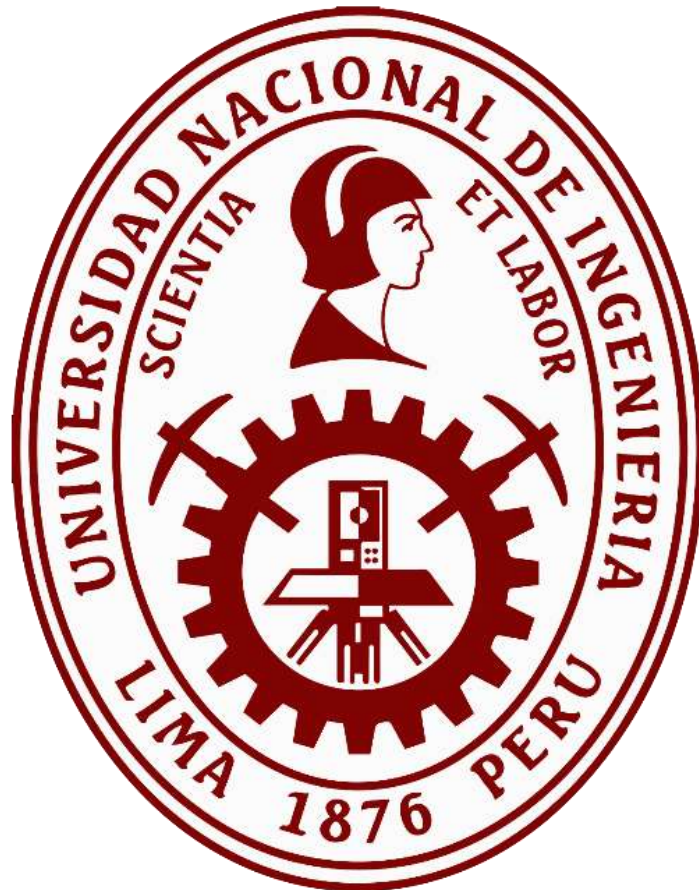


UNIVERSIDAD NACIONAL DE INGENIERIA



Tarea05: Redes neuronales dinámicas

Carlos Alberto Espinoza Mansilla | Inteligencia Artificial | 01/05/2019

Con el archivo 'DynamicBPModelamiento.m' vamos a comprobar la dependencia de los valores introducidos al programa y como afectan su rapidez de procesamiento y el error del resultado de una red neuronal dinámica.

1er Entrenamiento de u_1:

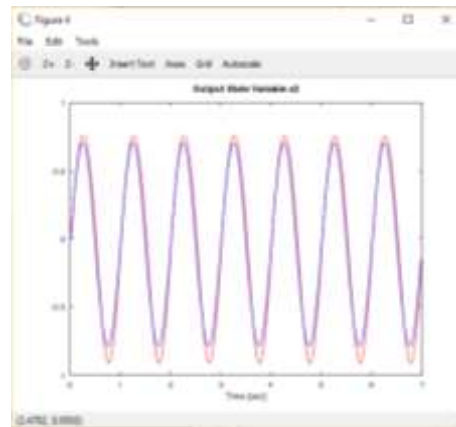
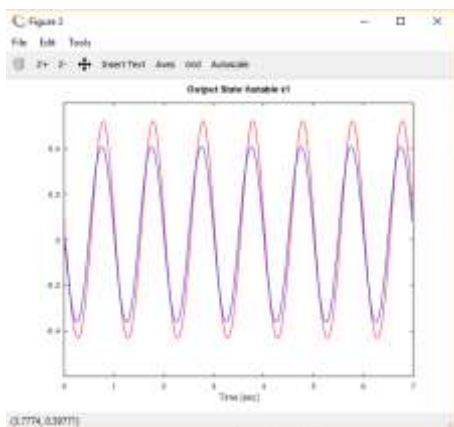
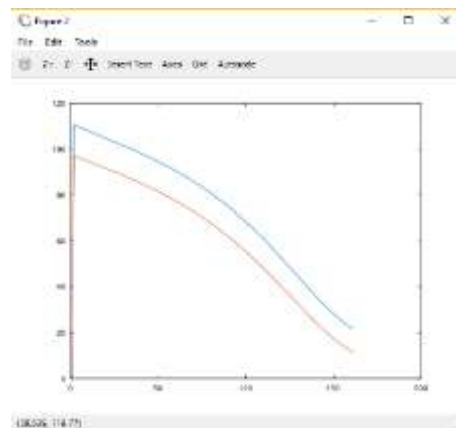
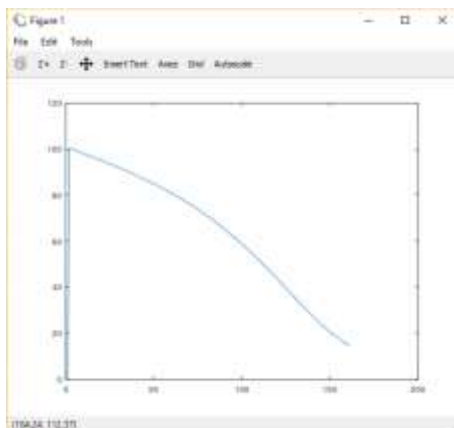
Introduce learning rate [v w]: 0.04

