

## Practicas de MATLAB:

Comenzaremos con los comandos básicos, una vez conocido los más fundamentales los mezclaremos con otros comandos más complejos para poder desarrollar un mejor entendimiento del programa.

Dado a que este no es un programa de desarrollo visual como se vio anteriormente con Visual Basic es irremediable que la práctica se haga sobre problemas reales, ya que fundamentalmente el MATLAB es una especie de calculadora muy poderosa que nos servirá para resolver problemas.

### Práctica - N° 1: Manejo de Matrices.

1) Ingresar las siguientes matrices al MATLAB.

$$m1 = \begin{bmatrix} 1 & -7 & 4 \\ -8 & 3 & 6 \\ 7 & 8 & 3 \end{bmatrix} \quad m2 = \begin{bmatrix} 10 & -17 & 40 \\ 18 & 3 & 5 \\ 7 & 83 & 34 \end{bmatrix} \quad m3 = \begin{bmatrix} 2 & 7 \\ 5 & -6 \end{bmatrix}$$
$$m4 = \begin{bmatrix} 10 & 14 & 122 \\ -18 & 3 & 56 \\ 70 & -66 & 88 \end{bmatrix} \quad m5 = \begin{bmatrix} 15 & -79 & 0 \\ -81 & 30 & -15 \\ 76 & -13 & 1 \end{bmatrix}$$

2) Guardar las matrices con el nombre matrices\_01.mat

3) Calcular:

3.1 Calcular las transpuestas  $\rightarrow$  mi.'

3.2 Calcular el determinante  $\rightarrow$  det(mi)

3.3 Calcular la inversa  $\rightarrow$  inv(mi)

4) Guardar los nuevos resultados como matrices\_02.mat

5) Realicemos un cálculo:

$$cal\_01 = (m1.*m2) + (m4./m5)$$

6) Comparemos:

$$comp\_01 = m1 > m2$$

$$comp\_02 = m1 < m2$$

$$comp\_03 = m1 == m2$$

$$comp\_04 = (m1 + m2) >= (m4 + m5)$$

$$comp\_05 = m1([2 \ 3], [2 \ 3]) > m3$$

7) Comparaciones especiales:

$$comp\_06 = isinf((m1.*m2) + (m4./m5))$$

$$comp\_07 = isnan((m2-3)./(m5-30))$$

$$comp\_08 = any(comp\_07)$$

$$comp\_07 = all(m2-3)$$

8) Guardar los nuevos resultados como matrices\_03.mat

9) borrar memorias con el comando "clear".

## Práctica - N° 2: Manejo de Números Complejos.

1) Ingresar los siguientes complejos:

$$d01=(4+7i)$$

$$d02=(8-2i)$$

$$d03=(-4+3i)$$

$$d04=(-6-2i)$$

2) Calcular:

$$res01=d01+d02+d03+d04$$

$$res02=d02*d03$$

$$res03=d02/d03$$

3) Extraer y calcular:

$$real(res01)$$

$$imag(res02)$$

$$conj(res03)$$

$$angle(res01+res02)$$

## Práctica - N° 3: Manejo de Archivos \*.M

Resolver una solución cuadrática verificar y graficar con un sencillo archivo \*.M

1) Crear un archivo llamado cuadratica\_01.m

2) Ingresar estos datos:

**%Necesitamos calcular la solución de la ecuación cuadrática**

**%ax<sup>2</sup>+bx+c=0**

**%Y si a=-2, b=4 y c=3**

**a=-2;**

**b=4;**

**c=3;**

**%Lo hacemos utilizando la ecuación cuadrática, obteniendo las dos soluciones**

**x1=[-b+sqrt(b<sup>2</sup>-4\*a\*c)]/(2\*a)**

**x2=[-b-sqrt(b<sup>2</sup>-4\*a\*c)]/(2\*a)**

**pause**

**%Podemos verificar que el resultado es correcto, reemplazando los valores obtenidos en la ecuación y viendo si el resultado es cero**

**v1=a\*x1<sup>2</sup>+b\*x1+c**

**v2=a\*x2<sup>2</sup>+b\*x2+c**

**pause**

**x=[x1-2:0.1:x2+2]**

**y=a\*x.<sup>2</sup>+b\*x+c;**

**plot(x,y)**

**gris**

## Práctica - N° 4: Programación.

Dado el sistema de ecuaciones hallar los valores  $X_1$ ,  $X_2$ ,  $X_3$ .

$$3X_1 - 5X_2 - 7X_3 = 40$$

$$8X_1 - 4X_2 - 2X_3 = -12$$

$$6X_1 - 7X_2 - 3X_3 = -41$$

1) Construir un fichero valores\_01.m

2) Ingresar:

```
dm=det(m);
mx1=m;
mx1(1)=n(1);
mx1(2)=n(2);
mx1(3)=n(3);
dmx1=det(mx1);
mx2=m;
mx2(4)=n(1);
mx2(5)=n(2);
```

```
mx2(6)=n(3);
dmx2=det(mx2);
mx3=m;
mx3(7)=n(1);
mx3(8)=n(2);
mx3(9)=n(3);
dmx3=det(mx3);
x=[(dmx1/dm) (dmx2/dm)
(dmx3/dm)]
```

3) Construir un fichero valores\_02.m

```
% Textos Superior:
txt_01=uicontrol(gcf,...
    'Style','text','String','X1',...
    'Position',[40 375 50 22]);
txt_02=uicontrol(gcf,...
    'Style','text','String','X2',...
    'Position',[100 375 50 22]);
txt_03=uicontrol(gcf,...
    'Style','text','String','X3',...
    'Position',[160 375 50 22]);

% Valores X1:
pos_01 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[40 350 50 22],...
    'Callback','m_01=str2double(get(pos_01,'String'))');

pos_02 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[40 325 50 22],...
    'Callback','m_02=str2double(get(pos_02,'String'))');

pos_03 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[40 300 50 22],...
    'Callback','m_03=str2double(get(pos_03,'String'))');
```

```
% Valores X2:
pos_04 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[100 350 50 22],...
    'Callback','m_04=str2double(get(pos_04,'String'))');

pos_05 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[100 325 50 22],...
    'Callback','m_05=str2double(get(pos_05,'String'))');

pos_06 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[100 300 50 22],...
    'Callback','m_06=str2double(get(pos_06,'String'))');

% Valores X3:
pos_07 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[160 350 50 22],...
    'Callback','m_07=str2double(get(pos_07,'String'))');

pos_08 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[160 325 50 22],...
    'Callback','m_08=str2double(get(pos_08,'String'))');

pos_09 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[160 300 50 22],...
    'Callback','m_09=str2double(get(pos_09,'String'))');

% Valores de los terminos Independientes:
npos_01 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[220 350 50 22],...
    'Callback','n_01=str2double(get(npos_01,'String'))');
```

```

npos_02 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[220 325 50 22],...
    'Callback','n_02=str2double(get(npos_02,'String'))');

npos_03 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','white',...
    'FontSize',10,'FontName','Arial',...
    'String',[],...
    'Position',[220 300 50 22],...
    'Callback','n_03=str2double(get(npos_03,'String'))');

% Botón que realiza el calculo:
boton_calculo = uicontrol(gcf,...
    'Style','push',...
    'FontSize',10,'FontName','Arial',...
    'String','Calcular',...
    'Position',[50 250 150 22],...
    'Callback','valores_03');

```

3) Construir un fichero valores\_03.m

% Resolución:

```

m=[m_01 m_04 m_07;m_02 m_05 m_08;m_03 m_06 m_09]
n=[n_01;n_02;n_03]
valores_01;

```

% Textos:

```

txt_01=uicontrol(gcf,...
    'Style','text','String','X1',...
    'Position',[280 350 50 22]);
txt_02=uicontrol(gcf,...
    'Style','text','String','X2',...
    'Position',[280 325 50 22]);
txt_03=uicontrol(gcf,...
    'Style','text','String','X3',...
    'Position',[280 300 50 22]);

```

% Resultados:

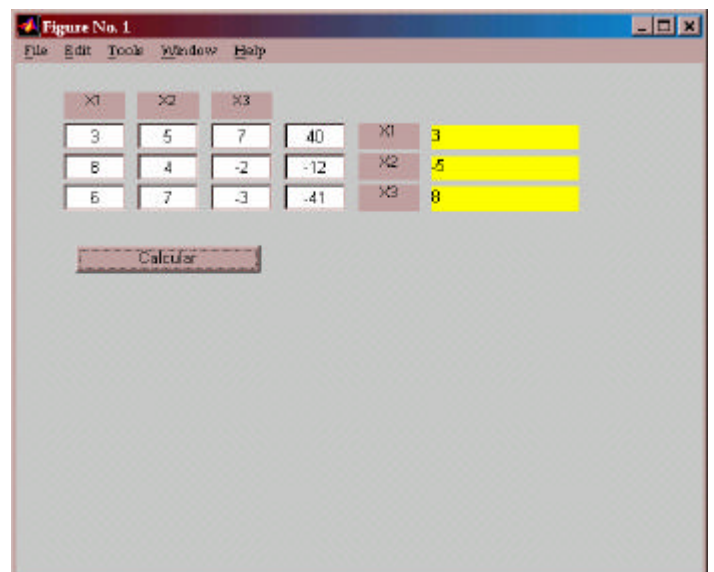
```

res_01 = uicontrol(gcf,...
    'Style','text',...
    'BackgroundColor','y',...
    'FontSize',10,'FontName','Arial',...
    'HorizontalAlignment','Left',...
    'String',[x(1)],...
    'Position',[340 350 120 20]);

res_02 = uicontrol(gcf,...
    'Style','text',...
    'BackgroundColor','y',...
    'FontSize',10,'FontName','Arial',...
    'HorizontalAlignment','Left',...
    'String',[x(2)],...
    'Position',[340 325 120 20]);

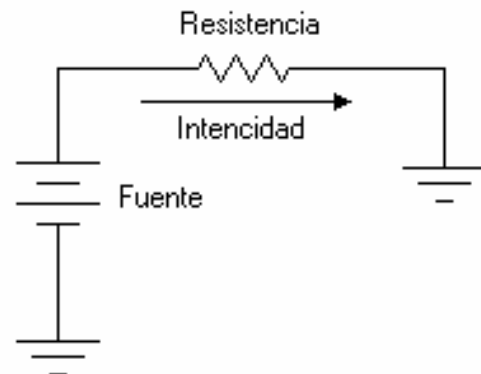
res_03 = uicontrol(gcf,...
    'Style','text',...
    'BackgroundColor','y',...
    'FontSize',10,'FontName','Arial',...
    'HorizontalAlignment','Left',...
    'String',[x(3)],...
    'Position',[340 300 120 20]);

```



## Práctica - N° 5: Programación.

Dado el siguiente circuito eléctrico resolver con la ley de (OHM) si el sistema Funciona bien, Funciona en el limite o Se Quema.



1) Construir un fichero cal\_01.m

```
tol=5
va=1
vb=0
vc=0
```

```
% Definir el exto de titulo para este grupo de controles
```

```
txt_tdir = uicontrol(gcf,...
'Style','text',...
'BackgroundColor',[ 0.8 0.8 0.8 ],...
'FontSize',10,'FontName','Arial',...
'HorizontalAlignment','Left',...
'String','Tolerancia de la Resistencia',...
'Position',[350 380 250 20]);
```

```
% Definir la propiedad TickDir In con radiobutton (defecto)
```

```
td_5 = uicontrol(gcf,...
'Style','radio',...
'BackgroundColor',[ 0.8 0.8 0.8 ],...
'FontSize',10,'FontName','Arial',...
'String','+/- 5%',...
'Position',[350 355 75 25],...
'Value',va,...
'Callback','set(td_5,''Value'',1),set(td_10,''Value'',0),set(td_15,''Value'',0),va
=1,vb=0,vc=0,tol=5');
```

```
% Definir la propiedad TickDir Out con radiobutton (defecto)
```

```
td_10 = uicontrol(gcf,...
'Style','radio',...
'BackgroundColor',[ 0.8 0.8 0.8 ],...
'FontSize',10,'FontName','Arial',...
'String','+/- 10%',...
'Position',[350 330 75 25],...
'Value',vb,...
'Callback','set(td_5,''Value'',0),set(td_10,''Value'',1),set(td_15,''Value'',0),va
=0,vb=1,vc=0,tol=10');
```

```
% Definir la propiedad TickDir Out con radiobutton (defecto)
```

```
td_15 = uicontrol(gcf,...
'Style','radio',...
'BackgroundColor',[ 0.8 0.8 0.8 ],...
'FontSize',10,'FontName','Arial',...
'String','+/- 15%',...
'Position',[350 305 75 25],...
'Value',vc,...
'Callback','set(td_5,''Value'',0),set(td_10,''Value'',0),set(td_15,''Value'',1),va
=0,vb=0,vc=1,tol=15');
```

```
% Resistencia Texto y Valor:
txt_resistencia = uicontrol(gcf,...
'Style','text',...
'BackgroundColor',[ 0.8 0.8 0.8 ],...
'FontSize',10,'FontName','Arial',...
'HorizontalAlignment','Center',...
'String','Resistencia (ohms)',...
'Position',[40 360 150 30]);
valor_resistencia = uicontrol(gcf,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',14,'FontName','Arial',...
'String','',...
'Position',[40 340 150 30],...
'Callback','r=str2double(get(valor_resistencia,'String'))');

% Potencia Texto y Valor:
txt_potencia = uicontrol(gcf,...
'Style','text',...
'BackgroundColor',[ 0.8 0.8 0.8 ],...
'FontSize',10,'FontName','Arial',...
'HorizontalAlignment','Center',...
'String','Potencia (W)',...
'Position',[40 266 150 30]);
valor_potencia = uicontrol(gcf,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',14,'FontName','Arial',...
'String','',...
'Position',[40 245 150 30],...
'Callback','p=str2double(get(valor_potencia,'String'))');

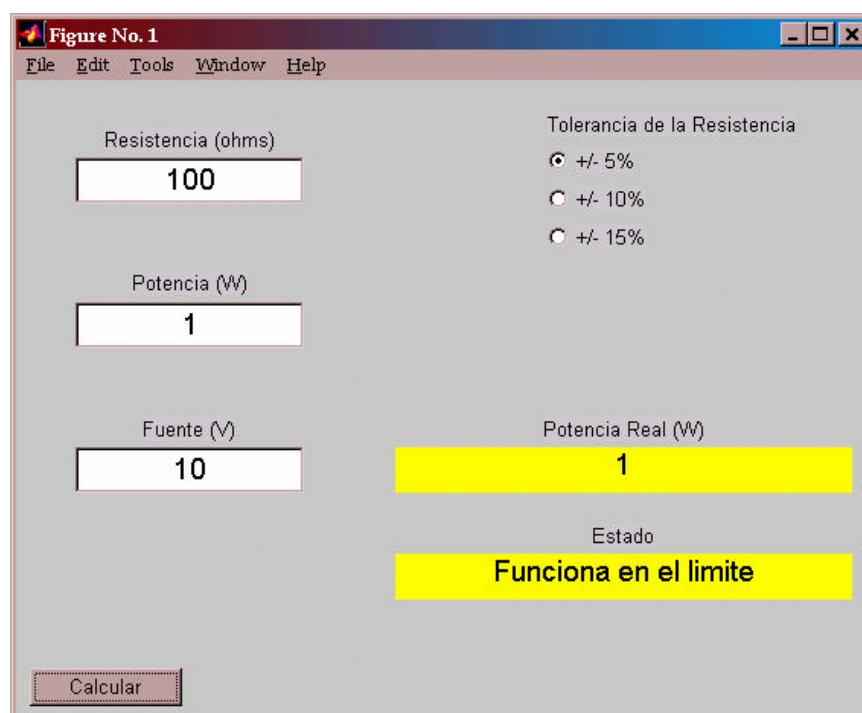
% Fuente Texto y Valor:
txt_fuente = uicontrol(gcf,...
'Style','text',...
'BackgroundColor',[ 0.8 0.8 0.8 ],...
'FontSize',10,'FontName','Arial',...
'HorizontalAlignment','Center',...
'String','Fuente (V)',...
'Position',[40 170 150 30]);
valor_fuente = uicontrol(gcf,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',14,'FontName','Arial',...
'String','',...
'Position',[40 150 150 30],...
'Callback','f=str2double(get(valor_fuente,'String'))');

% Botón que realiza el calculo:
boton_calculo = uicontrol(gcf,...
'Style','push',...
'FontSize',10,'FontName','Arial',...
'String','Calcular',...
'Position',[10 10 100 25],...
'Callback','cal_02');
```

