

Grupo 3

Participantes:

David Arias Calderón 20181020149

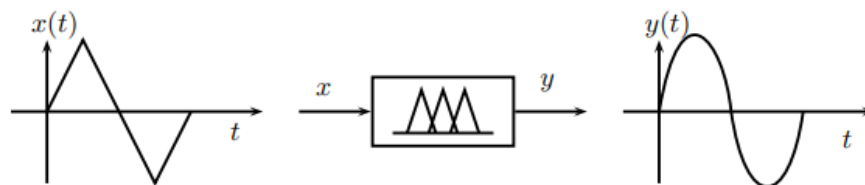
Luis Miguel Polo 20182020158

Taller 3 Ejercicio 2

Enunciado

Realizar la sintonía de un sistema difuso tipo Mamdani que permita convertir una señal triangular a seno como se muestra en la figura 1.

Figura 1



Configuraciones

- Funciones de pertenencia del antecedente: 7
- Funciones de pertenencia del consecuente: Libre
- Tipo de las funciones de pertenencia: Libre (sugeridas gaussianas)

Requerimientos de diseño

- Error máximo del 5%
- Error cuadrático medio inferior al 2%
- Considerando el valor máximo de la señal

Solución

Configuración inicial de sistema difuso

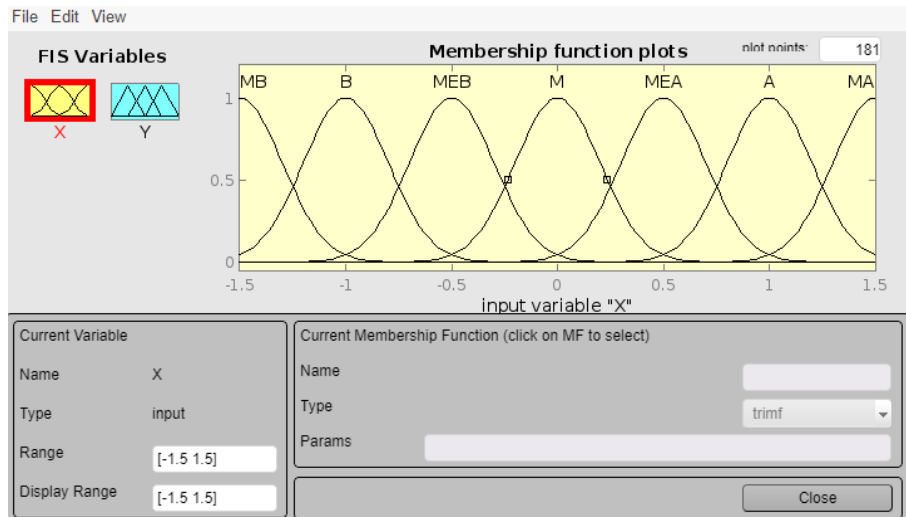
```
%Aplicación optimización de un sistema difuso que convierte una señal triangular a seno

close all
clear all
warning off

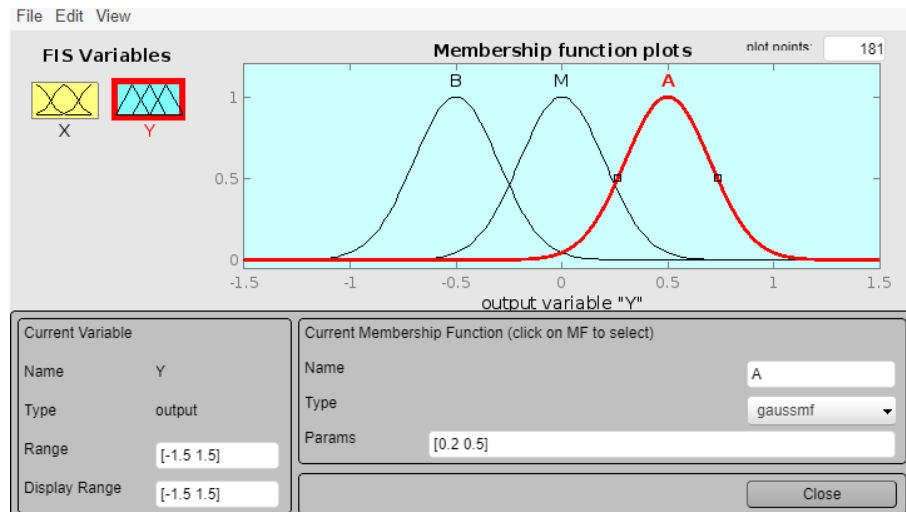
%Condición inicial

x=[0.2 -1.5 0.2 -1 0.2 -0.5 0.2 0 0.2 0.5 0.2 1 0.2 1.5 0.2 -0.5 0.2 0 0.2 0.5];
```

