IIk);
KBR=NBR./DBR;
NR=(Vk1+Vk).*(Ik2-Ik1)-(Ik1-Ik).*(Vk2-Vk1);
DR=( Ik1+Ik).*(Ik2-Ik1)-(Ik2+Ik1).*(Ik1-Ik);
KR=NR./DR;
Kr=300*KR./KBR;
figure
subplot(2,1,1); plot(T,Kr);
title('Simulación de KR. ');
subplot(2,1,2); plot(T,KL);
title('Simulación de KL. ');
%-----

```

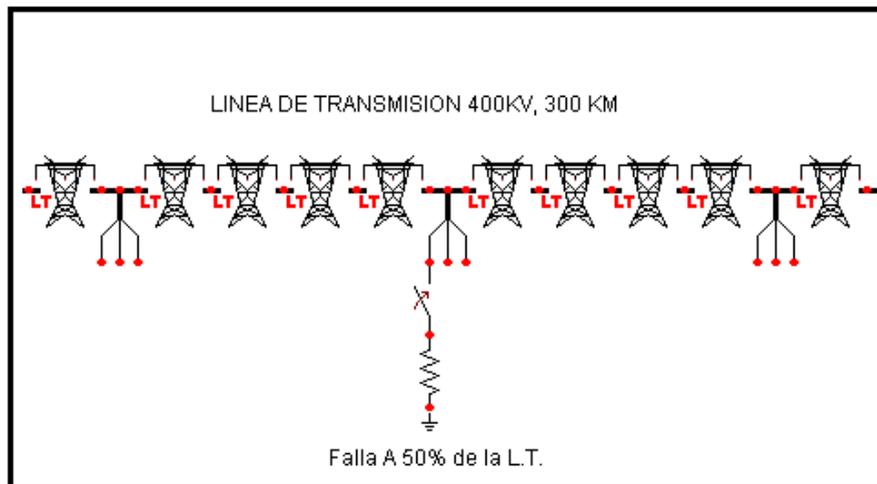
Del resultado de la simulación aplicando el análisis y cálculos correspondientes para las señales sin filtro y operaciones matriciales tenemos:



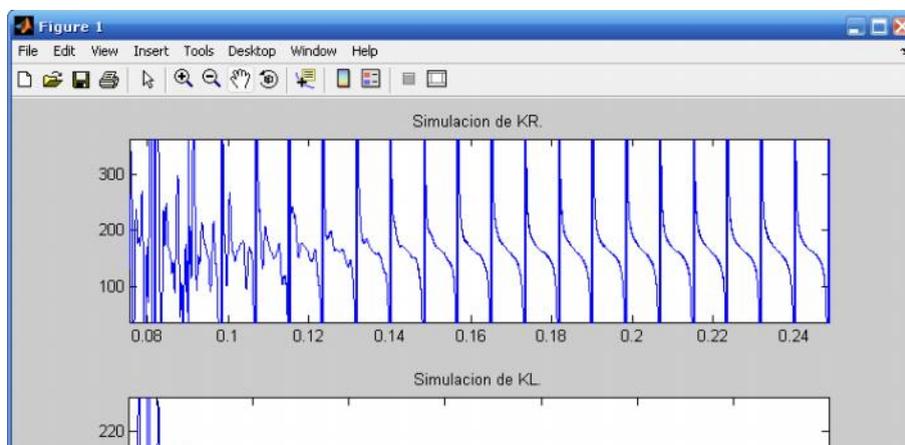
**Figura 4.10.** Estimado de la distancia al 10% de la línea de transmisión con carga y sin filtro.

Se puede apreciar que el resultado del algoritmo en cuanto al estimado al punto fallado el elemento resistivo no se puede ver claramente la distancia donde se encuentra el corto circuito, mientras que para el elemento inductivo se puede apreciar la longitud al punto fallado con un error de aproximadamente 1% de la longitud total de la línea de transmisión. Este resultado se puede ver en los 3 primeros ciclos considerando la frecuencia de comisión federal de electricidad.

También obtenidos los resultados en ATP Draw para la línea de transmisión se realizan los cálculos para una falla al 50 % de la longitud de la línea sin filtro ante una falla monofásica.