2) Construir un fichero cal\_02.m

```
p_real=(f^2)/r;
a=p*tol/100;
if p_real<=p-a
    s='Funciona bien'
    col='g'
elseif p-a<p_real&p_real<p+a
    s='Funciona en el limite'
    col='y'
elseif p_real>=p+a
    s='Se Quema'
    col='r'
end
% Potencia Real Texto y Valor:
txt_potencia_real = uicontrol(gcf,...
'Style','text',...
'BackgroundColor',[ 0.8 0.8 0.8 ],...
'FontSize',10,'FontName','Arial',...
'String','Potencia Real (W)',...
'Position',[250 170 300 30]);
valor_fuente = uicontrol(gcf,...
'Style','text',...
'BackgroundColor',[col],...
'FontSize',14,'FontName','Arial',...
'String',[p_real],...
'Position',[250 150 300 30]);

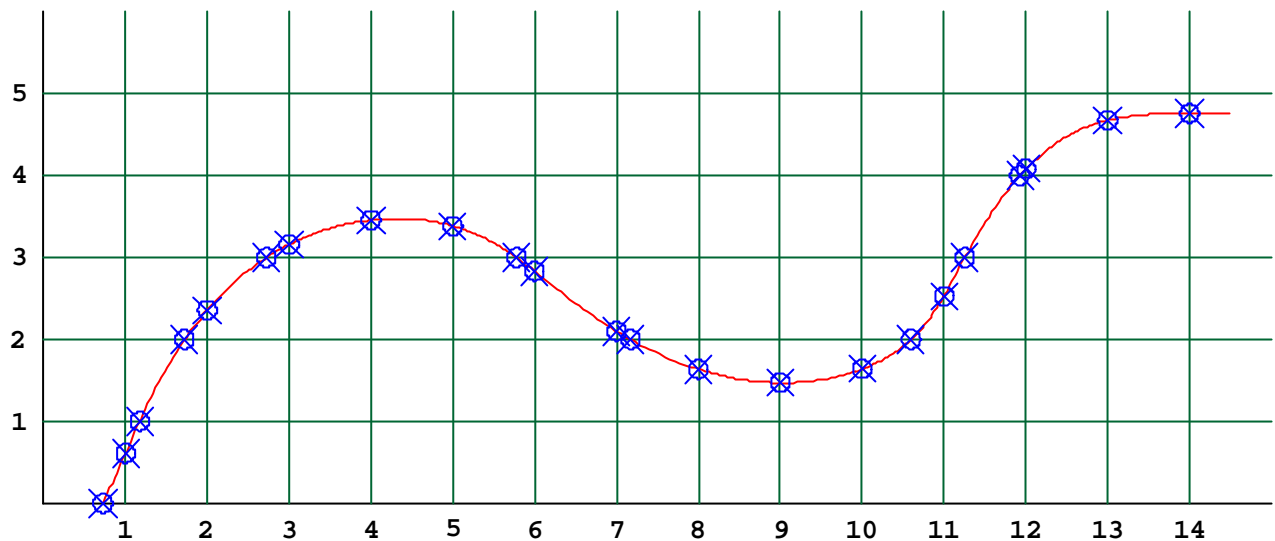
% Estado Texto y Valor:
txt_estado = uicontrol(gcf,...
'Style','text',...
'BackgroundColor',[ 0.8 0.8 0.8 ],...
'FontSize',10,'FontName','Arial',...
'String','Estado',...
'Position',[250 100 300 30]);
valor_fuente = uicontrol(gcf,...
'Style','text',...
'BackgroundColor',[col],...
'FontSize',14,'FontName','Arial',...
'String',[s],...
'Position',[250 80 300 30]);
```





## Práctica - N° 6: Programación.

Trabajaremos ahora con gráficas y como llevar una curva del papel a una o varias funciones que se acoplen a dicha curva.



1) Extraer la mayor cantidad de puntos para llevar al MATLAB.

```
x=[0.72 1.00 1.17 1.72 2.00 2.72 3.00 4.00 5.00 5.78 6.00 7.00 7.17 8.00 9.00
10.00 10.60 11.00 11.26 11.92 12.00 13.00 14.00];
y=[0.00 0.61 1.00 2.00 2.36 3.00 3.16 3.46 3.39 3.00 2.83 2.10 2.00 1.64 1.47 1.64
2.00 2.53 3.00 4.00 4.08 4.68 4.76];
```

2) Construir un fichero curva\_w1.m

```
x=[0.72 1.00 1.17 1.72 2.00 2.72 3.00 4.00 5.00 5.78 6.00 7.00 7.17 8.00 9.00
10.00 10.60 11.00 11.26 11.92 12.00 13.00 14.00];
y=[0.00 0.61 1.00 2.00 2.36 3.00 3.16 3.46 3.39 3.00 2.83 2.10 2.00 1.64 1.47 1.64
2.00 2.53 3.00 4.00 4.08 4.68 4.76];
```

```
fig=gcf; clf;
```

```
axes('position',[.1 .35 .8 .6])
plot(x,y,'o')
grid
```

```
txt_val_a = uicontrol(gcf,...
'Style','text',...
'String','Lim. a:',...
'FontSize',8,'FontName','Arial',...
'Position',[60 80 50 20]);
val_a = uicontrol(fig,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',8,'FontName','Arial',...
'String','',...
'Position',[110 80 50 20],...
'CallBack','a=str2double(get(val_a,'String'))');
```

```
txt_val_b = uicontrol(gcf,...
'Style','text',...
'String','Lim. b:',...
'FontSize',8,'FontName','Arial',...
'Position',[60 55 50 20]);
```

```
val_b = uicontrol(fig,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',8,'FontName','Arial',...
'String',[],...
'Position',[110 55 50 20],...
'Callback','b=str2double(get(val_b,'String'))');

txt_val_c = uicontrol(gcf,...
'Style','text',...
'String','Grado :',...
'FontSize',8,'FontName','Arial',...
'Position',[60 30 50 20]);
val_c = uicontrol(fig,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',8,'FontName','Arial',...
'String',[],...
'Position',[110 30 50 20],...
'Callback','c=str2double(get(val_c,'String'))');

txt_val_d = uicontrol(gcf,...
'Style','text',...
'String','Lim. d:',...
'FontSize',8,'FontName','Arial',...
'Position',[160 80 50 20]);
val_d = uicontrol(fig,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',8,'FontName','Arial',...
'String',[],...
'Position',[210 80 50 20],...
'Callback','d=str2double(get(val_d,'String'))');

txt_val_e = uicontrol(gcf,...
'Style','text',...
'String','Lim. e:',...
'FontSize',8,'FontName','Arial',...
'Position',[160 55 50 20]);
val_e = uicontrol(fig,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',8,'FontName','Arial',...
'String',[],...
'Position',[210 55 50 20],...
'Callback','e=str2double(get(val_e,'String'))');

txt_val_f = uicontrol(gcf,...
'Style','text',...
'String','Grado :',...
'FontSize',8,'FontName','Arial',...
'Position',[160 30 50 20]);
val_f = uicontrol(fig,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',8,'FontName','Arial',...
'String',[],...
'Position',[210 30 50 20],...
'Callback','f=str2double(get(val_f,'String'))');
```

```
boton_calculo = uicontrol(gcf,...  
'Style','push',...  
'Position',[60 6 100 20],...  
'String','Calcular',...  
'Callback','curva_w2');
```

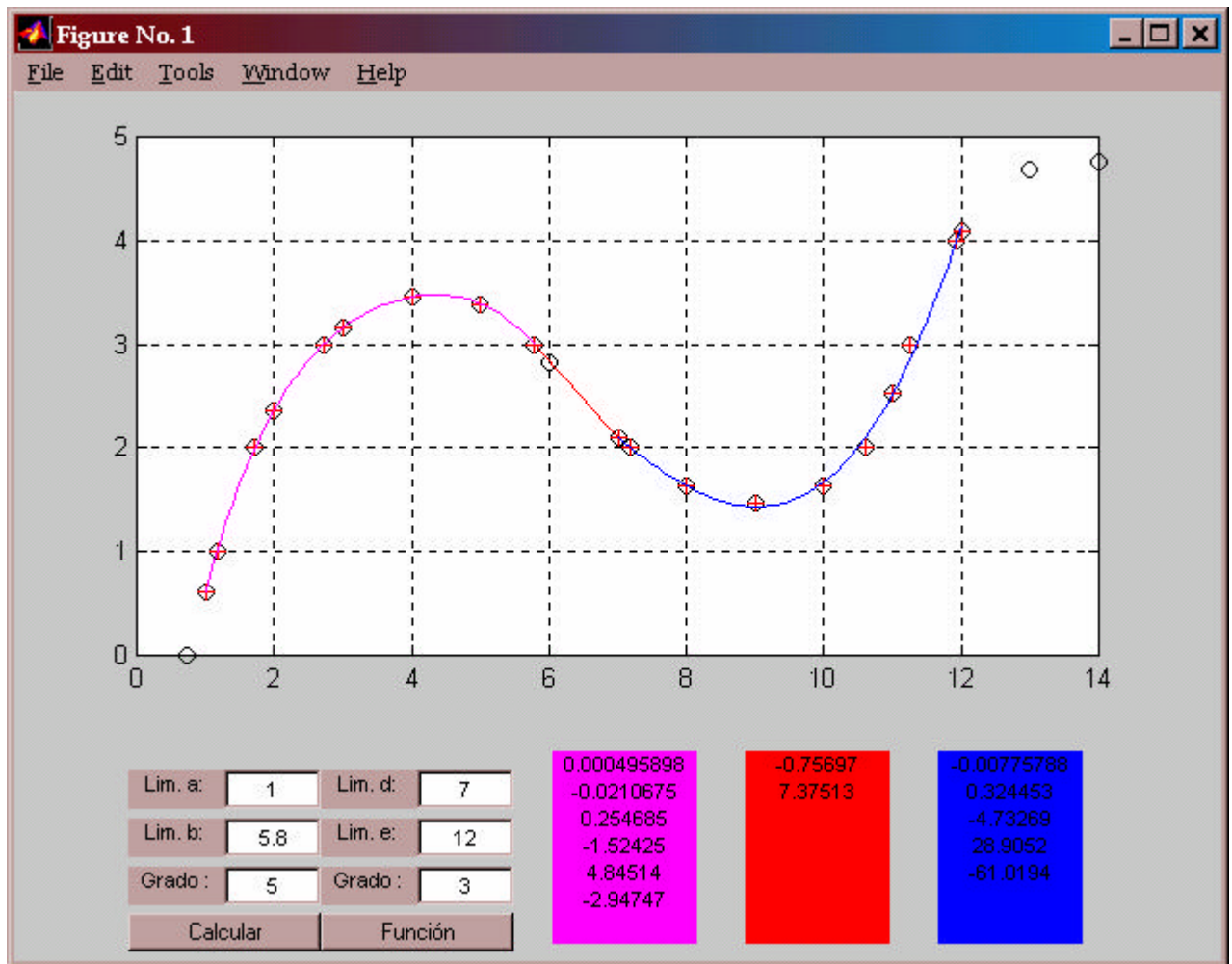
```
boton_funcion = uicontrol(gcf,...  
'Style','push',...  
'Position',[160 6 100 20],...  
'String','Función',...  
'Callback','curva_w3');
```

3) Construir un fichero curva\_w2.m

```
%Curva 1  
a1=1+sum(x>=0)-sum(x>=a);  
b1=sum(x<=b);  
x1=x([a1:b1]);  
y1=y([a1:b1]);  
g=polyfit(x1,y1,c)  
x2=[a:0.1:b];  
y2=polyval(g,x2);  
  
%Curva 2  
d1=1+sum(x>=0)-sum(x>=d);  
e1=sum(x<=e);  
w1=x([d1:e1]);  
z1=y([d1:e1]);  
h=polyfit(w1,z1,f)  
w2=[d:0.1:e];  
z2=polyval(h,w2);  
  
%Curva 3  
i=polyfit([b d],[polyval(g,b) polyval(h,d)],1);  
t2=[b:0.1:d];  
u2=polyval(i,t2);  
  
axes('position',[.1 .35 .8 .6])  
plot(x,y,'ko',x1,y1,'r+',x2,y2,'m',w1,z1,'r+',w2,z2,'b',t2,u2,'r')  
grid
```

3) Construir un fichero curva\_w3.m

```
txt_val_g = uicontrol(gcf,...  
'Style','text',...  
'BackgroundColor','m',...  
'String',[g],...  
'FontSize',8,'FontName','Arial',...  
'Position',[280 10 75 100]);  
  
txt_val_i = uicontrol(gcf,...  
'Style','text',...  
'BackgroundColor','r',...  
'String',[i],...  
'FontSize',8,'FontName','Arial',...  
'Position',[380 10 75 100]);  
  
txt_val_h = uicontrol(gcf,...  
'Style','text',...  
'BackgroundColor','b',...  
'String',[h],...  
'FontSize',8,'FontName','Arial',...  
'Position',[480 10 75 100]);
```



## Práctica - N° 7: Programación.

Resolver el siguiente problema de denominado como funciones económicas:

En un taller se fabrican piezas: A y B. Ambas están sujetas a tres procesos, Estampado, Soldado y Pintado.