Entradas



Salidas



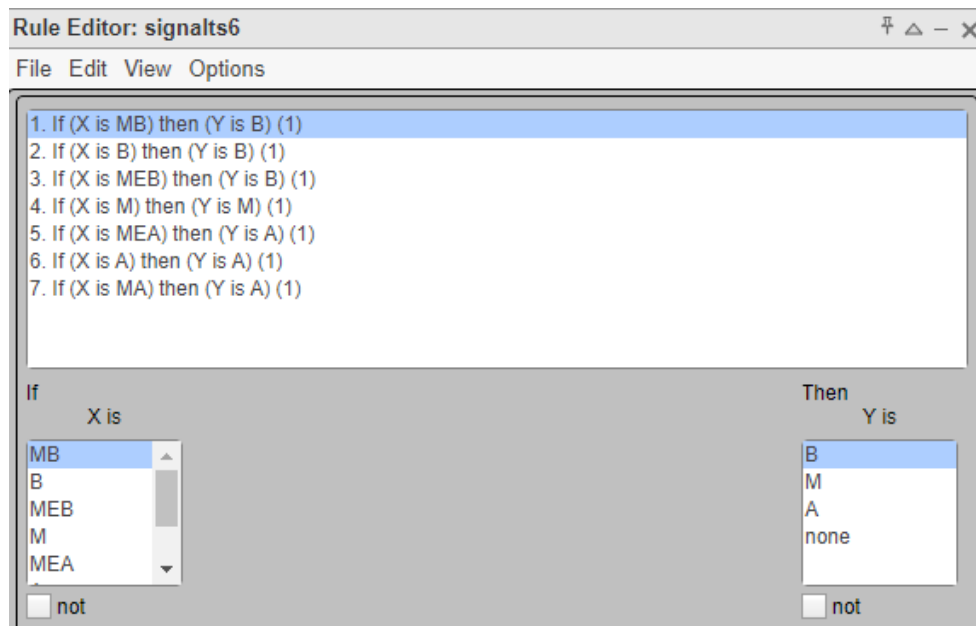
Configuración de funciones de pertenencia (7)

```
%Entrada, salida y funciones de pertenencia
a=newfis('signalts');
a=addvar(a,'input','X',[-1.5 1.5]);
a=addmf(a,'input',1,'MB','gaussmf',[abs(x(1)) x(2)]);
a=addmf(a,'input',1,'B','gaussmf',[abs(x(3)) x(4)]);
a=addmf(a,'input',1,'MEB','gaussmf',[abs(x(5)) x(6)]);
a=addmf(a,'input',1,'M','gaussmf',[abs(x(7)) x(8)]);
a=addmf(a,'input',1,'MEA','gaussmf',[abs(x(9)) x(10)]);
a=addmf(a,'input',1,'A','gaussmf',[abs(x(11)) x(12)]);
a=addmf(a,'input',1,'MA','gaussmf',[abs(x(13)) x(14)]);
%plotmf(a,'input',1)

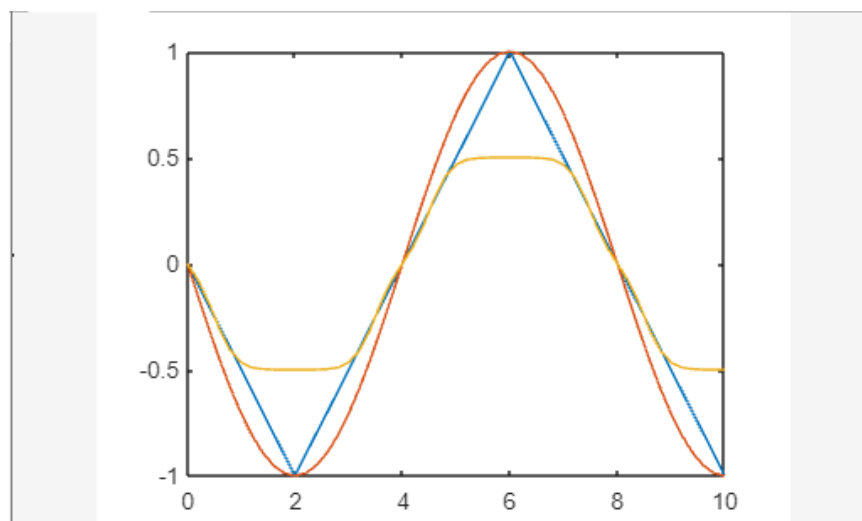
a=addvar(a,'output','Y',[-1.5 1.5]);
a=addmf(a,'output',1,'B','gaussmf',[abs(x(15)) x(16)]);
a=addmf(a,'output',1,'M','gaussmf',[abs(x(17)) x(18)]);
a=addmf(a,'output',1,'A','gaussmf',[abs(x(19)) x(20)]);
%plotmf(a,'output',1)
```