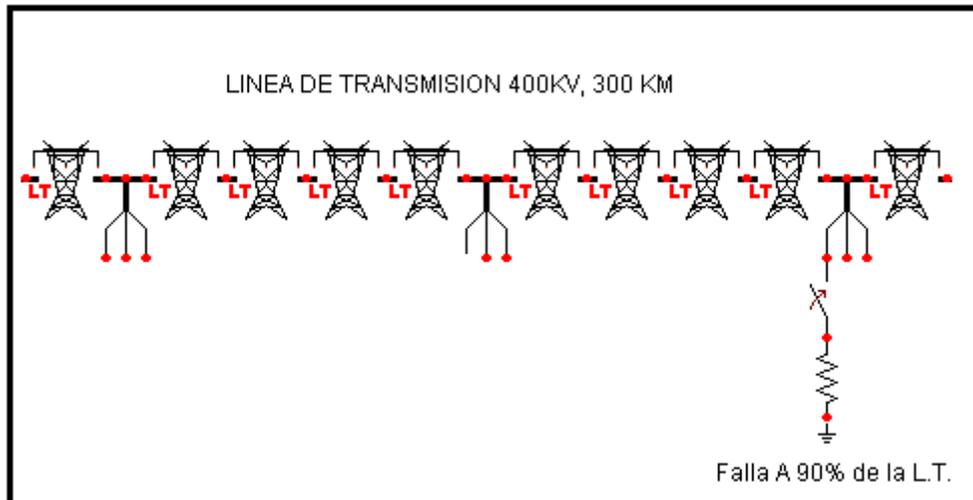
**Figura 4.11.** Modelo del corto circuito al 50% de la línea de transmisión.



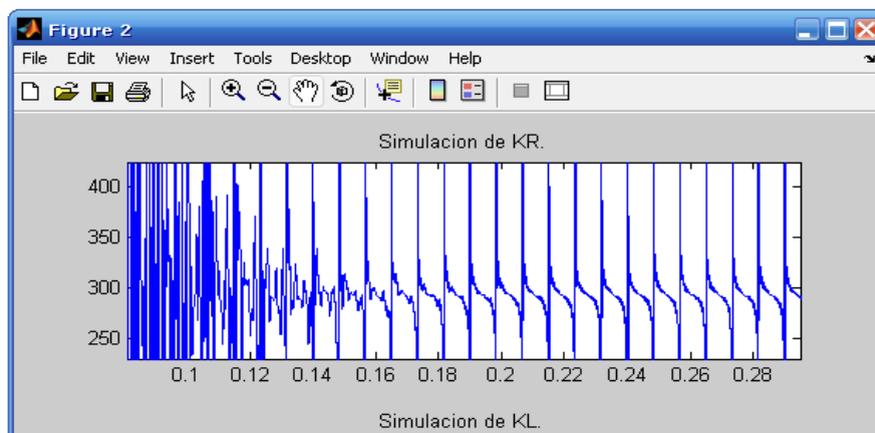
**Figura 4.12.** Estimado de la distancia al 50% de la línea de transmisión sin filtro.

De la misma manera que para la falla el 10% tenemos el mismo comportamiento para el elemento resistivo y un error del 4% del total de la línea de transmisión para el elemento inductivo.

Ahora para la falla al 10% y 50% de la longitud de la línea y obtenidos los valores base para la referencia de la línea de transmisión se realiza el cálculos para una falla al 90 % de la longitud de la línea sin filtro ante una falla monofásica.



**Figura 4.13.** Modelo del corto circuito al 90% de la línea de transmisión.



**Figura 4.14.** *Estimado de la distancia al 90% de la línea de transmisión sin filtro.*

#### **4.4.2. Diseño del filtro de Butterworth para el algoritmo de la ecuación diferencial en Matlab.**

A continuación se demuestra como generar un código para un filtro de Butterworth de 5o. Orden, la frecuencia de corte para esta filtro es relativamente baja a un rango de muestras, permitiendo al filtro hacerlo de su dificultad natural hacerlo mas practico.

Si se tiene una frecuencia de muestreo 30 Khz. y una frecuencia de corte de 500 Hz. primero, crear el objeto para el diseño del filtro que en este caso es para el análisis numérico de una línea de transmisión, entonces se creara un filtro de doble precisión. Convirtiéndolo a una estructura DF1SOS implícita en la configuración de filtros en el comando “filtDES” de Matlab. Finalmente, examinaremos la respuesta a la frecuencia utilizando la herramienta “fvtool” de Matlab.

Tenemos las siguientes constantes a declarar en el programa:

$$F_s = 30001;$$

$$F_n = F_s/2;$$

$$F_{co} = 500;$$

Donde:

$F_s$  = frecuencia de muestreo

$F_n$  = frecuencia de la señal muestreada

$F_{co}$  = frecuencia de corte

Línea de instrucciones para el filtro de Butterworth.

```

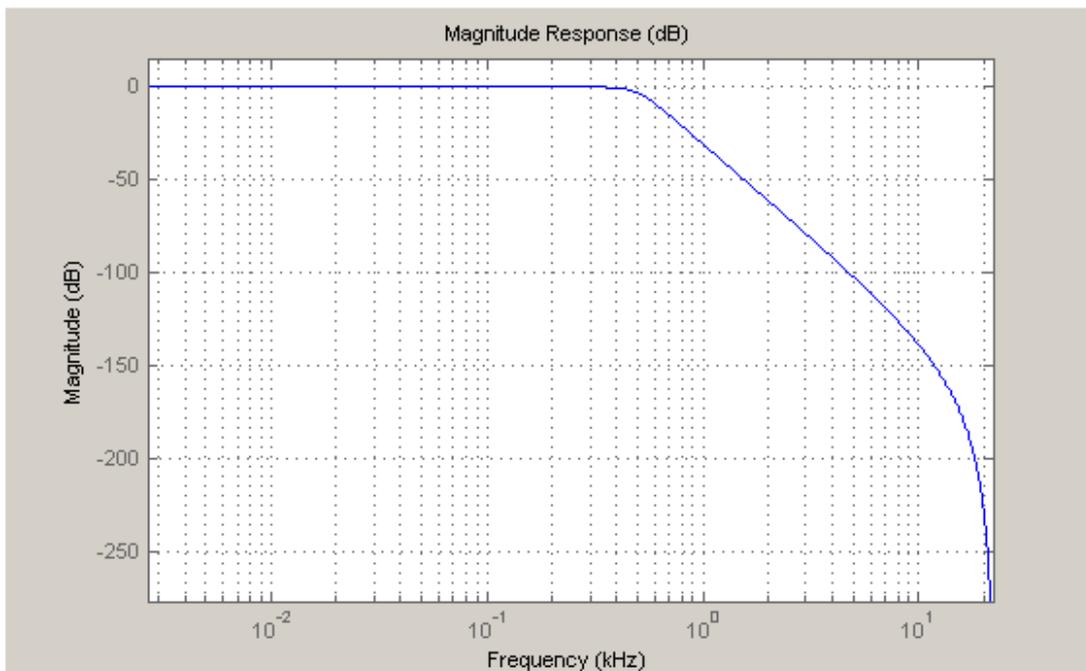
filtDES = fdesign.lowpass('n,fc', 5, Fco, Fs);
Hd = butter(filtDES);
Hd = convert(Hd, 'df1sos');
fvtool(Hd, 'Fs', Fs, 'FrequencyScale', 'log');

```

La forma de utilizar este comando en la ventana de Matlab es la siguiente línea:

```
y = filter(Hd, userstim).
```

De este comando “y” es la señal ya filtrada por el comando “Hd” que contiene las instrucciones para las características de filtrado como son tipo de filtro, frecuencia de corte, orden del filtro, frecuencia de muestreo el análisis numérico de los comandos “fvtool”, “filtDES” y “fdesign.lowpass”. “userstim” es el nombre que corresponde al gráfico a filtrar.