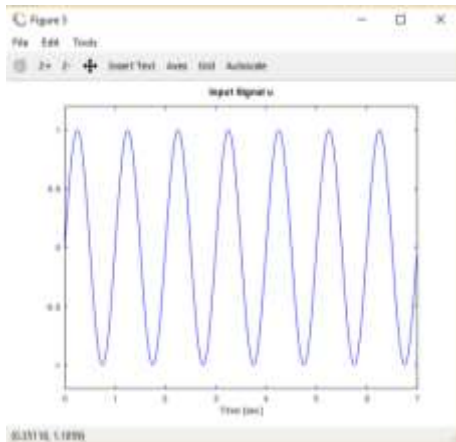
Introduce learning rate [c: sigmoid center]: 0.06

Introduce learning rate [a: sigmoid slope]: 0.05

Introduce maximum value of error function (percentage %): 15

Introduce number of iteration steps: 1000





erreltotal = 0.14834

Validando u_2 :

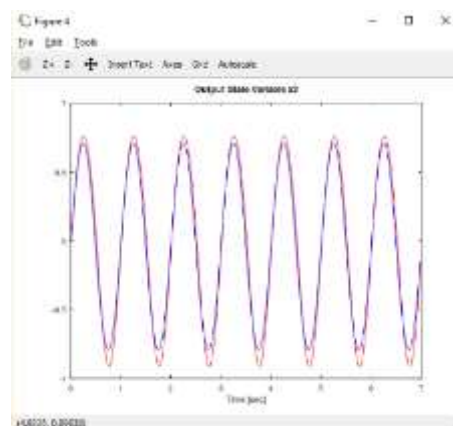
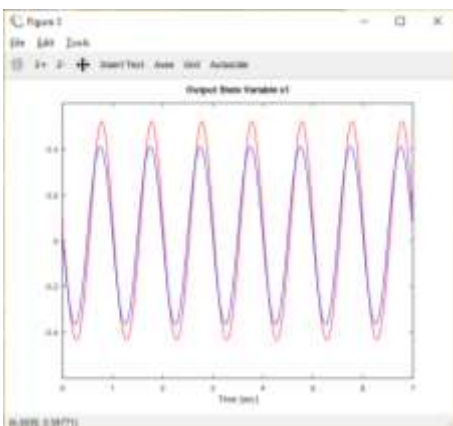
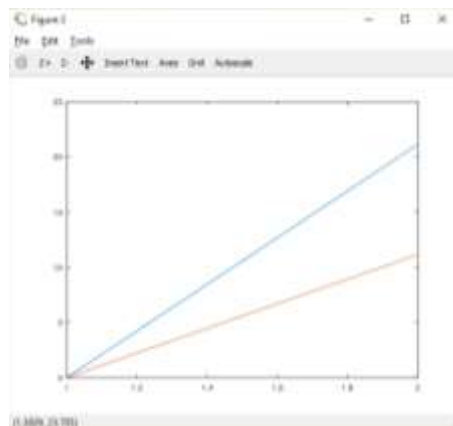
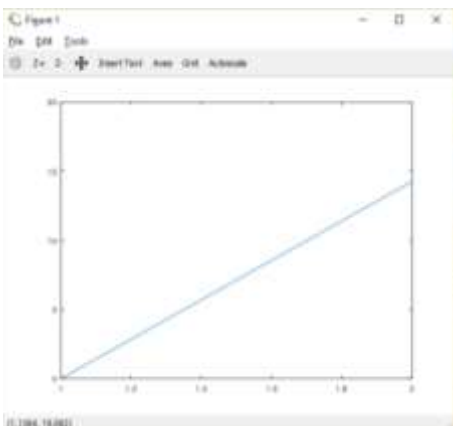
Introduce learning rate [v w]: 0