**Las disponibilidades en minutos para cada proceso son:**

Estampado → 480 min.

Soldado → 420 min.

Pintado → 360 min.

**El tipo de pieza A consume:**

Estampado → 6 min.

Soldado → 12 min.

Pintado → 9 min.

**El tipo de pieza B consume:**

Estampado → 16 min.

Soldado → 6 min.

Pintado → 9 min.

**Ganancia \$4 la A y \$3 la B.**

**Evaluemos las restricciones:**

Estampado →  $6x_1 + 16x_2$  480

Soldado →  $12x_1 + 6x_2$  420

Pintado →  $9x_1 + 9x_2$  360

**1) Pasemos las restricciones a funciones con variable x1.**

F(Estampado) = [-6/16 480/16]

F(Soldado) = [-12/6 420/6]

F(Pintado) = [-9/9 360/9]

**2) Curva de ganancia can variable x1:**

$Z = 4x_1 + 3x_2$

**3) Construir un fichero economica\_01.m**

% Generación de las 3 curvas.

pol\_01=[-6/16 480/16]

pol\_02=[-12/6 420/6]

pol\_03=[-9/9 360/9]

x=[0:0.1:100];

y1=[polyval(pol\_01,x)];

y2=[polyval(pol\_02,x)];

y3=[polyval(pol\_03,x)];

xa=[0:0.1:100];

ya=[0:1:100];

p=0;

for a=0:0.1:100

ya1=[polyval(pol\_01,a)];

ya2=[polyval(pol\_02,a)];

ya3=[polyval(pol\_03,a)];

yamim=min([ya1 ya2 ya3]);

p=p+1;

ya(p)=yamim;

end

%Curva de ganancia

z=4.\*xa+3.\*ya;

zmax=max(z);

p=0;

for a=0:0.1:100

p=p+1;

if zmax==z(p)

xp=a

yp=zmax/3-4/3\*xp

end

end

axes('position',[.1 .35 .4 .6])

plot(x,y1,x,y2,x,y3,xp,yp,'ro')

XLim([0 100])

YLim([0 100])

GRID

axes('position',[.55 .35 .4 .6])

plot(xa,z,xp,zmax,'ro')

GRID

txt\_x = uicontrol(gcf,...

'Style','text',...

'String','Punto Optimo X:',...

'FontSize',8,'FontName','Arial',...

'Position',[60 80 100 15]);

txt\_vx = uicontrol(gcf,...

'Style','text',...

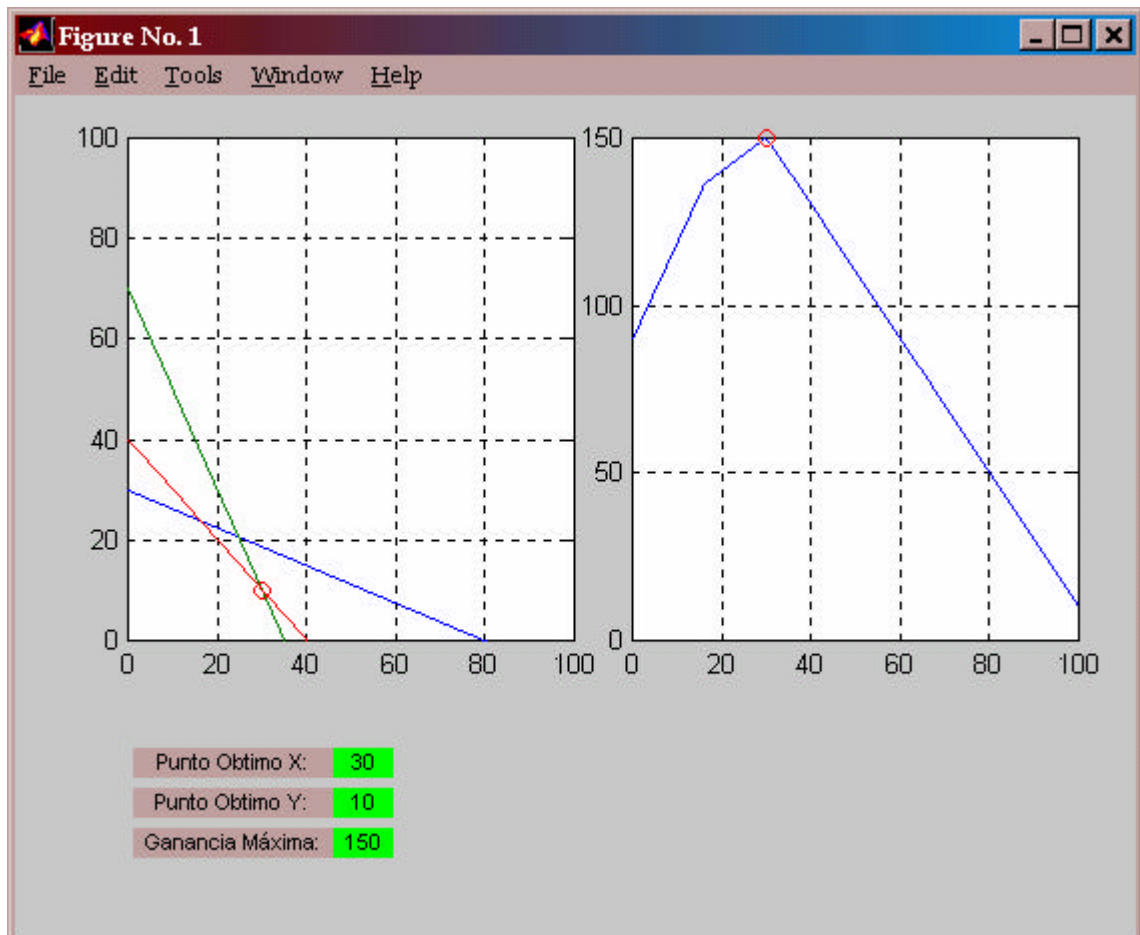
'String',[xp],...

'FontSize',8,'FontName','Arial',...

'BackgroundColor','g',...

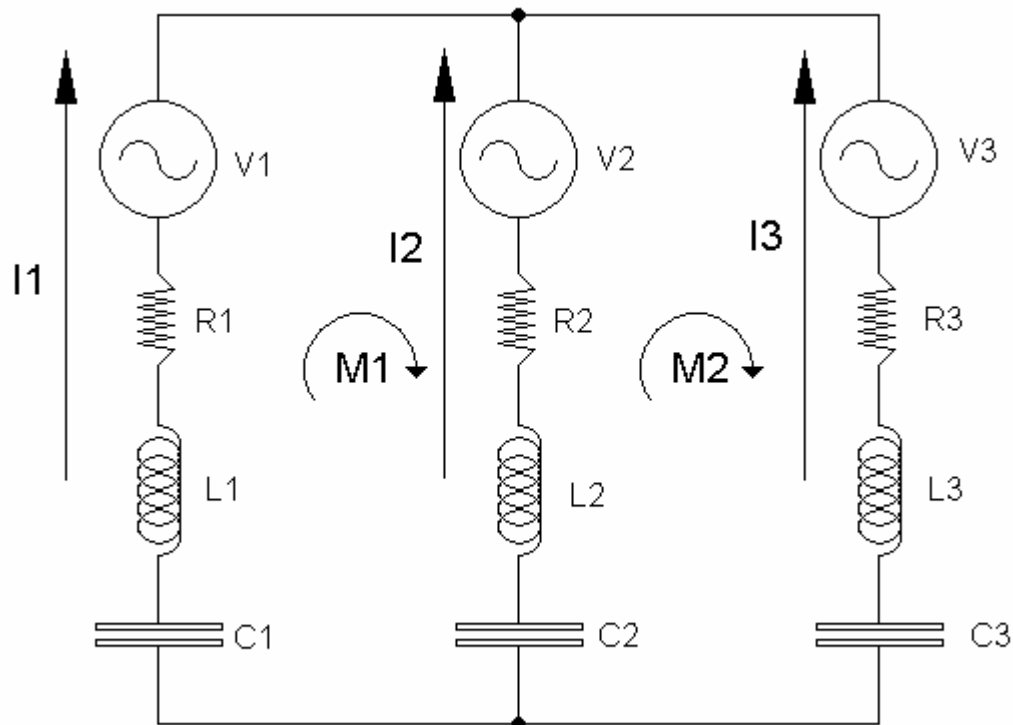
'Position',[160 80 30 15]);

```
txt_y = uicontrol(gcf,...  
'Style','text',...  
'String','Punto Optimo Y:',...  
'FontSize',8,'FontName','Arial',...  
'Position',[60 60 100 15]);  
txt_vy = uicontrol(gcf,...  
'Style','text',...  
'String',[yp],...  
'FontSize',8,'FontName','Arial',...  
'BackgroundColor','g',...  
'Position',[160 60 30 15]);  
  
txt_z = uicontrol(gcf,...  
'Style','text',...  
'String','Ganancia Máxima:',...  
'FontSize',8,'FontName','Arial',...  
'Position',[60 40 100 15]);  
txt_vz = uicontrol(gcf,...  
'Style','text',...  
'String',[zmax],...  
'FontSize',8,'FontName','Arial',...  
'BackgroundColor','g',...  
'Position',[160 40 30 15]);
```



## Práctica - N° 8: Programación.

Resolución de un sistema de 2 mallas, combinando el Matlab & Simulink.



Primer paso definimos la entrada de datos por el Matlab:

Generando un archivo m01.m y m02.m

Donde m01:

```
% Textos
txt_v = uicontrol(gcf,...
    'Style','text','String','V',...
    'Position',[100 375 50 22]);

txt_r = uicontrol(gcf,...
    'Style','text','String','R',...
    'Position',[160 375 50 22]);

txt_l = uicontrol(gcf,...
    'Style','text','String','L',...
    'Position',[220 375 50 22]);

txt_c = uicontrol(gcf,...
    'Style','text','String','C',...
    'Position',[280 375 50 22]);

txt_r1 = uicontrol(gcf,...
    'Style','text','String','R1',...
    'Position',[40 350 50 22]);

txt_r2 = uicontrol(gcf,...
    'Style','text','String','R2',...
    'Position',[40 325 50 22]);

txt_r3 = uicontrol(gcf,...
    'Style','text','String','R3',...
    'Position',[40 300 50 22]);

txt_f = uicontrol(gcf,...
    'Style','text','String','Fr',...
    'Position',[40 275 50 22]);
```

```
% Cajas editables
% Rama N°1
txt_v1 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[10],...
    'Position',[100 350 50 22],...
    'Callback','v1=str2double(get(txt_v1,'String'))');
txt_r1 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[100],...
    'Position',[160 350 50 22],...
    'Callback','r1=str2double(get(txt_r1,'String'))');
txt_l1 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[10],...
    'Position',[220 350 50 22],...
    'Callback','l1=str2double(get(txt_l1,'String'))');
txt_c1 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[100],...
    'Position',[280 350 50 22],...
    'Callback','c1=str2double(get(txt_c1,'String'))');

% Rama N°2
txt_v2 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[20],...
    'Position',[100 325 50 22],...
    'Callback','v2=str2double(get(txt_v2,'String'))');
txt_r2 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[200],...
    'Position',[160 325 50 22],...
    'Callback','r2=str2double(get(txt_r2,'String'))');
txt_l2 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[20],...
    'Position',[220 325 50 22],...
    'Callback','l2=str2double(get(txt_l2,'String'))');
txt_c2 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[200],...
    'Position',[280 325 50 22],...
    'Callback','c2=str2double(get(txt_c2,'String'))');
```



```
% Rama N°3
txt_v3 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[30],...
    'Position',[100 300 50 22],...
    'Callback','v3=str2double(get(txt_v3,'String'))');
txt_r3 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[300],...
    'Position',[160 300 50 22],...
    'Callback','r3=str2double(get(txt_r3,'String'))');
txt_l3 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[30],...
    'Position',[220 300 50 22],...
    'Callback','l3=str2double(get(txt_l3,'String'))');
txt_c3 = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[300],...
    'Position',[280 300 50 22],...
    'Callback','c3=str2double(get(txt_c3,'String'))');

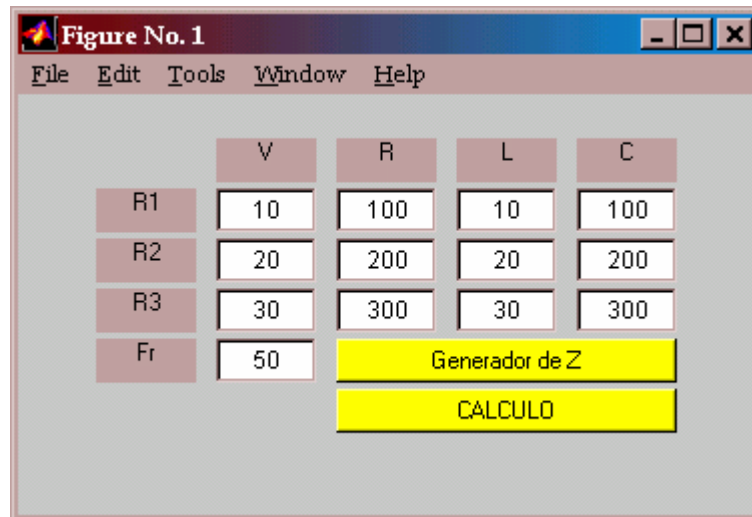
% Frecuencia
txt_f = uicontrol(gcf,...
    'Style','edit',...
    'BackgroundColor','White',...
    'FontSize',9,'FontName','Arial',...
    'String',[50],...
    'Position',[100 275 50 22],...
    'Callback','f=str2double(get(txt_f,'String'))');

% Boton MEMORIA
Boton_memoria = uicontrol(gcf,...
    'Style','push',...
    'BackgroundColor','y',...
    'String','Generador de Z',...
    'Position',[160 275 170 22],...
    'Callback','generador_de_z,m02');
```