**Figura 4.15.** Filtro de butterworth de orden 5.

#### 4.4.3. Desarrollo del Algoritmo de la Ecuación Diferencial con una línea de transmisión con carga con filtro de Butterworth

Para el desarrollo del algoritmo de la ecuación diferencial se introduce dentro de las líneas de comandos en Matlab filter para los elementos de  $k_l$  y  $k_r$ . Para una falla al 10 % de la longitud de la línea con filtro con una frecuencia de corte de 500 Hz. Ante una falla monofásica las líneas de comandos son las siguientes:

```

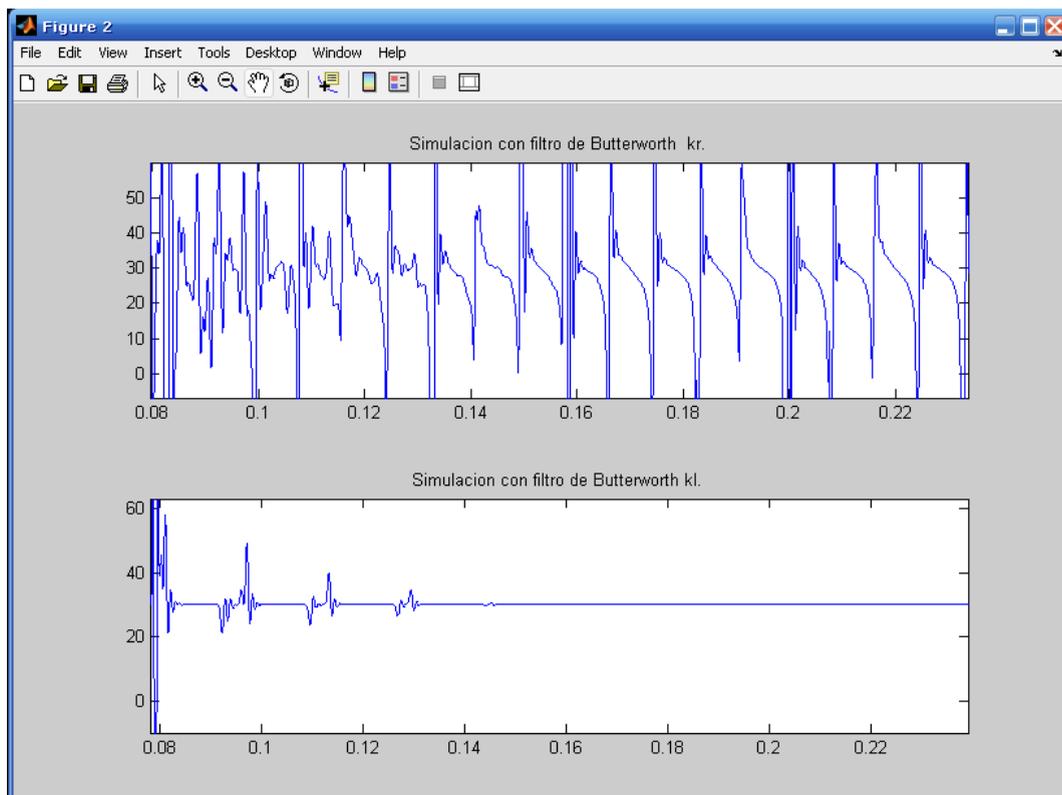
%-----
% Desarrollo Del Algoritmo De La Ecuación Diferencial
%-----
% "DEA" Algoritmo de Proteccion de Distancia
>> !copy c:\atpgnu\T100.pl4 "C:\Archivos de programa\MATLAB71\work"
>> !Pl42mat4.exe T100.pl4 T100.mat;
>> !copy c:\atpgnu\P.pl4 "C:\Archivos de programa\MATLAB71\work"
>> !Pl42mat4.exe P.pl4 P.mat;
>> !copy c:\atpgnu\T10.pl4 "C:\Archivos de programa\MATLAB71\work"
>> !Pl42mat4.exe T10.pl4 T10.mat;
>> %Por medio del programa Pl42mat4.exe pasamos el PL4 a un formato
>> %compatible con MATLAB en este caso .mat
>> clear all
load 'T10.mat';
load 'T100.mat';
load 'P.mat';
load 'filtrobutterworth.mat';
%Para el calculo de calculo de parametros base tenemos:
k=1;% Pertenece a la dimension de la matriz
n=30001;% Pertenece a la dimension de la matriz
T=t(k:n-2);
KB=vKb(k+2:n);
%CALCULO DE PARAMETROS BASE
Iks=iIcC(k:n-2);
Ik=filter(Hd,Iks);
Ik1s=iIcC(k+1:n-1);
Ik1=filter(Hd,Ik1s);
Ik2s=iIcC(k+2:n);
Ik2=filter(Hd,Ik2s);
Vks=vVc(k:n-2);
Vk=filter(Hd,Vks);
Vk1s=vVc(k+1:n-1);
Vk1=filter(Hd,Vk1s);
Vk2s=vVc(k+2:n);
Vk2=filter(Hd,Vk2s);
% numerador y denominador
N=0.000005*((Vk2+Vk1).*(Ik1+Ik)-(Ik2+Ik1).*(Vk1+Vk));
D=(Ik1+Ik).*(Ik2-Ik1)-(Ik2+Ik1).*(Ik1-Ik);
%-----
% Filtrado del numerador y denominador
NF=filter(Hd,N);
DF=filter(Hd,D);
K=NF./DF;
KL=K./KB;
KLF=filter(Hd,KL);
kl=KLF;
kl=300*KLF;

```

```

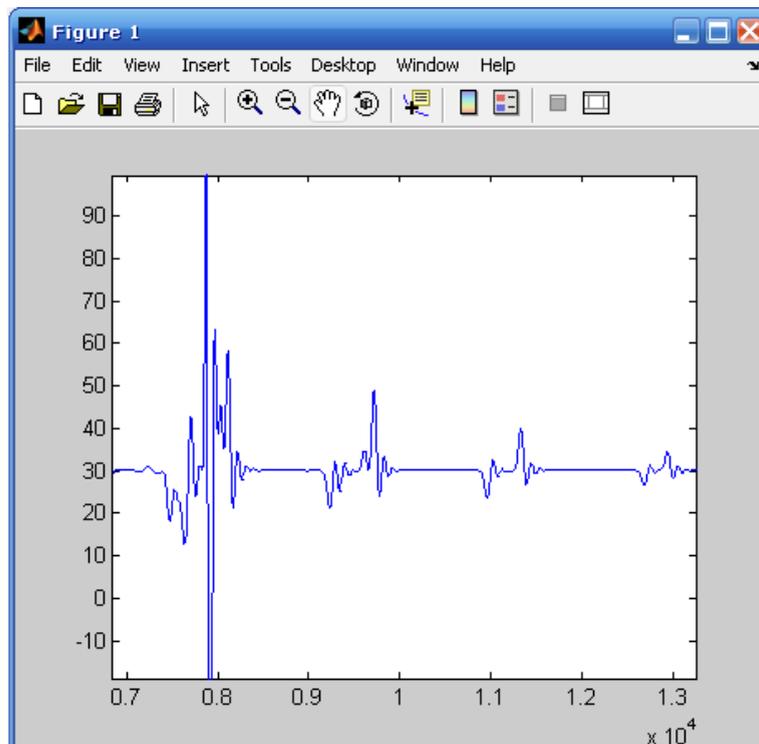
%-----