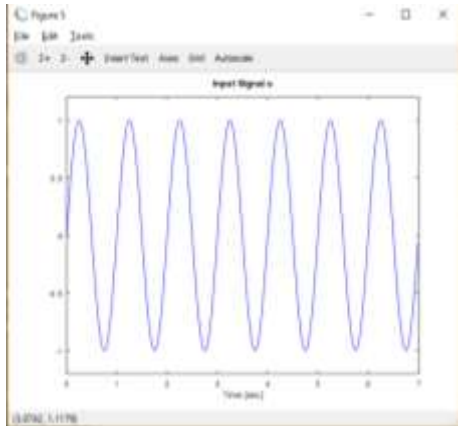
Introduce learning rate [c: sigmoid center]: 0

Introduce learning rate [a: sigmoid slope]: 0

Introduce maximum value of error function (percentage %): 15

Introduce number of iteration steps: 2





erreltotal = 0.14268

Validando u_3 :

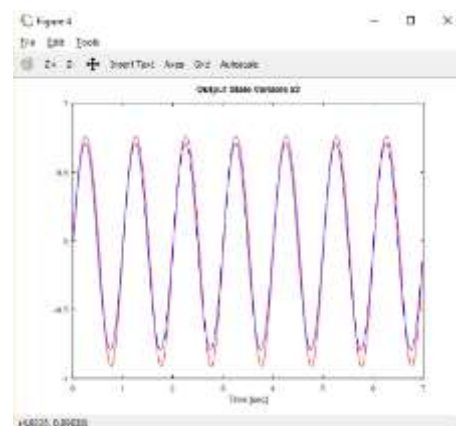
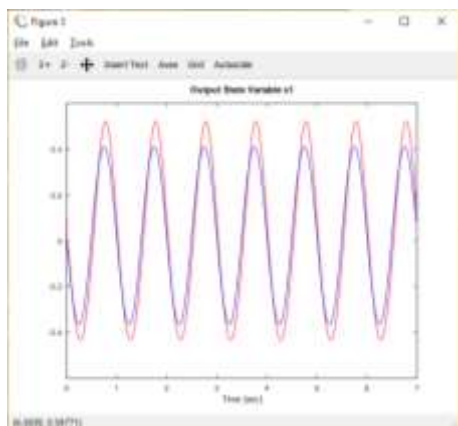
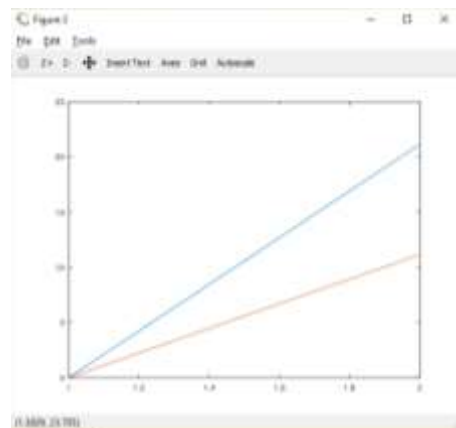
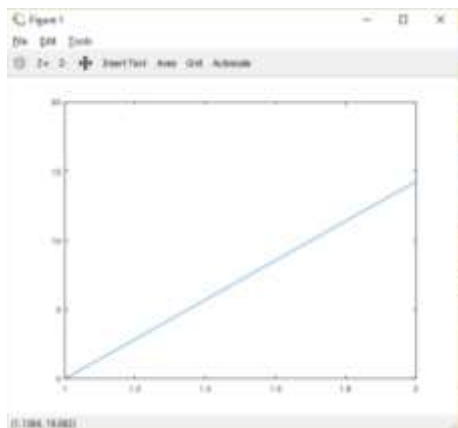
Introduce learning rate [v w]: 0