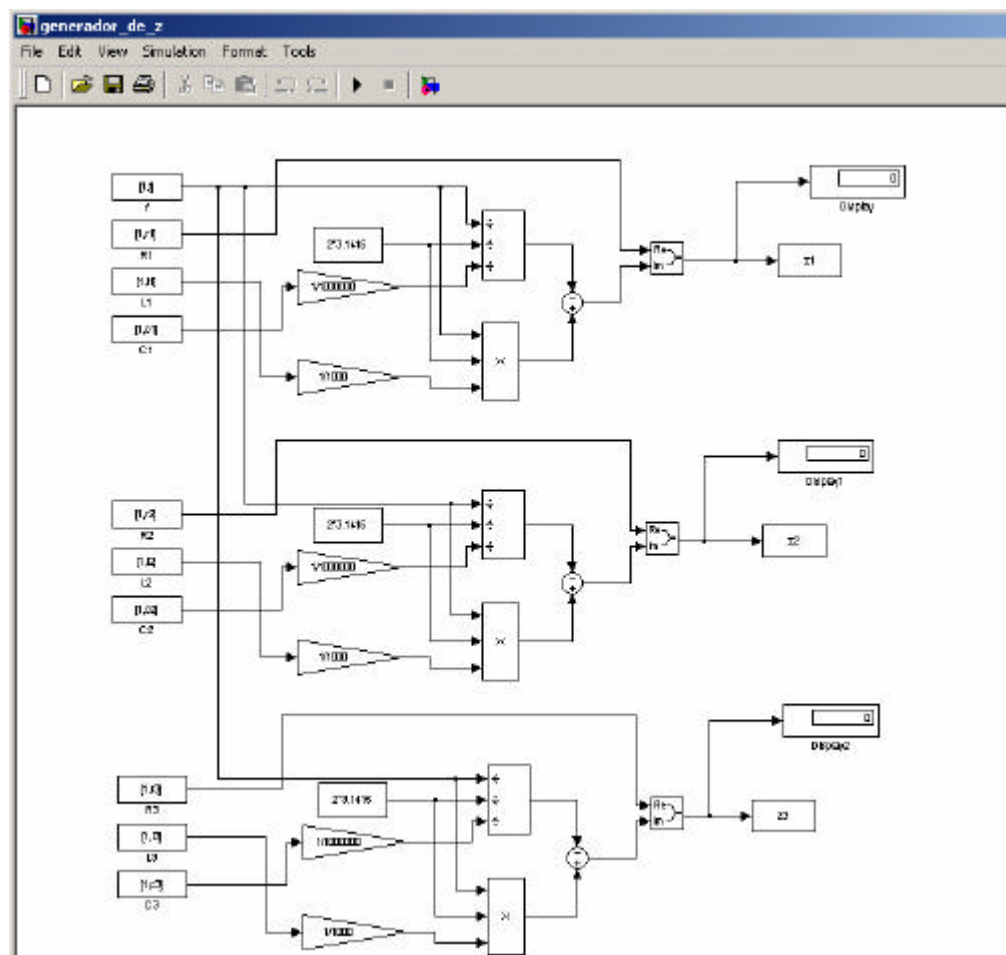
Donde m02:

```
% Boton MEMORIA
Boton_memoria = uicontrol(gcf,...
    'Style','push',...
    'BackgroundColor','y',...
    'String','CALCULO',...
    'Position',[160 250 170 22],...
    'Callback','valores_i,res_final');
```

Este será el resultado de los dos comandos:



Con Simulink generamos los valores de las impedancias Z1, Z2, Z3.  
Creamos un fichero generador\_de\_z.mdl

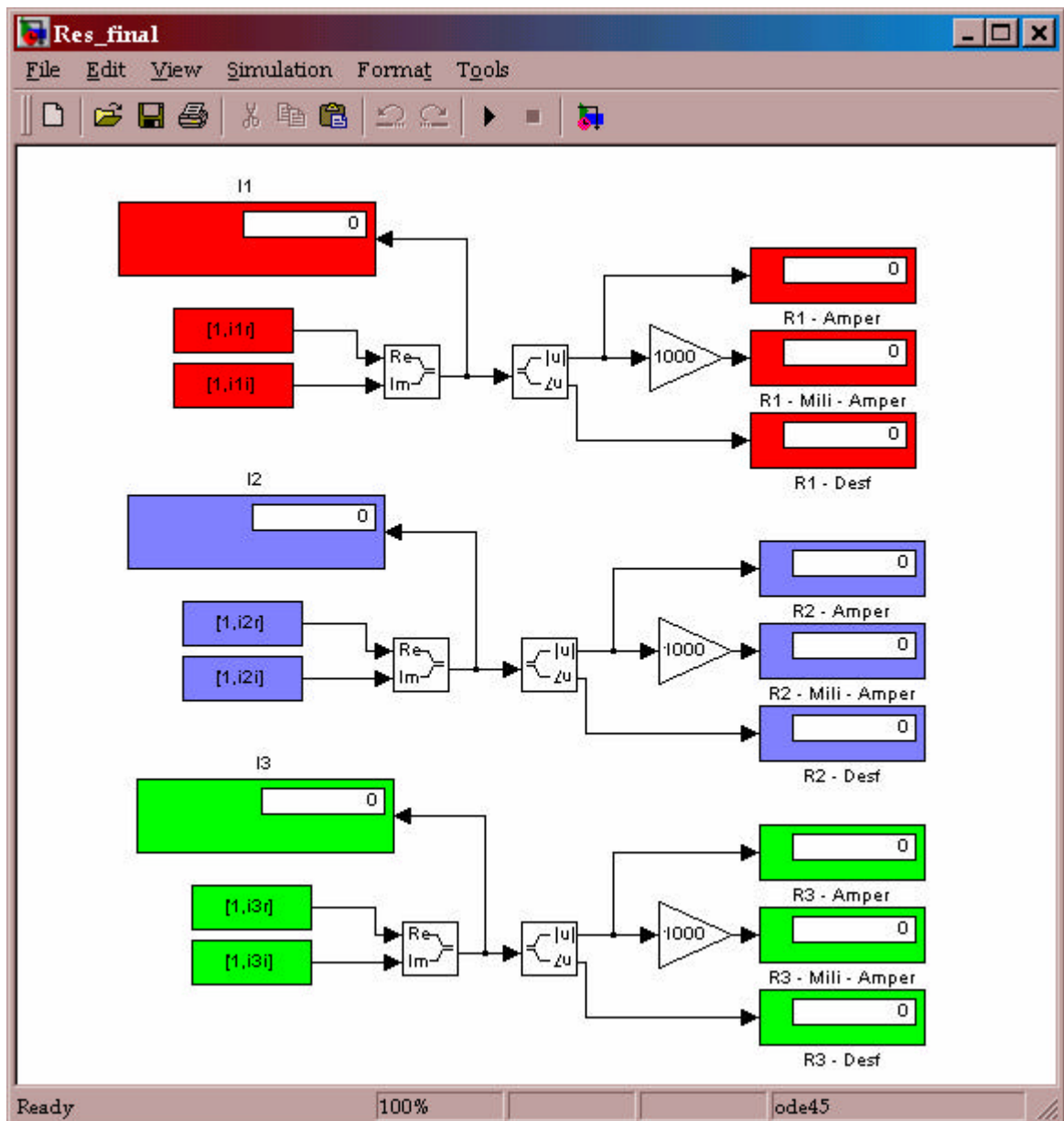


Luego de tener las Z generadas chequeamos sobre el botón CALCULO. Pero antes debemos cargar el fichero valores\_i.m que resuelve el sistema.

```
m=[z1 -z2 0; 0 z2 -z3; 1 1 1];
n=[-v1+v2;-v2+v3;0];
dm=det(m);
mx1=m;
mx1(1)=n(1);
mx1(2)=n(2);
mx1(3)=n(3);
dmx1=det(mx1);
mx2=m;
mx2(4)=n(1);
mx2(5)=n(2);
mx2(6)=n(3);
```

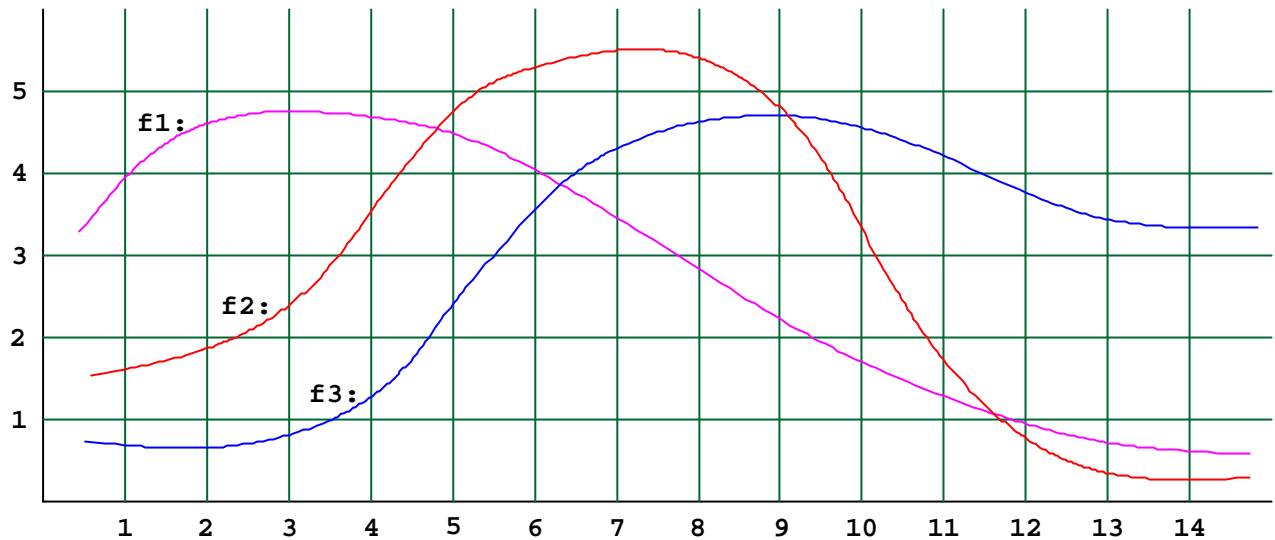
```
dmx2=det(mx2);
mx3=m;
mx3(7)=n(1);
mx3(8)=n(2);
mx3(9)=n(3);
dmx3=det(mx3);
i1=dmx1/dm;
i2=dmx2/dm;
i3=dmx3/dm;
i1r=real(i1);i1i=imag(i1);
i2r=real(i2);i2i=imag(i2);
i3r=real(i3);i3i=imag(i3);
```

Una vez finalizada la resolución de las I1, I2, I3. Pasamos al Simulink en donde creamos un fichero Res\_final para obtener la visualización de los módulos de las corrientes y sus desfases:



## Práctica - N° 9: Programación.

Dadas las siguientes curvas f1, f2 y f3 convertirlas en funciones aproximadas utilizando como herramienta la **práctica N° 6**, para llegar a dichas funciones. Hacer que se vea gráficamente los puntos elegidos x1, x2 y x3 sobre las curvas aproximadas, y obtener un resultado aplicando la función pedida.