%Corresponde al calculo de kr
IIk=iIcCc(k:n-2);
IIk2=iIcCc(k+2:n);
VVk1=vVcc(k+1:n-1);
IIk1=iIcCc(k+1:n-1);
VVk2=vVcc(k+2:n);
VVk=vVcc(k:n-2);
NBR=((VVk1+VVk).*(IIk2-IIk1)-(IIk1-IIk).*(VVk2-VVk1));
DBR=(IIk1+IIk).*(IIk2-IIk1)-(IIk2+IIk1).*(IIk1-IIk);
%-----
% Filtrado del numerador y denominador para obtener kr
NBRF=filter(Hd,NBR);
DBRF=filter(Hd,DBR);
KBR=NBRF/DBRF;
NR=((Vk1+Vk).*(Ik2-Ik1)-(Ik1-Ik).*(Vk2-Vk1));
DR=(Ik1+Ik).*(Ik2-Ik1)-(Ik2+Ik1).*(Ik1-Ik);
NRF=filter(Hd,NR);
DRF=filter(Hd,DR);
KR=NRF/DRF;
Kr=KR./KBR;
KrF=filter(Hd,*Kr);
kr=300*KrF;
%-----
% Para la obtención de la grafica de kr y kl
figure
grid;
hold on
subplot(2,1,1); plot(T,kr);
title('Simulacion con filtro de Butterworth kr. ');
subplot(2,1,2); plot(T,kl);
title('Simulación con filtro de Butterworth kl. ');
%-----
%-----

```



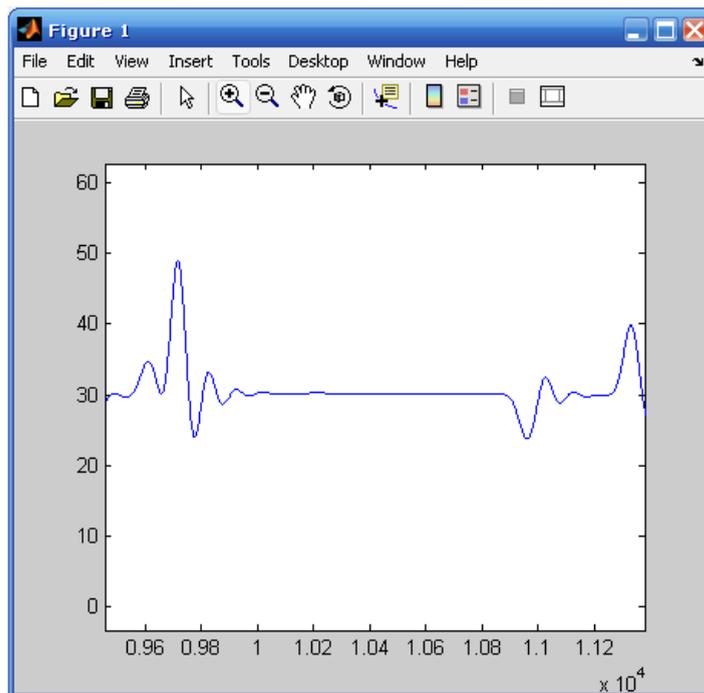
**Figura 4.16.** *Estimado de la distancia al 10% de la línea de transmisión con filtro de Butterworth.*

Ahora la señal obtenida de la simulación realizada en el programa ATP Draw podemos claramente apreciar la exactitud del estimado al punto fallado. En el caso del elemento resistivo se puede ver zonas en la curva donde tiene una característica a la altura de los 30 Km. la grafica tiende a mantenerse en ese valor como en el elemento inductivo. El valor del estimado al punto fallado en el elemento inductivo tiene un error de menos del 0.5% de la longitud total de la línea de transmisión. E incluso antes de los tres ciclos ya se puede apreciar la distancia al punto fallado. Como se aprecia en la siguiente figura.

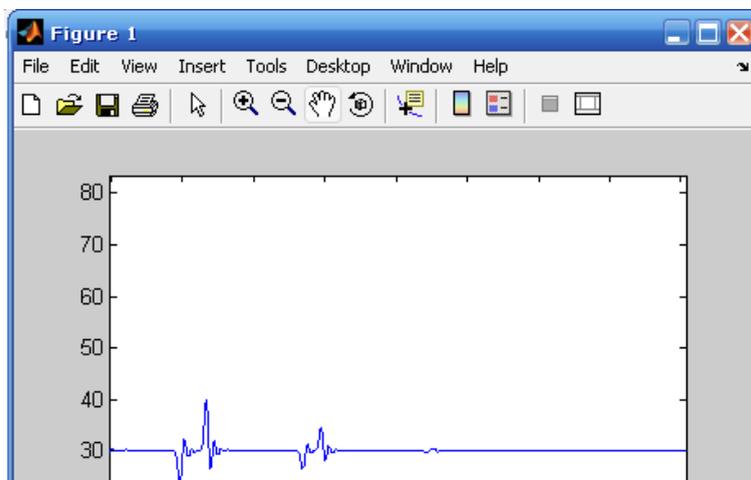


**Figura 4.17.** *Estimado kl de la distancia al 10% de la línea de transmisión con filtro de Butterworth.*

Como ya se menciona en el elemento inductivo claramente se muestra la distancia al punto fallado, para este caso se realizo la simulación de un corto circuito monofásico a 30 Km. en una líneas de transmisión de 300 Km.