Configuración de reglas difusas

```
%Reglas del sistema difuso
ruleList=[
    1 1 1 1
    2 1 1 1
    3 1 1 1
    4 2 1 1
    5 3 1 1
    6 3 1 1
    7 3 1 1];
```



Gráfica de salida sin optimizar



Resultados de Optimización de sistema difuso con algoritmo Cuasi-Newton (MSE = 8.3411e-06)

Iteration	Func-count	f(x)	Step-size	optimality
60	1533	9.19606e-06	1	0.000515
61	1554	9.02568e-06	1	0.000544
62	1596	8.68799e-06	2.00142	0.000165
63	1638	8.61056e-06	0.258043	0.00036
64	1659	8.51357e-06	1	0.000166
65	1701	8.50259e-06	0.5	6.98e-05
66	1785	8.5001e-06	0.0464175	5.36e-05
67	1806	8.49666e-06	1	6.97e-05
68	1827	8.49399e-06	1	3.68e-05
69	1848	8.4908e-06	1	8.02e-05
70	1869	8.48513e-06	1	0.000116
71	1890	8.45806e-06	1	0.000296
72	1911	8.44194e-06	1	0.0003
73	1932	8.4128e-06	1	0.000134
74	1953	8.39275e-06	1	0.000132
75	1974	8.34107e-06	1	0.000273

x =

Columns 1 through 14

0.2000 -1.5000 0.2197 -0.9711 0.4689 -0.5327 -0.0265 0.0054 0.4556 0.4802 0.2022 0.9996 0.2000 1.5000