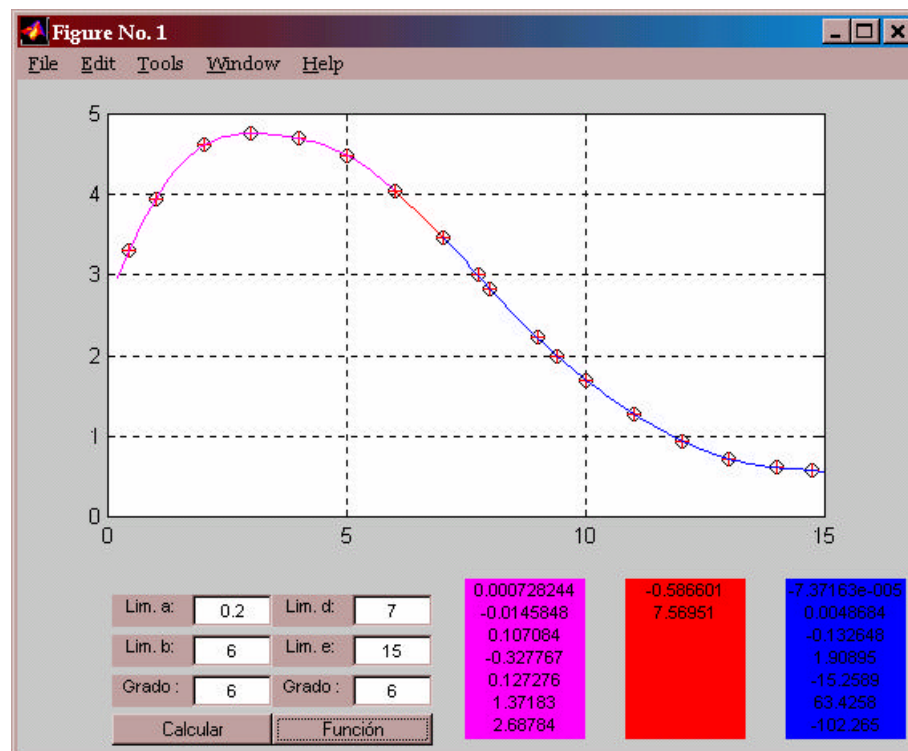


Calcular mediante una interfase gráfica la función:

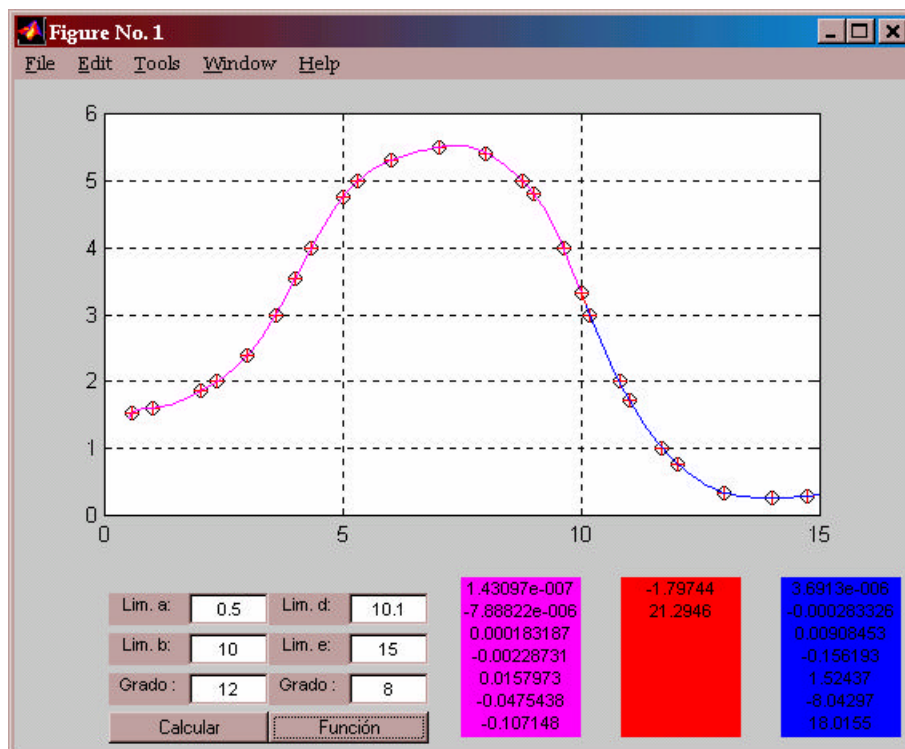
$$F_{(x1, x2, x3)} = \frac{F1_{(x1)}^2 + F2_{(x2)}^3 - \sqrt{F3_{(x3)}}}{F1_{(x1)} + F2_{(x2)} + F3_{(x3)}} \text{ y poder calcular en cualquier punto.}$$

Utilizando la herramienta que armamos en el práctico 6 sacamos las funciones que más se adecuan a la grafica de la tabla:

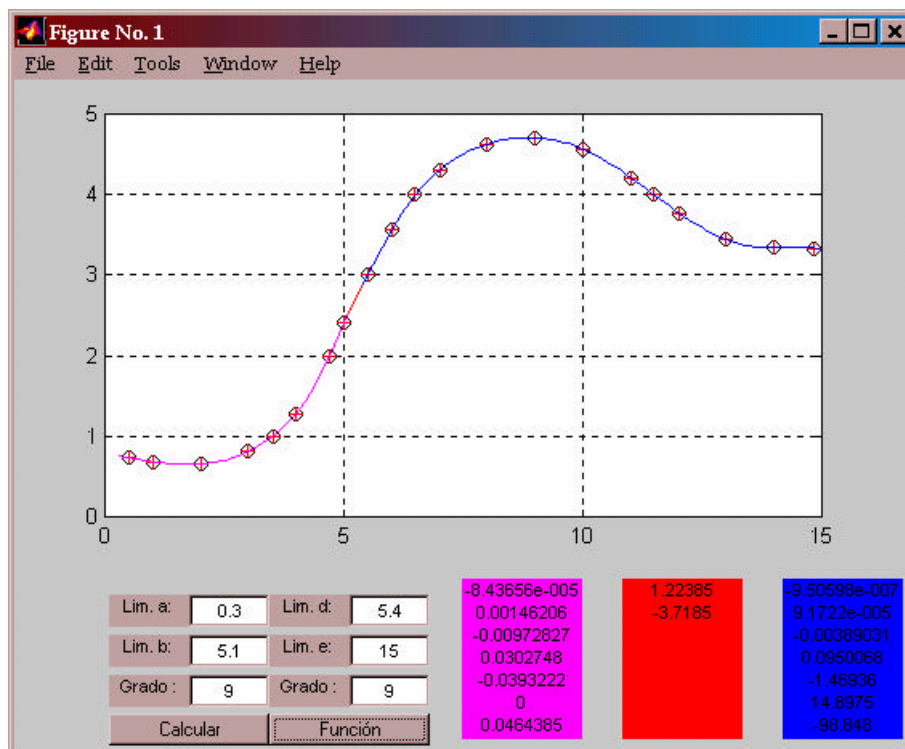
Función F1:



Función F2:



Función F3:



Para ingresar en el programa que se armó en la **práctica N° 6**, se necesitan estos puntos, entre más puntos mejor será el resultado final.

**Para F1:**

```
x=[0.4454 1.0000 2.0000 3.0000 4.0000 5.0000 6.0000 7.0000 7.7396 8.0000 9.0000  
9.3955 10.0000 11.0000 12.0000 13.0000 14.0000 14.7222]  
y=[3.2991 3.9524 4.6117 4.7597 4.6961 4.4870 4.0499 3.4638 3.0000 2.8354 2.2221  
2.0000 1.6971 1.2817 0.9460 0.7176 0.6134 0.5770]
```

**Para F2:**

```
x=[0.5799 1.0000 2.0000 2.3417 3.0000 3.6060 4.0000 4.3447 5.0000 5.2975 6.0000  
7.0000 8.0000 8.7660 9.0000 9.6201 10.0000 10.1736 10.7887 11.0000 11.6944 12.0000  
13.0000 14.0000 14.7390]  
y=[1.5348 1.6114 1.8634 2.0000 2.3958 3.0000 3.5346 4.0000 4.7453 5.0000 5.2988  
5.5056 5.4124 5.0000 4.7991 4.0000 3.3200 3.0000 2.0000 1.7187 1.0000 0.7701  
0.3448 0.2638 0.2913]
```

**Para F3:**

```
x=[0.4958 1.0000 2.0000 3.0000 3.5150 4.0000 4.7027 5.0000 5.4968 6.0000 6.4867  
7.0000 8.0000 9.0000 10.0000 11.0000 11.4771 12.0000 13.0000 14.0000 14.8399]  
y=[0.7282 0.6844 0.6572 0.8109 1.0000 1.2765 2.0000 2.4037 3.0000 3.5595 4.0000  
4.3101 4.6307 4.7108 4.5628 4.2115 4.0000 3.7687 3.4431 3.3399 3.3327]
```

Ahora, ya que tenemos los resultados armamos 3 archivos llamados Tabla\_01, Tabla\_02 y Tabla\_03. Extraer los datos de las funciones y guardarlos en:

**Tabla 01.m**

```
fun1_1=[7.282435128443753e-004 -1.458478043639861e-002 1.070842814144333e-001  
-3.277673152739540e-001 1.272757981926073e-001 1.371828443342523e+000  
2.687835329247940e+000];  
fun1_2=[-5.866009195766742e-001 7.569505517460017e+000];  
fun1_3=[-7.371630854963460e-005 4.868400104707042e-003 -1.326476100741741e-001  
1.908948154247105e+000 -1.525887569020492e+001 6.342580237474054e+001  
-1.022652644166399e+002];  
  
fun2_1=[1.430971109567314e-007 -7.888221515354604e-006 1.831869460477721e-004  
-2.287311675139206e-003 1.579732558429807e-002 -4.754377503087397e-002  
-1.071482628479750e-001 1.542297362989769e+000 -6.178620007305208e+000  
1.299102589430986e+001 -1.499127909561578e+001 8.911864245538343e+000  
-5.227668579464684e-001];  
fun2_2=[-1.794577919733014e+000 2.126589834690893e+001];  
fun2_3=[3.183604795947678e-004 -2.402402972831082e-002 7.509810495570706e-001  
-1.247670612456362e+001 1.166900688710378e+002 -5.865685185517266e+002  
1.250956258599278e+003];  
  
fun3_1=[-8.436555816443513e-005 1.462063135633769e-003 -9.728267860166128e-003  
3.027479423280630e-002 -3.932219561774492e-002 0 4.643849447080897e-002 0  
-1.348187149358504e-001 7.901781921326763e-001];  
fun3_2=[1.223854536708892e+000 -3.718501998938394e+000];  
fun3_3=[-9.505983988470844e-007 9.172204344213016e-005 -3.890313210136284e-003  
9.500676366668570e-002 -1.469359234158917e+000 1.489748125876928e+001  
-9.884795865637356e+001 4.130712957332726e+002 -9.833531422642176e+002  
1.014799355059362e+003];  
  
txt_val_x1 = uicontrol(gcf,...  
'Style','text',...  
'String','X1:',...  
'FontSize',8,'FontName','Arial',...  
'Position',[60 280 50 20]);
```

```

val_x1 = uicontrol(gcf,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',8,'FontName','Arial',...
'String',[],...
'Position',[110 280 50 20],...
'Max',100,...
'Callback','x1=str2double(get(val_x1,'String'))');

txt_val_x2 = uicontrol(gcf,...
'Style','text',...
'String','X2:',...
'FontSize',8,'FontName','Arial',...
'Position',[60 255 50 20]);
val_x2 = uicontrol(gcf,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',8,'FontName','Arial',...
'String',[],...
'Position',[110 255 50 20],...
'Max',100,...
'Callback','x2=str2double(get(val_x2,'String'))');

txt_val_x3 = uicontrol(gcf,...
'Style','text',...
'String','X3:',...
'FontSize',8,'FontName','Arial',...
'Position',[60 230 50 20]);
val_x3 = uicontrol(gcf,...
'Style','edit',...
'BackgroundColor','white',...
'FontSize',8,'FontName','Arial',...
'String',[],...
'Position',[110 230 50 20],...
'Max',100,...
'Callback','x3=str2double(get(val_x3,'String'))');

boton_buscar = uicontrol(gcf,...
'Style','push',...
'Position',[60 205 100 20],...
'String','Buscar',...
'Callback','tabla_02');

```

**Tabla 02:**

```

if      1 <= x1 & x1 <= 6,rx1=polyval(fun1_1,x1)
elseif 6 <  x1 & x1 <= 7,rx1=polyval(fun1_2,x1)
elseif 7 <  x1 & x1 <= 14,rx1=polyval(fun1_3,x1)
else    rx1='Error: 1<=x1<=14'
end

if      1 <= x2 & x2 <= 10 ,rx2=polyval(fun2_1,x2)
elseif 10 <  x2 & x2 <= 10.1 ,rx2=polyval(fun2_2,x2)
elseif 10.1 <  x2 & x2 <= 14 ,rx2=polyval(fun2_3,x2)
else    rx2='Error: 1<=x2<=14'
end

if      1 <= x3 & x3 <= 5.1 ,rx3=polyval(fun3_1,x3)
elseif 5.1 <  x3 & x3 <= 5.4 ,rx3=polyval(fun3_2,x3)
elseif 5.4 <  x3 & x3 <= 14 ,rx3=polyval(fun3_3,x3)
else    rx3='Error: 1<=x3<=14'
end

```

```

txt_val_x1 = uicontrol(gcf,...
'Style','text',...
'BackgroundColor','y',...
'String',[rx1],...
'FontSize',10,'FontName','Arial',...
'Position',[160 280 200 20]);

txt_val_x2 = uicontrol(gcf,...
'Style','text',...
'BackgroundColor','y',...
'String',[rx2],...
'FontSize',10,'FontName','Arial',...
'Position',[160 255 200 20]);

txt_val_x3 = uicontrol(gcf,...
'Style','text',...
'BackgroundColor','y',...
'String',[rx3],...
'FontSize',10,'FontName','Arial',...
'Position',[160 230 200 20]);

boton_calculo = uicontrol(gcf,...
'Style','push',...
'Position',[60 180 100 20],...
'String','Calcular',...
'CallBack','tabla_03');

```

**Tabla 03:**

```

if 1<=x1 & x1<=14 & 1<=x2 & x2<=14 & 1<=x3 & x3<=14 ,rfinal=(rx1^2+rx2^3-
(rx3)^(1/2))/(rx1+rx2+rx3)
else rfinal='Error'
end

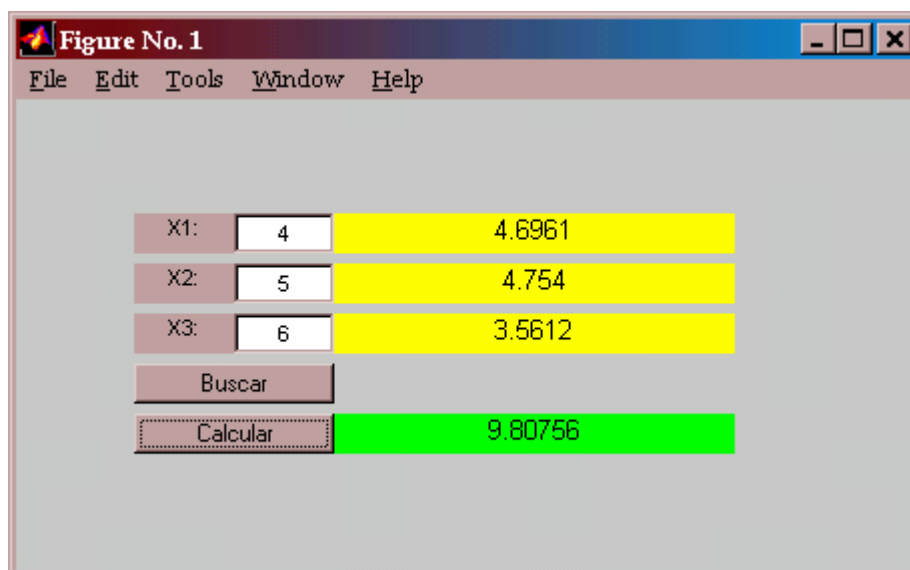
```

```

txt_val_final = uicontrol(gcf,...
'Style','text',...
'BackgroundColor','g',...
'String',[rfinal],...
'FontSize',10,'FontName','Arial',...
'Position',[160 180 200 20]);

```

El resultado final quedaría así:





## Prácticas de SIMULINK:

### Práctica - N° 1: Cal cul o de desfase.

Comenzaremos con un calculo sencillo el que consiste en averiguar el modulo y el ángulo de desfase de un sistema eléctrico con una impedancia compuesta por RLC.