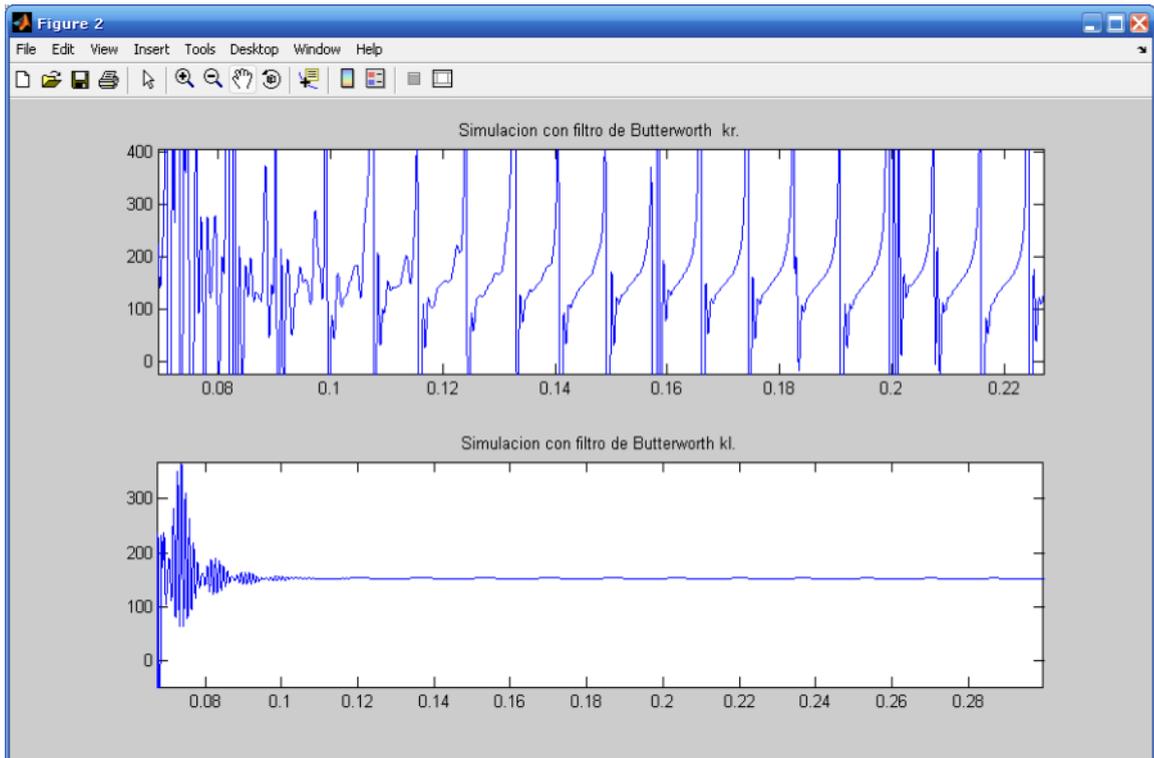


**Figura 4.18.** Estimado  $kl$  de la distancia al 10% de la línea de transmisión con filtro de Butterworth en los tres ciclos después de la falla.

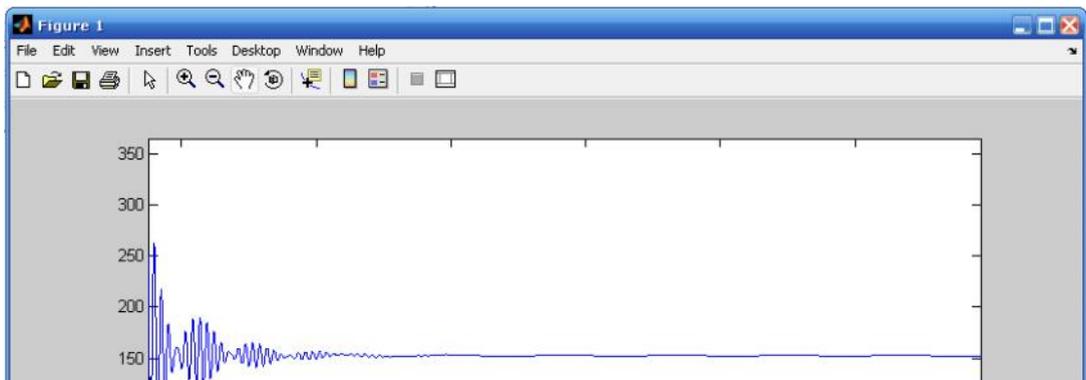


**Figura 4.19.** Estimado  $k_l$  de la distancia al 10% de la línea de transmisión con filtro de Butterworth después de tres ciclos de ocurrida la falla.

Falla al 50 % de la longitud de la línea con filtro con una frecuencia de corte de 500 Hz. Con una falla monofásica.