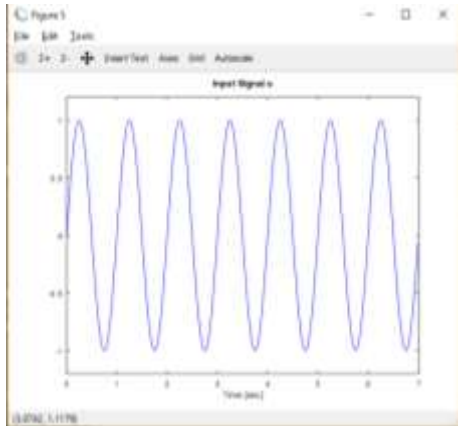
Introduce learning rate [c: sigmoid center]: 0

Introduce learning rate [a: sigmoid slope]: 0

Introduce maximum value of error function (percentage %): 15

Introduce number of iteration steps: 2





erreltotal = 0.14268

2do Entrenamiento de u_1 :

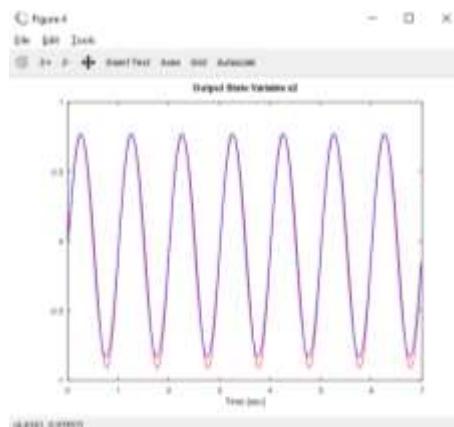
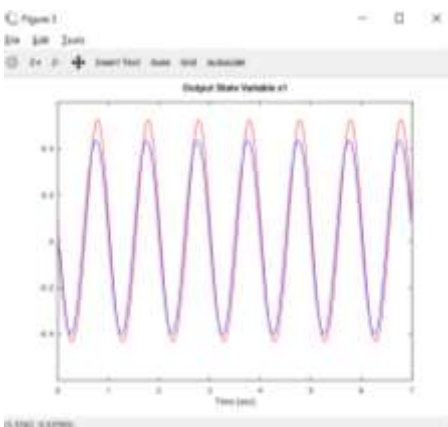
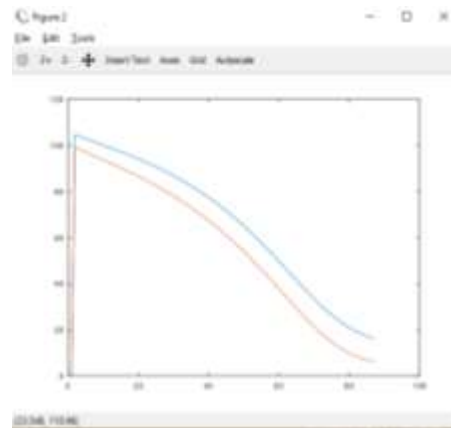
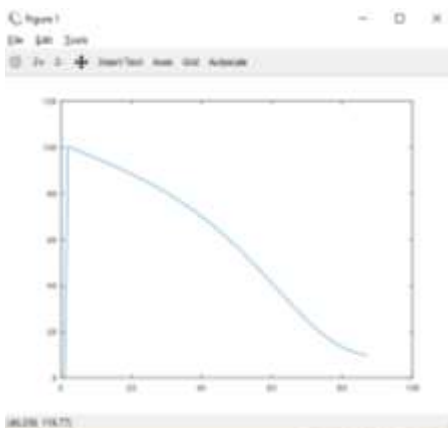
Introduce learning rate [v w]: 0.08

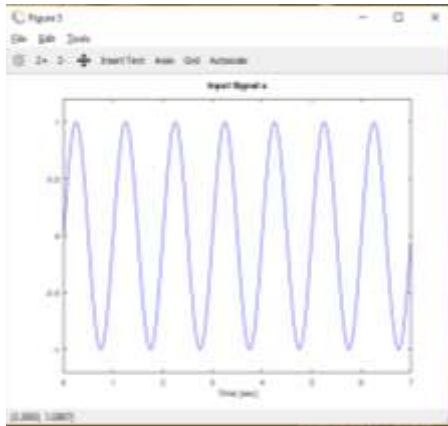
Introduce learning rate [c: sigmoid center]: 0.04