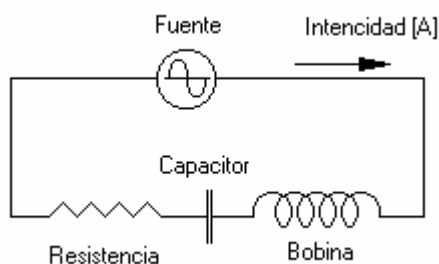
Datos:

Fuente = 20 V / 50 Hz / 0 Desf.

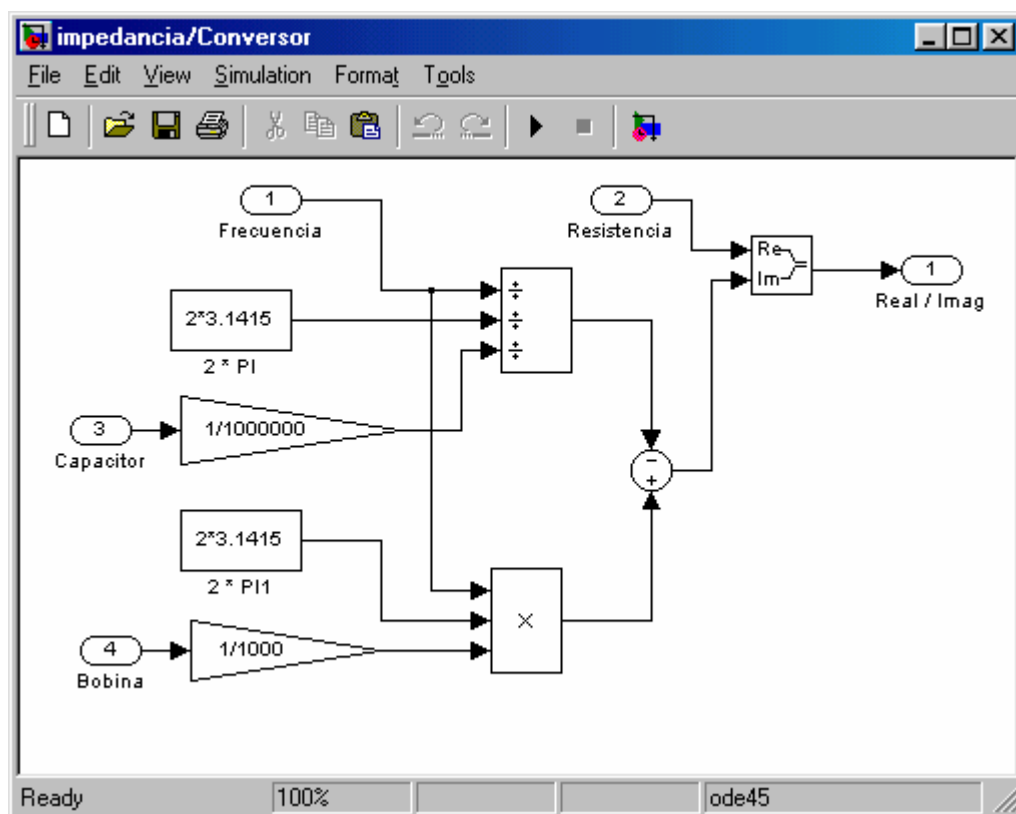
Resistencia = 100

Capacitor = 1 $\mu$ F

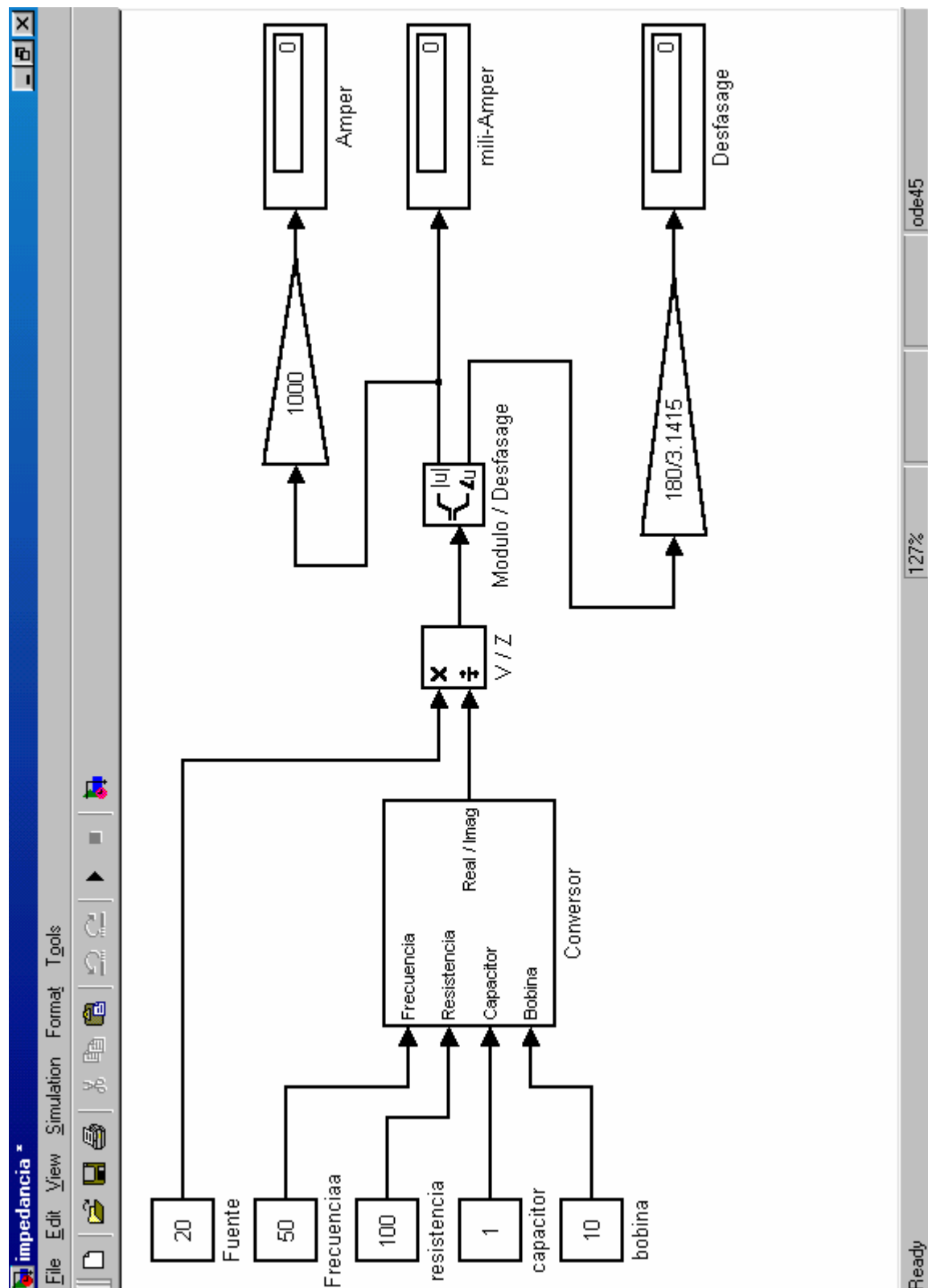
Bobina = 10mHe



El primer paso será crear un conversor para pasar de una señal normal a una compleja:

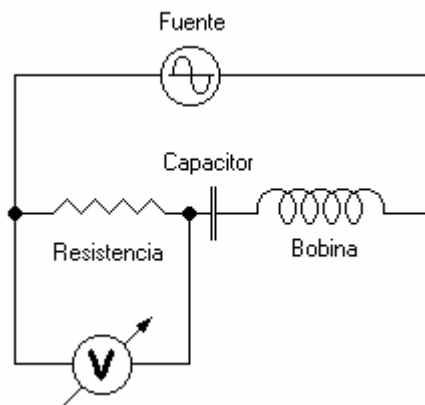


Segundo paso Utilizaremos el conversor:

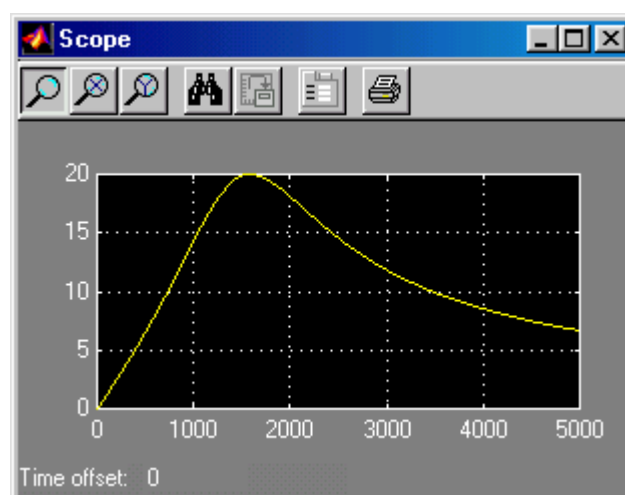
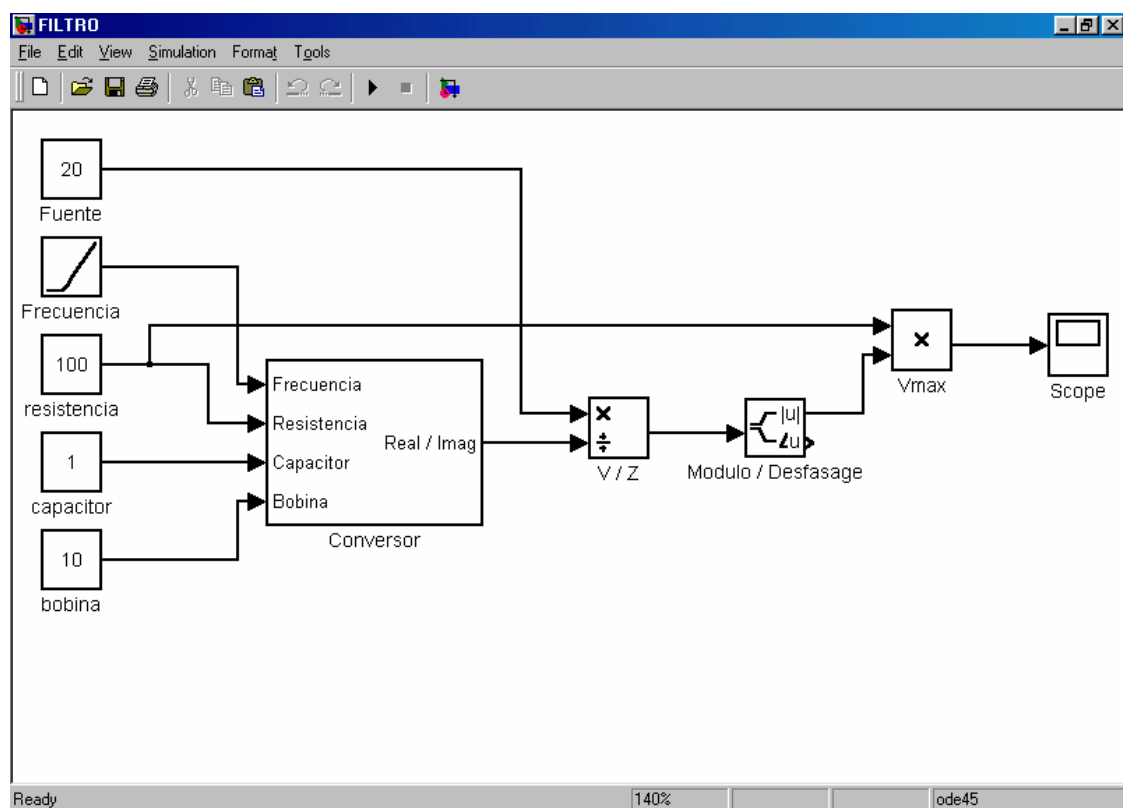


## Práctica - N° 2: Máxima resistencia.

Utilizando el sistema anterior encontraremos la frecuencia donde el sistema siguiente actúa con su máximo potencial.



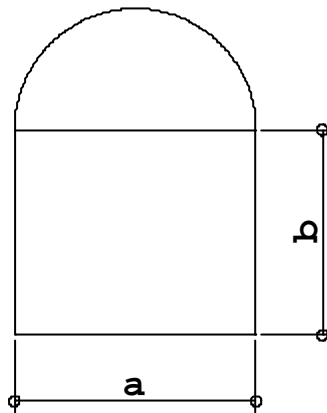
Fuente = 20 V / 0-5000 Hz / 0 Desf.  
 Resistencia = 100  
 Capacitor = 1 F  
 Bobina = 10mHe



## Práctica - N° 3: Maximización de área.

Debemos construir una ventana con la forma que se ve en la figura se dispone de 3m continuo de marco y se desea tener el área máxima de la ventana para que entre la mayo cantidad de luz.