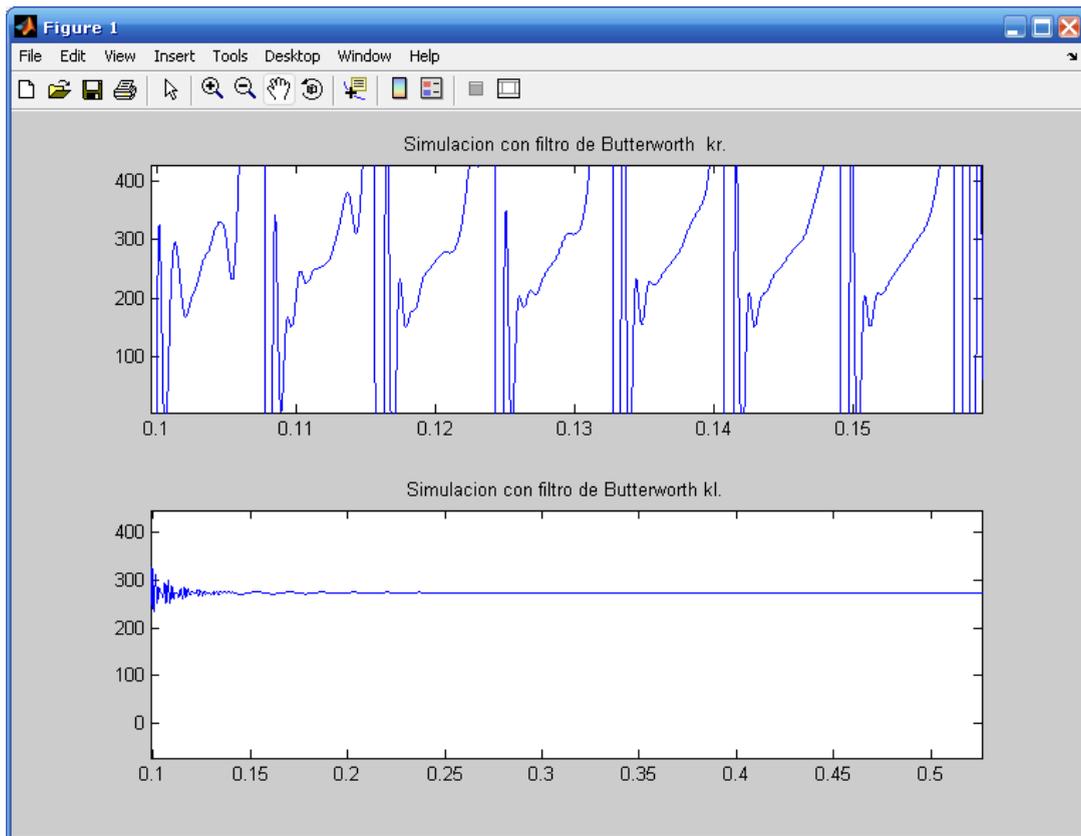


**Figura 4.20.** Estimado de la distancia al 50% de la línea de transmisión filtro de Butterworth.



**Figura 4.21.** *Estimado  $kl$  de la distancia al 50% de la línea de transmisión filtro de Butterworth con menos de tres ciclos de ocurrida la falla.*

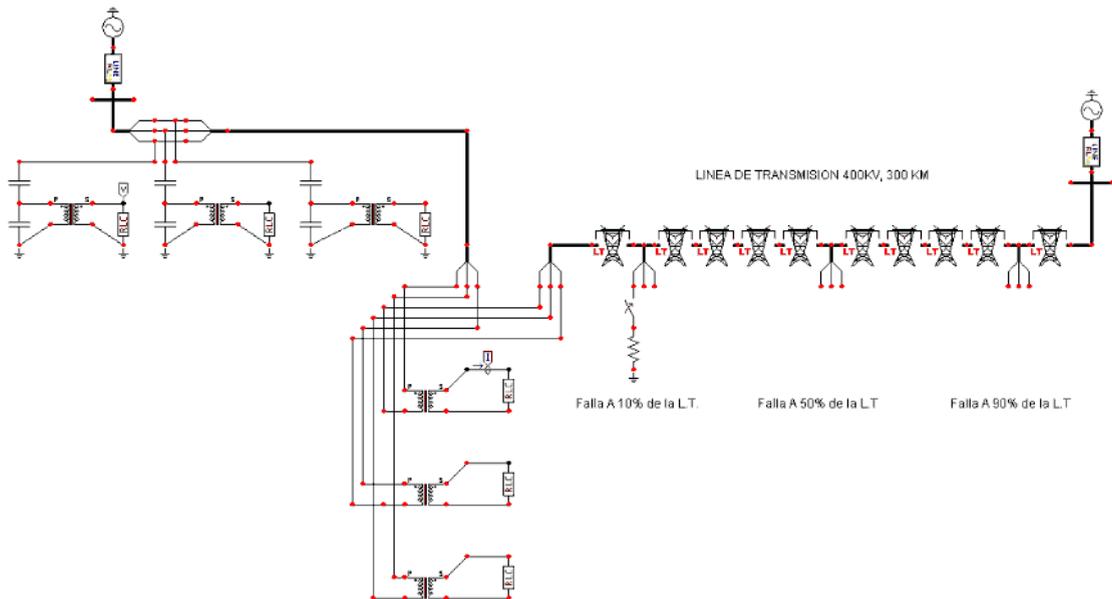
Falla al 90 % de la longitud de la línea con filtro con una frecuencia de corte de 500 Hz. Con una falla monofásica.



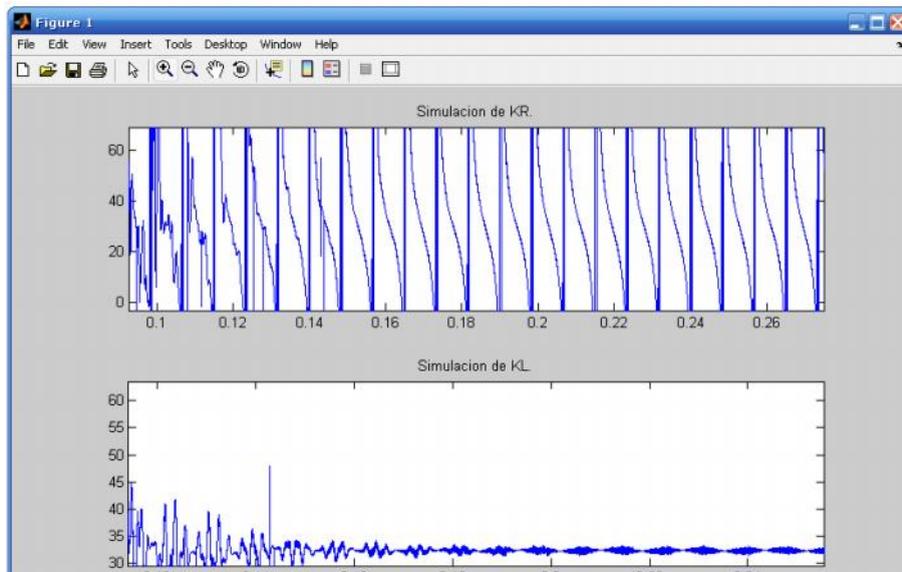
**Figura 4.22.** *Estimado de la distancia al 90% de la línea de transmisión filtro de Butterworth.*