Columns 15 through 20

0.1959 -1.0345 -0.0634 -0.0034 0.1843 1.0348

[mamfis](#) with properties:

```

        Name: "signalts"
        AndMethod: "min"
        OrMethod: "max"
        ImplicationMethod: "min"
        AggregationMethod: "max"
        DefuzzificationMethod: "centroid"
        DisableStructuralChecks: 0
        Inputs: [1x1 fisvar]
        Outputs: [1x1 fisvar]
        Rules: [1x7 fisrule]

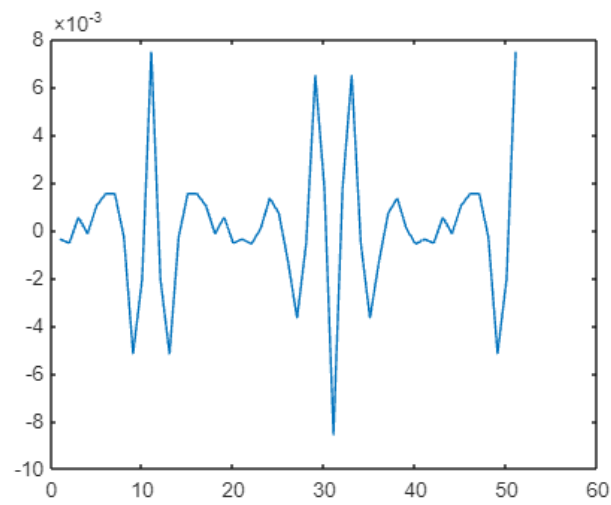
```

See 'getTunableSettings' method for parameter optimization.

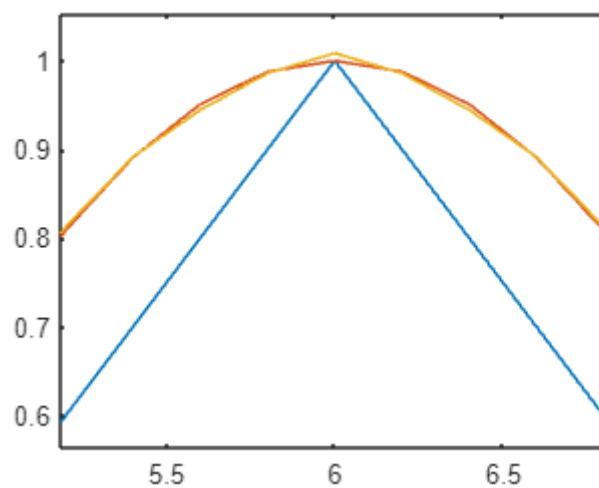
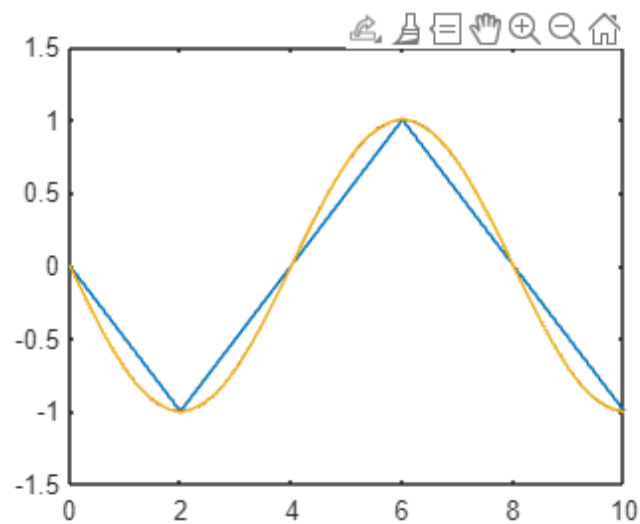
mse =