Se necesita maximizar una determinada área, utilizaremos el Simulink para determinar la forma y el tamaño de dicha ventana.



$$P = 3\text{m.}$$

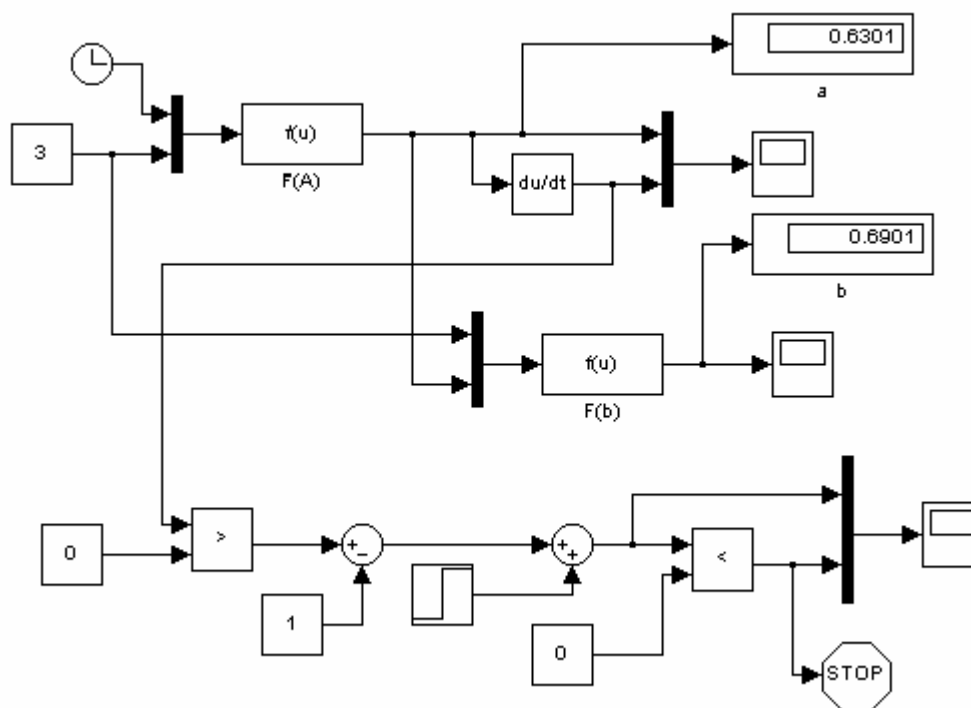
$$A_{\max} = ?$$

$$A = a \cdot b + \frac{\delta \cdot a^2}{8}$$

$$P = a + 2b + \frac{a \cdot \delta}{2} \rightarrow b = \frac{P}{2} - \frac{a \cdot \delta}{4} - \frac{a}{2}$$

$$F_{(A)} = a \cdot \left[ \frac{P}{2} - \frac{a \cdot \delta}{4} - \frac{a}{2} \right] + \frac{\delta \cdot a^2}{8}$$

El sistema que resuelve el problema seria el siguiente:



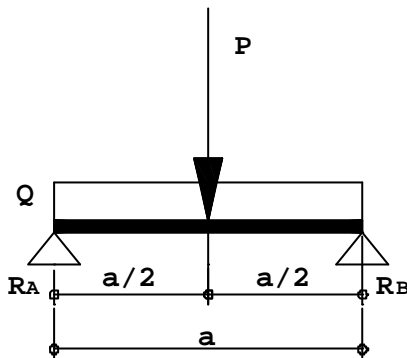
Donde:

$$F(A): u[1] \cdot (u[2]/2 - u[1] \cdot 3.141592654/4 - u[1]/2) + (3.141592654 \cdot u[1]^2)/8$$

$$F(B): u[1]/2 - u[2] \cdot 3.141592654/4 - u[2]/2$$

## Práctica - N° 4: Diagrama de M y Q (Sistema simétrico).

Supongamos tener el siguiente caso en donde se desea calcular el momento y el corte de una viga simplemente apoyada.



Debemos tener en cuenta 2 importantes datos:

1° Momento = Fuerza  $\times$  distancia.

2° Una carga repartida uniformemente puede remplazarse por una puntual aplicada en su centro.

La resolución se llega mediante 3 funciones:

1° resuelve la resultante  $R_A$  y  $R_B$  con la particularidad que en este ejemplo al ser simétricas son iguales, (no siempre se da el caso).

$$R_A = R_B = \frac{P + (Q \times a)}{2}$$

2° obtener el momento antes de el cambio que produce la carga puntual  $P$ .

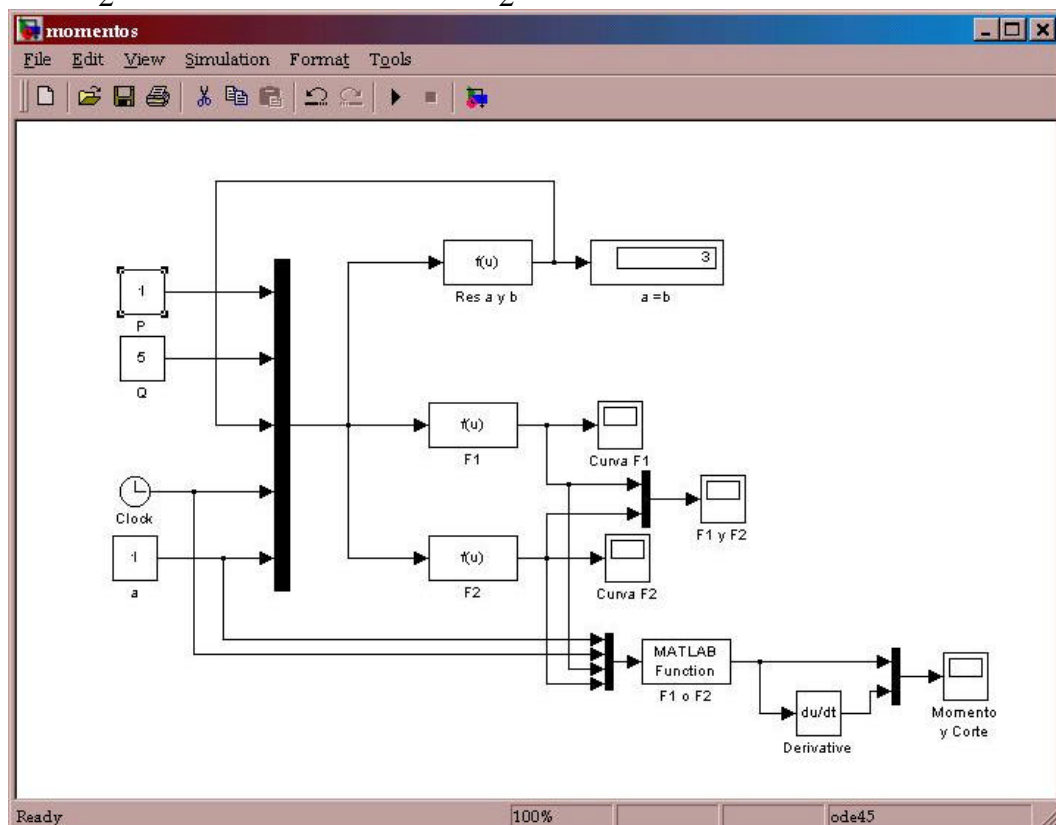
Tener en cuenta el sentido de los momentos y su signo:

$$M_1 = -(Q \times X) \times \frac{X}{2} + R_A \times X = -\frac{(Q \times X^2)}{2} + R_A \times X$$



3° obtener el momento después de el cambio que produce la carga puntual  $P$ .

$$M_2 = -(Q \times X) \times \frac{X}{2} - P \times (X - a) + R_A \times X = -\frac{(Q \times X^2)}{2} - P \times (X - a) + R_A \times X$$



Donde:

Res a y b:  $(u[5]*u[2]+u[1])/2$

F(1):  $(u[2]*u[4]^2)/2-u[3]*u[4]$

F(2):  $-(u[3]*u[4])+((u[2]*u[4]^2)/2)+(u[1]*(u[4]-u[5]/2))$

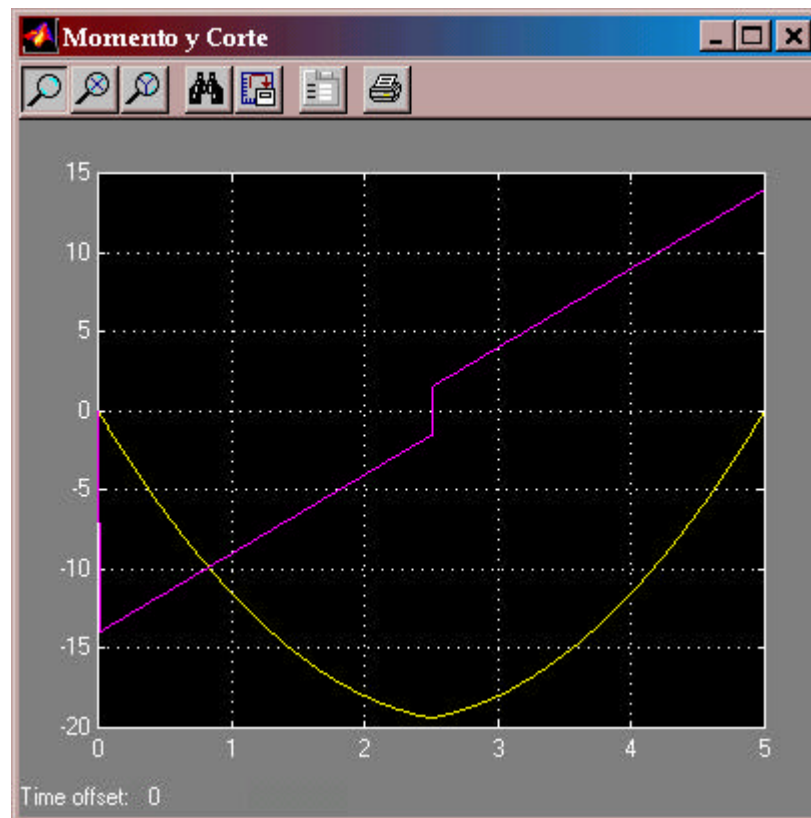
Crear un archivo momentos1.m

F1 o F2: `function[x] = variable(u)`

→ MATLAB  
Function `if (u(1)/2)>u(2),x=u(3)`  
F1 o F2 `else x=u(4)`  
`end`

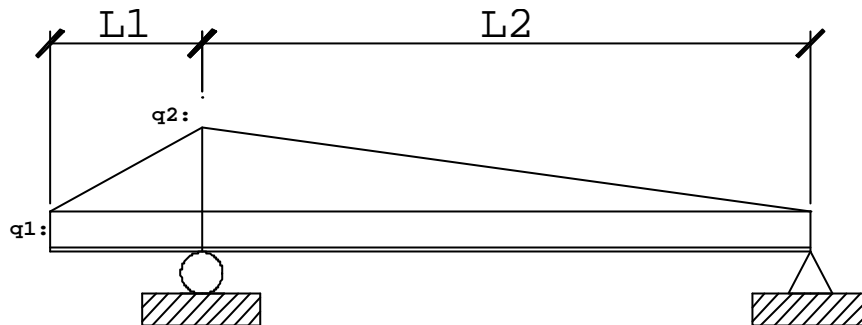
Ejemplo: P=3, Q=5, Clock de 0 a 5, a=5.

El resultado final seria como se ve aquí:



## Práctica - N° 5: Diagrama de M y Q (Sistema asimétrico).

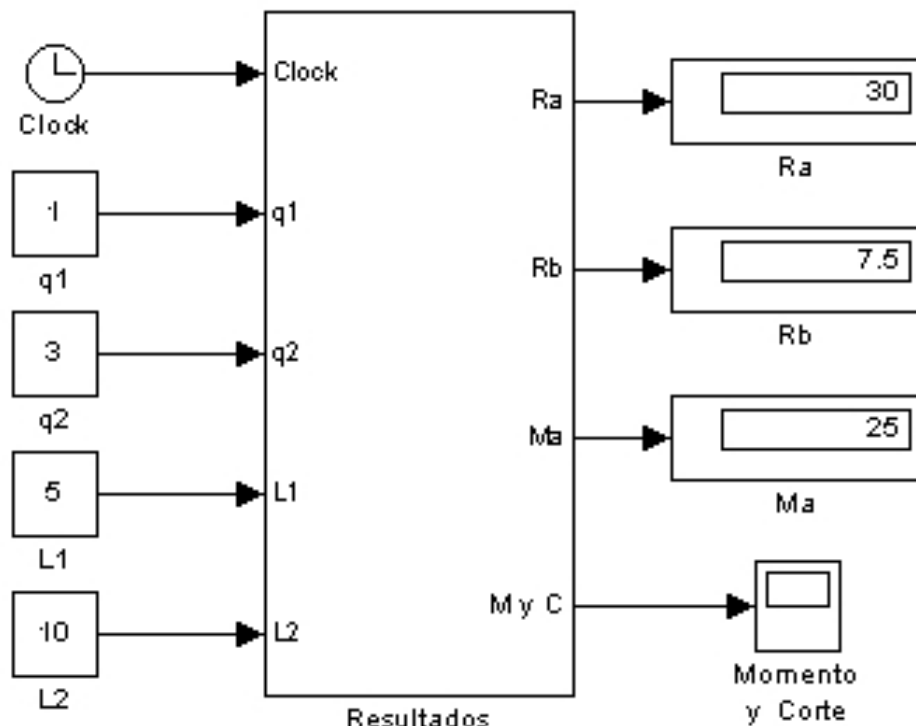
Utilizando como salida el SCOPE de Simulink obtener las gráficas del Momento y Corte de la viga simple teniendo como variable:  $q1$ ,  $q2$ ,  $L1$  y  $L2$ .