#### 4.4.4. Desarrollo del Algoritmo de la Ecuación Diferencial con una línea de transmisión sin carga, sin filtrado.

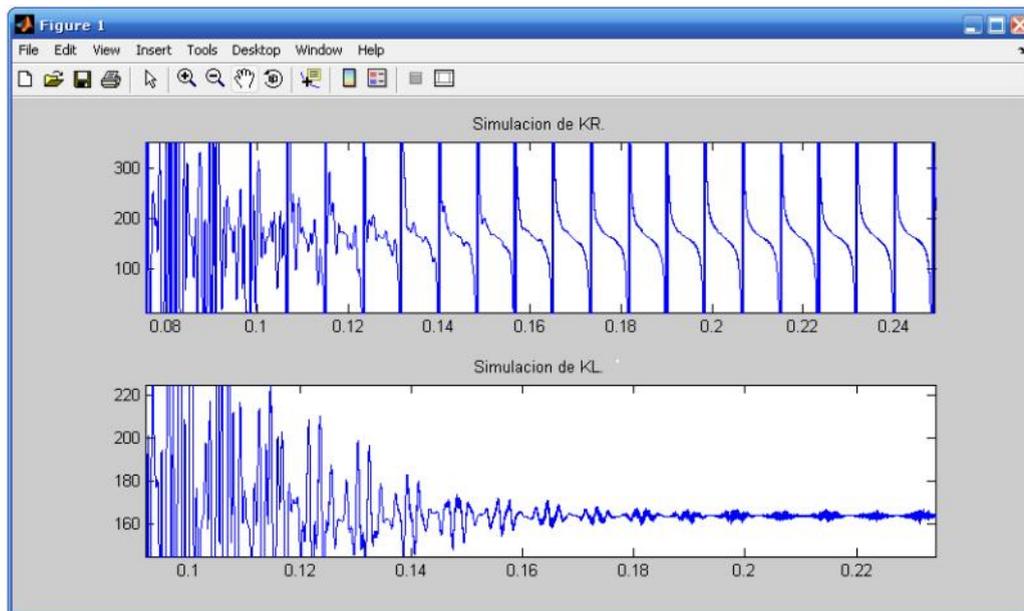
Para la simulación de la línea de transmisión es utilizado el mismo modelo bajo estudio, solo que la carga que alimentaba el sistema de potencia se elimina y se realizan las simulaciones para el corto circuito monofásico para las distancias del 10%, 50% y 90% de la longitud de la línea de transmisión.



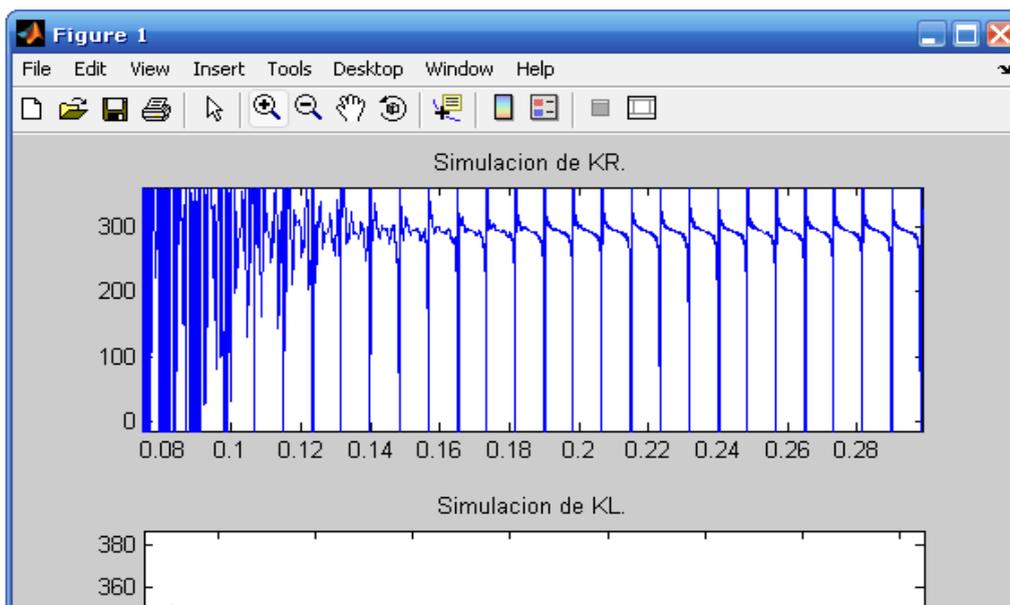
**Figura 4.23.** Modelo de la línea de transmisión con unidades medición de corriente y voltaje para análisis con carga.



**Figura 4.24.** Estimado de la distancia al 10% de la línea de transmisión sin carga y sin filtro.