Introduce learning rate [a: sigmoid slope]: 0.04

Introduce maximum value of error function (percentage %) : 10

Introduce number of iteration steps: 1000





erreltotal = 0.096593

Validación de u_2 :

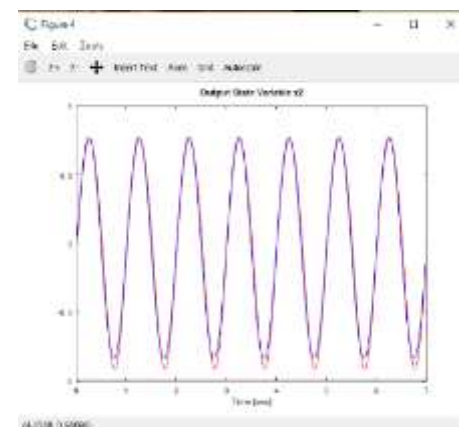
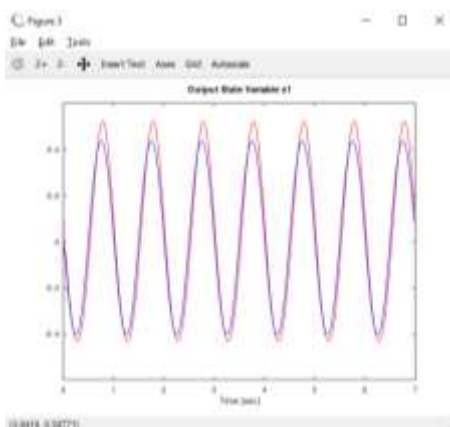
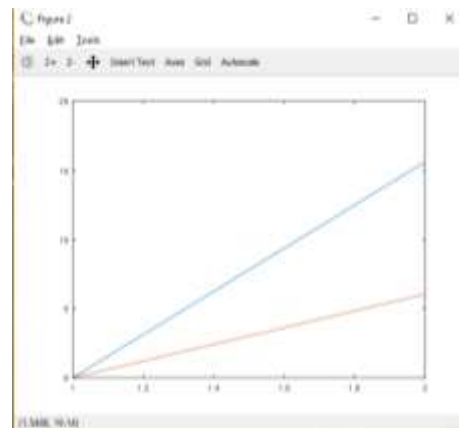
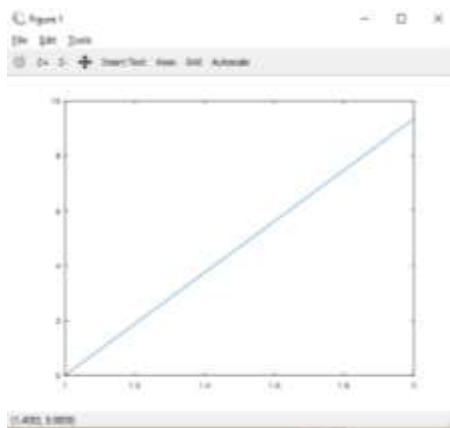
Introduce learning rate [v w]: 0

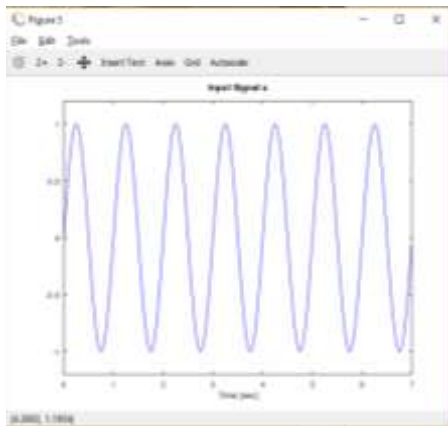
Introduce learning rate [c: sigmoid center]: 0

Introduce learning rate [a: sigmoid slope]: 0

Introduce maximum value of error function (percentage %) : 10

Introduce number of iteration steps: 2





erreltotal = 0.093536

Validando u_3:

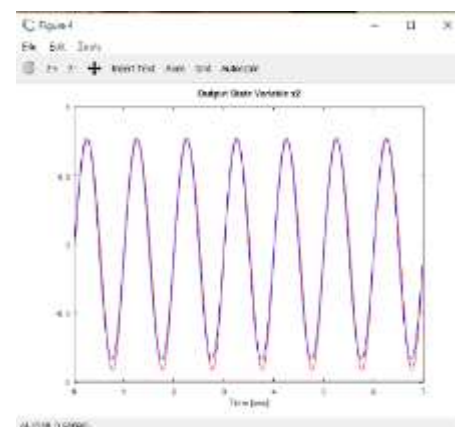
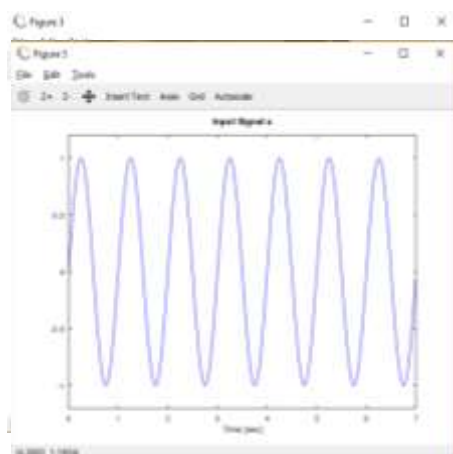
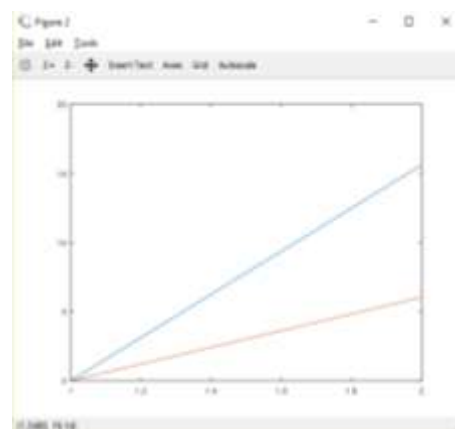
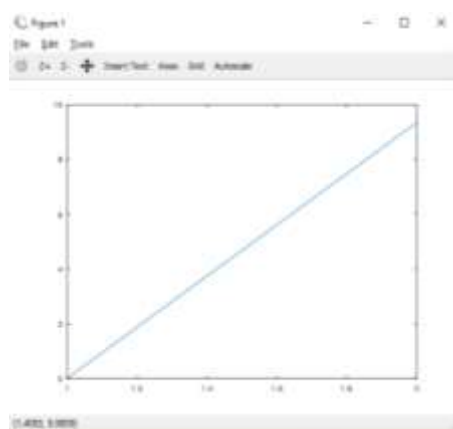
Introduce learning rate [v w]: 0

Introduce learning rate [c: sigmoid center]: 0

Introduce learning rate [a: sigmoid slope]: 0

Introduce maximum value of error function (percentage %): 10

Introduce number of iteration steps: 2



erreltotal = 0.093536

3er Entrenamiento de u_1 :