8.3411e-06

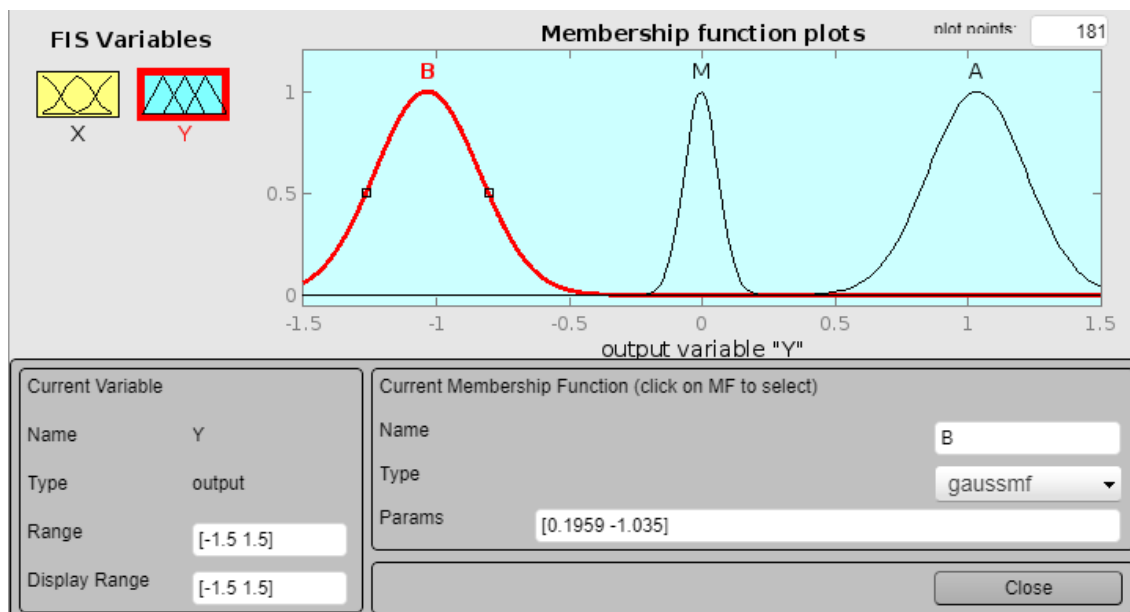
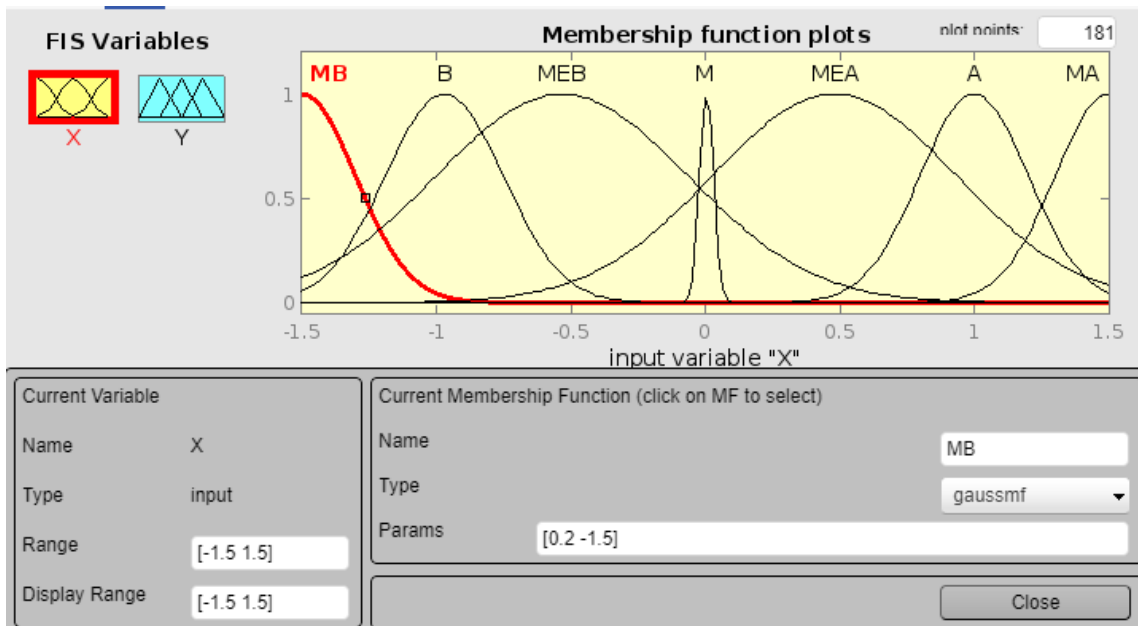
Figura del error



Gráfica de salida optimizada



Sistema difuso optimizado con Cuasi Newton (Entradas y Salidas)



Visualización de reglas de sistema difuso