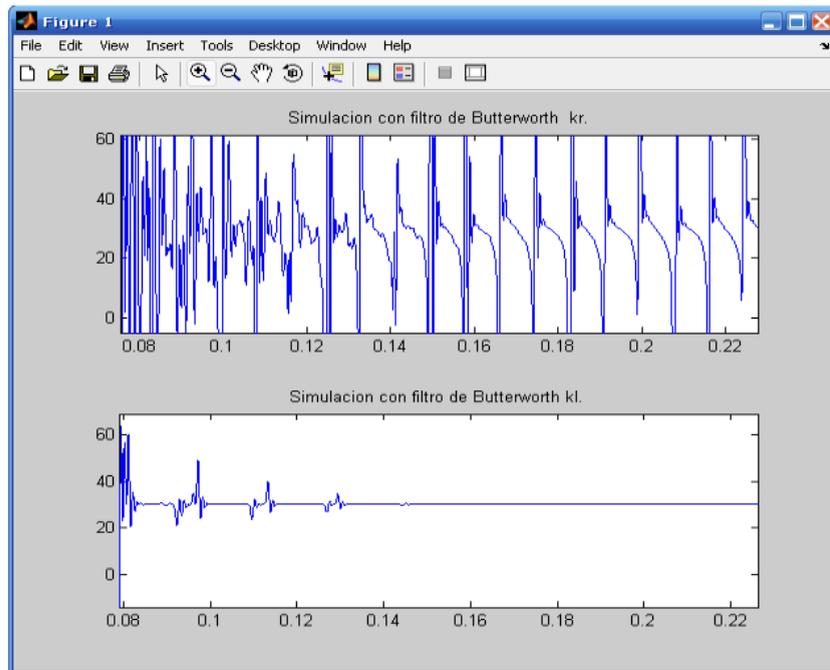
**Figura 4.25.** Estimado de la distancia al 50% de la línea de transmisión sin carga y sin filtro.



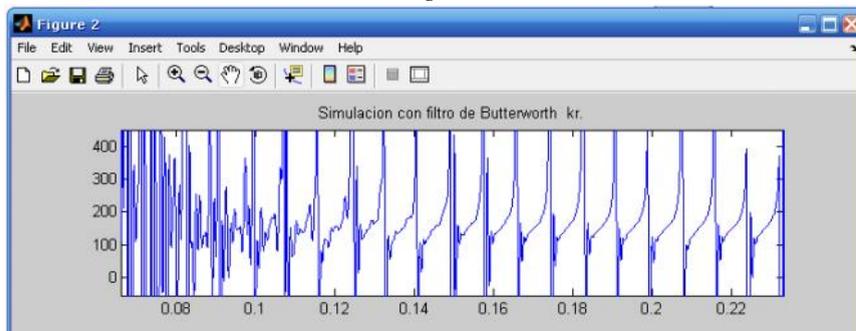
**Figura 4.26.** Estimado de la distancia al 90% de la línea de transmisión sin carga y sin filtro.

**4.4.5. Desarrollo del Algoritmo de la Ecuación Diferencial con una línea de transmisión sin carga con filtro de Butterworth.**

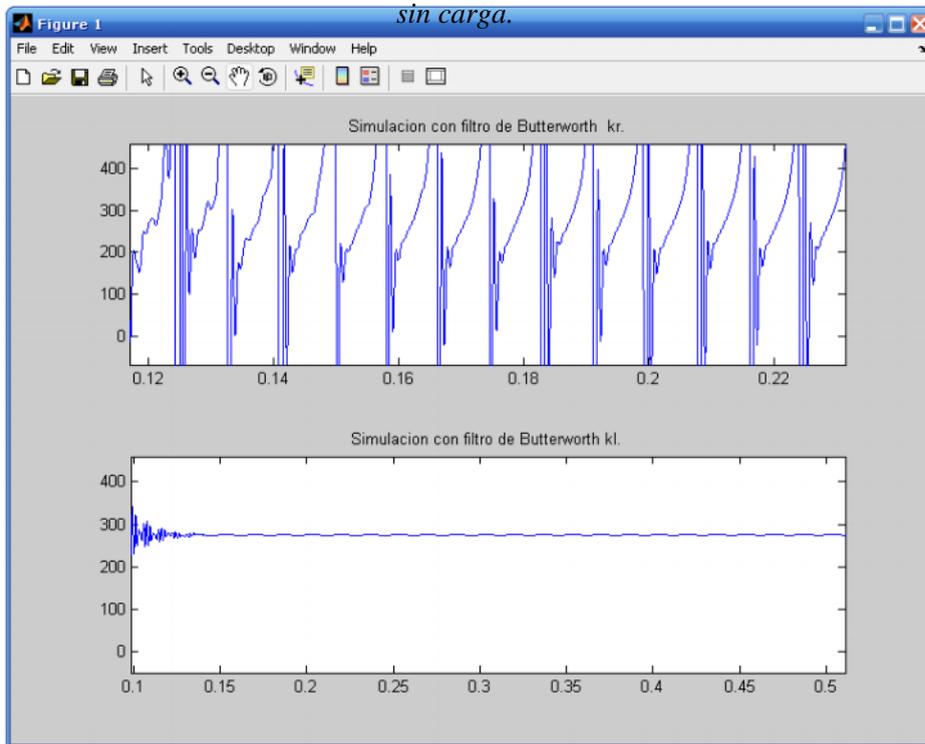
Para el desarrollo del algoritmo sin carga se utiliza la misma línea de comandos en Matlab para la obtención del estimado al punto fallado así como para la obtención del filtrado del numerador y denominador para  $k_l$  y  $k_r$ .



**Figura 4.27.** Estimado de la distancia al 10% de la línea de transmisión con filtro de Butterworth sin carga.

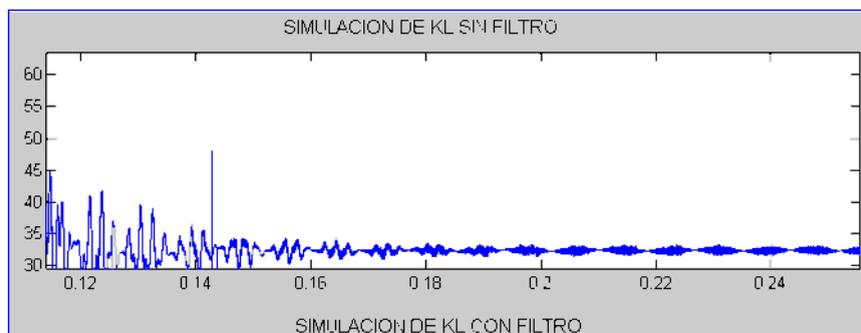


**Figura 4.28.** Estimado de la distancia al 50% de la línea de transmisión con filtro de Butterworth



**Figura 4.29.** Estimado de la distancia al 90% de la línea de transmisión con filtro de Butterworth sin carga.

Las graficas para el estimado al punto fallado para el sistema sin carga son obtenidas por el mismo algoritmo en Matlab para el sistema con carga. En la siguiente figura se muestra la diferencia entre la simulación con y sin filtro de Butterworth.



**Figura 4.30.** *Comparación de la simulación con y sin filtro de Butterworth.*