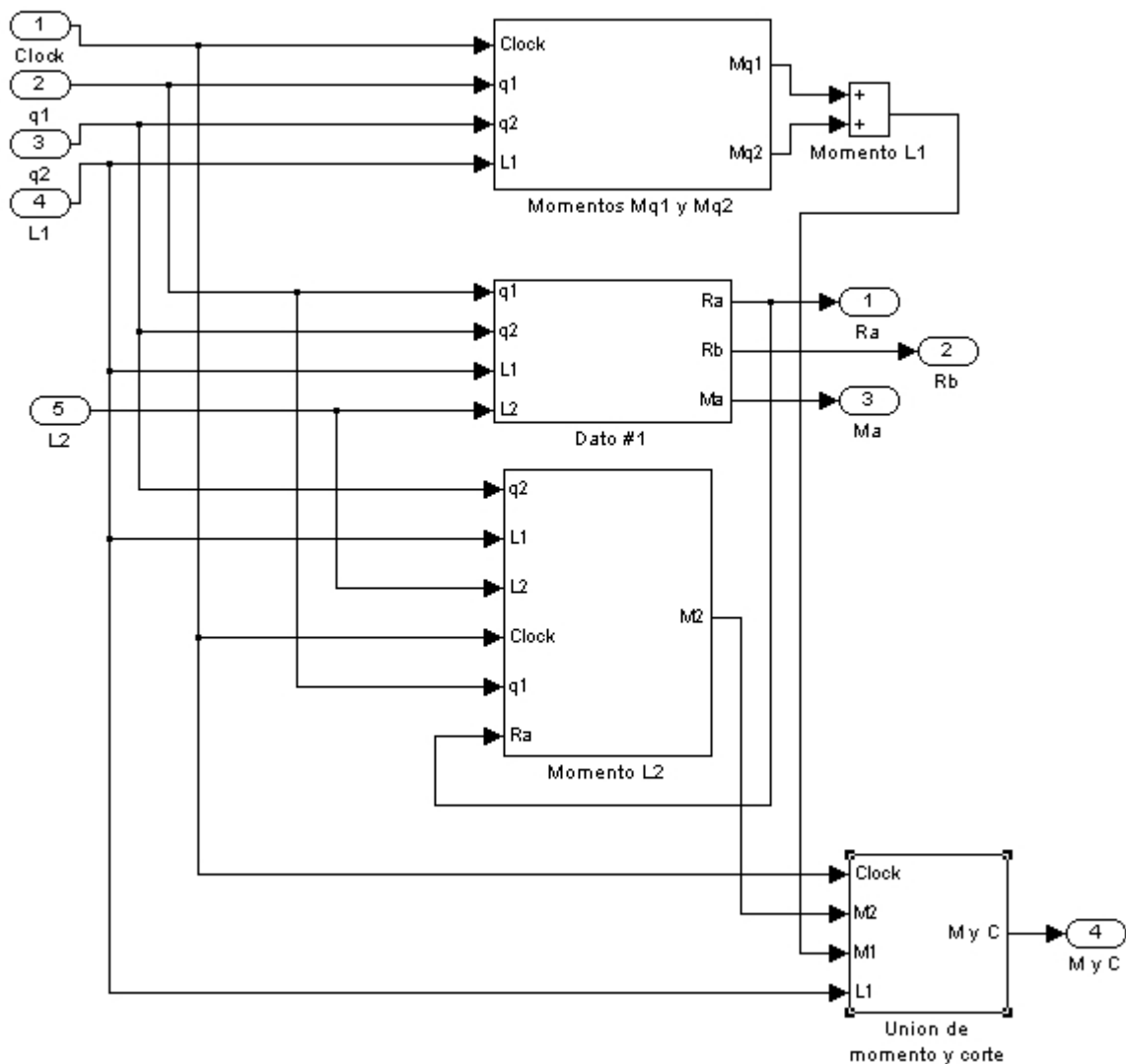
Para resolver el siguiente problema que es una simple suma de momentos variando las distancias debemos tener en cuenta 3 datos importantes.

- 1) Momento = Fuerza \* Distancia
- 2) La carga uniforme se representa como una puntual a la mitad de la misma.
- 3) La carga triangular se representa como una puntual a  $1/3$  de la parte cargada superior o sea a  $2/3$  de la que se hace cero "0".

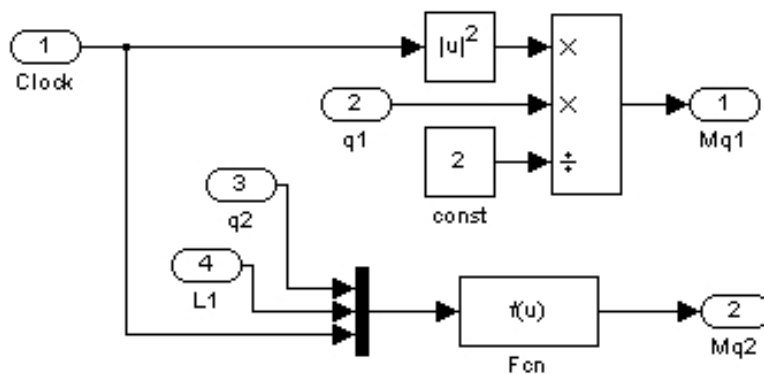
Generamos un bloque que reúne dentro, los distintos sub-bloques, que resuelven el sistema.



Dentro de **Resultados** tenemos este sistema formado por varios bloques:



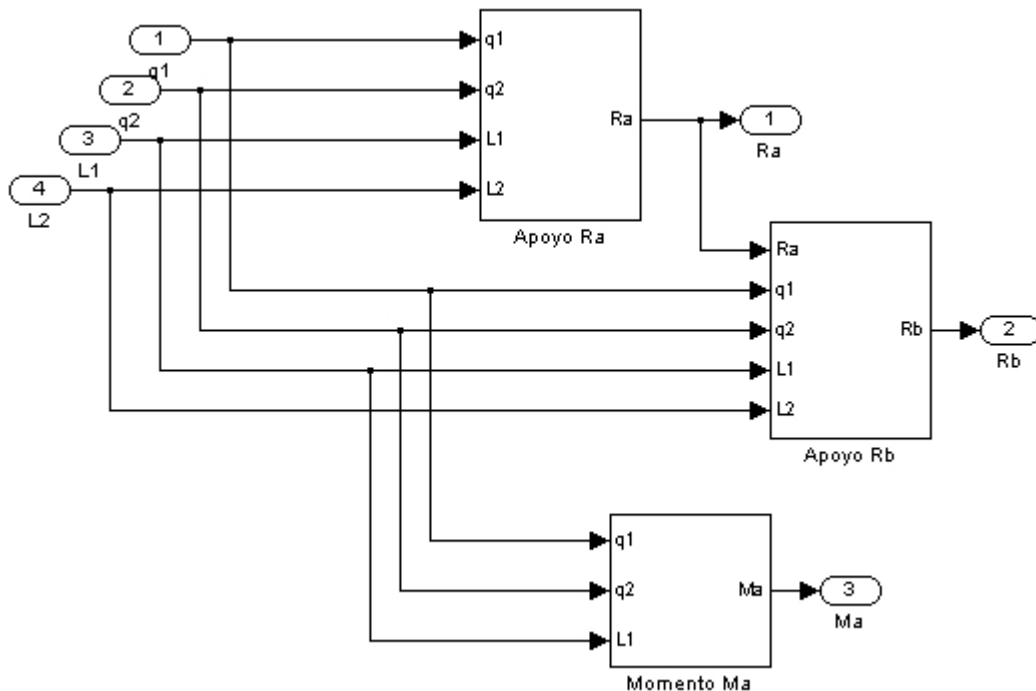
Analizaremos el cálculo del momento en la parte de la mensual calculamos individualmente el momento de la carga uniforme y de la carga triangular, y por ultimo sumamos ambos valores (Bloque **Momentos Mq1 y Mq2**)



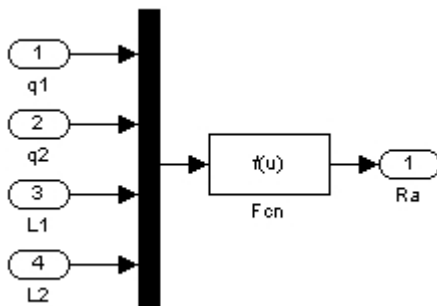
Dentro de Fcn:  $(u[1]/(6*u[2]))*u[3]^3$



Dentro de **Datos #1** tenemos varios sub-bloques en los cuales analizaremos 3 importantes resultados los dos resultantes en los apoyos Ra, Rb y el momento en el apoyo A llamado Ma:

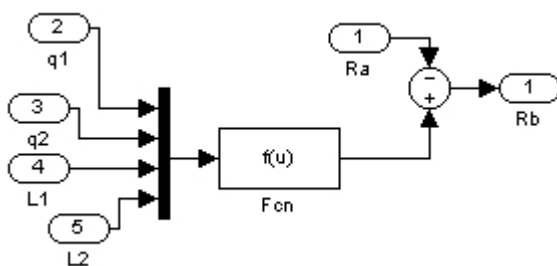


Dentro de **Apoyo Ra**:



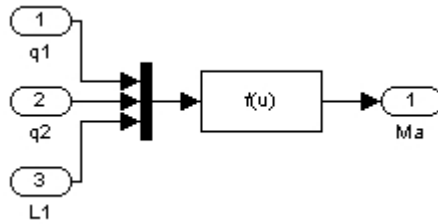
Dentro de Fcn:  $(u[1]*u[3]*(u[3]/2+u[4])+u[2]*u[3]/2*(u[3]/3+u[4])+u[2]*u[4]/2*(2/3*u[4])+u[1]*(u[4]^2)/2)/u[4]$

Dentro de **Apoyo Rb**:



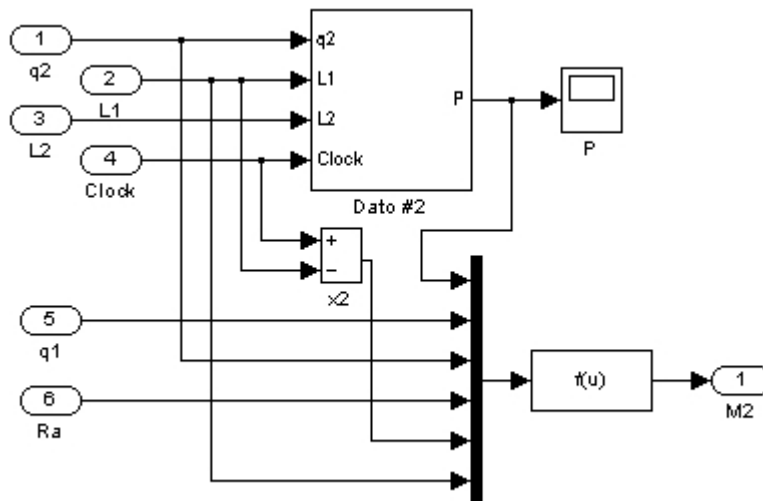
Dentro de Fcn:  $u[1]*(u[3]+u[4])+(u[2]*(u[3]+u[4])/2)$

Dentro de **Momento Ma** en Apoyo A:



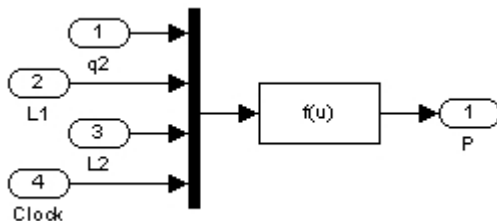
Dentro de Fcn:  $(u[2]*u[3]^2)/6+(u[1]*u[3]^2)/2$

Dentro de **Momento L2**:



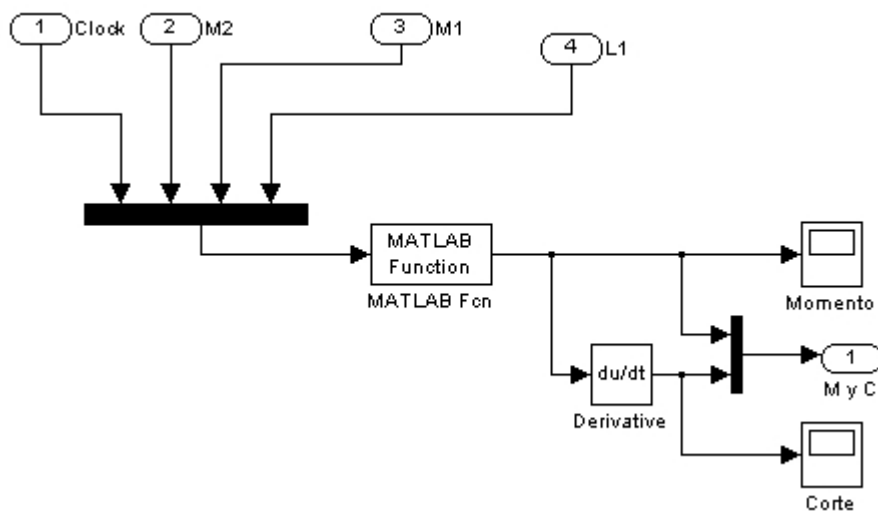
Dentro de Fcn:  $-(-u[3]*u[6]/2*(u[6]/3+u[5])-u[2]*u[6]*(u[6]/2+u[5])$   
 $-(u[1]*u[5]^2)/3-((u[3]-u[1]+u[2])*u[5]^2)/2+u[4]*u[5])$

Dentro de **Dato #2**, sacamos el valor punto a punto de la carga triangular para poder saber la carga distribuida triangular:

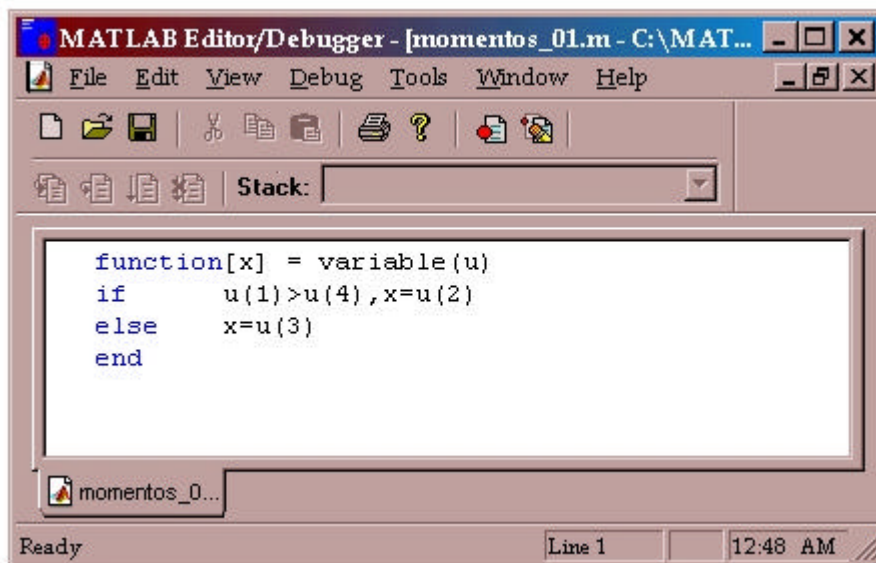


Dentro de Fcn:  $(u[1]/u[3])*(u[4]-u[2])$

Dentro del bloque **Unión de momento y corte**.



Dentro del MATLAB Fcn:



Ejemplo de un caso particular común:

Q1 = 1 [t/m]

Q2 = 3 [t/m]

L1 = 5 [m]

L2 = 10 [m]