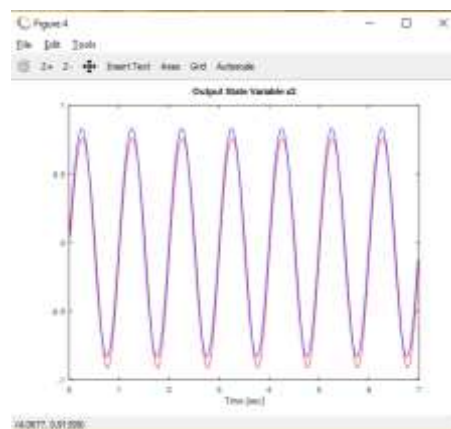
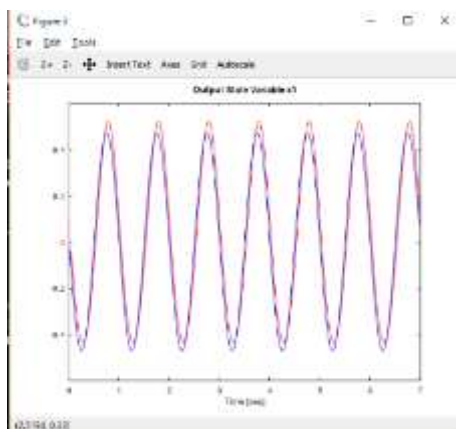
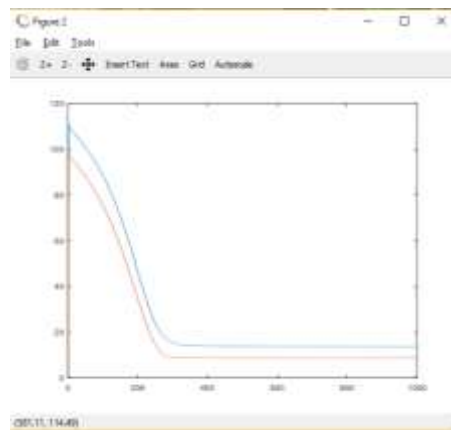
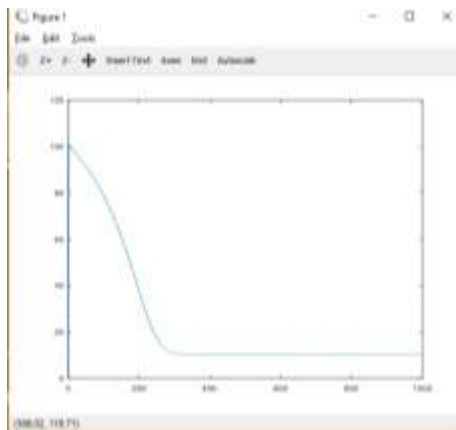
Introduce learning rate $[v \ w]$: 0.03

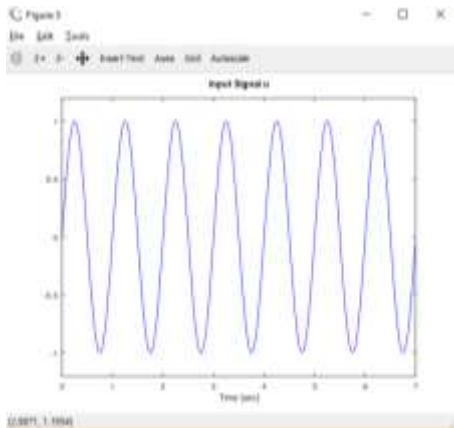
Introduce learning rate $[c: \text{sigmoid center}]$: 0

Introduce learning rate $[a: \text{sigmoid slope}]$: 0.02

Introduce maximum value of error function (percentage %): 7

Introduce number of iteration steps: 1000





erreltotal = 0.10063

Validando u_2 :

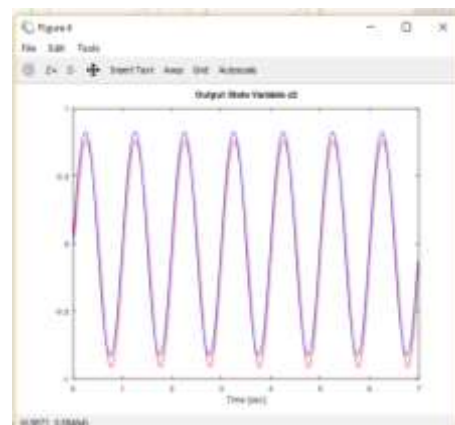
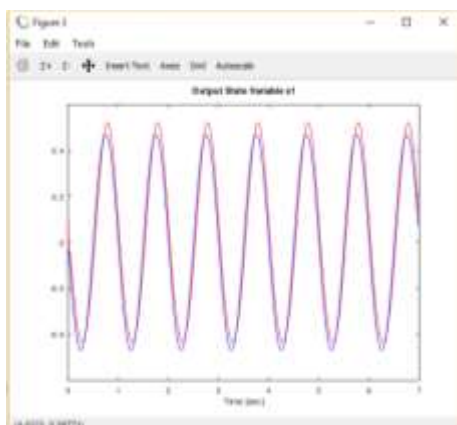
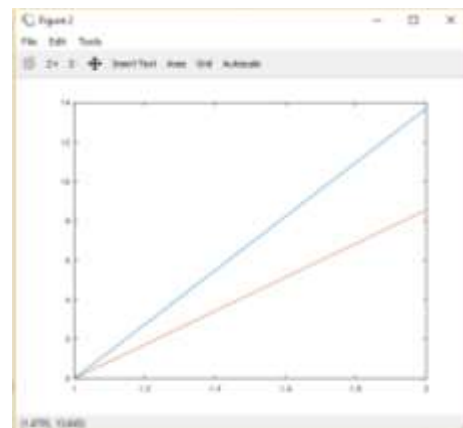
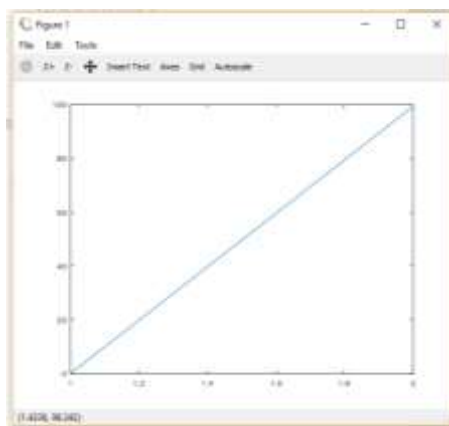
Introduce learning rate [v w]: 0

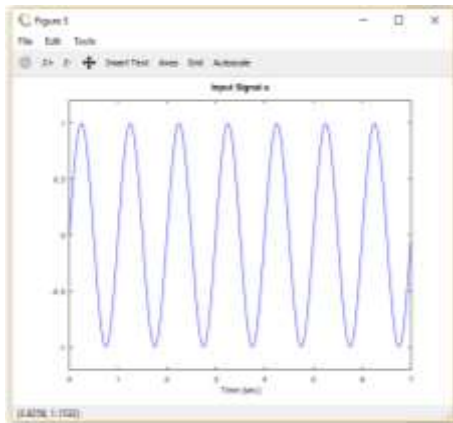
Introduce learning rate [c: sigmoid center]: 0

Introduce learning rate [a: sigmoid slope]: 0

Introduce maximum value of error function (percentage %): 7

Introduce number of iteration steps: 2





erreltotal = 0.10073

Validando u_3:

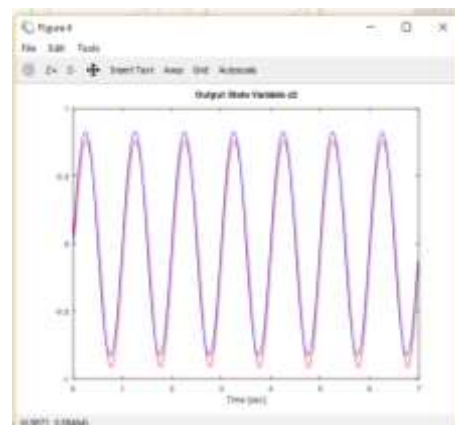
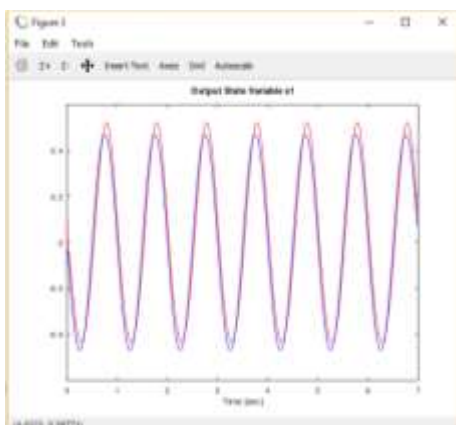
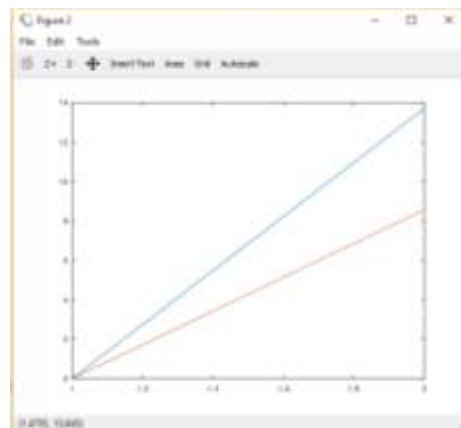
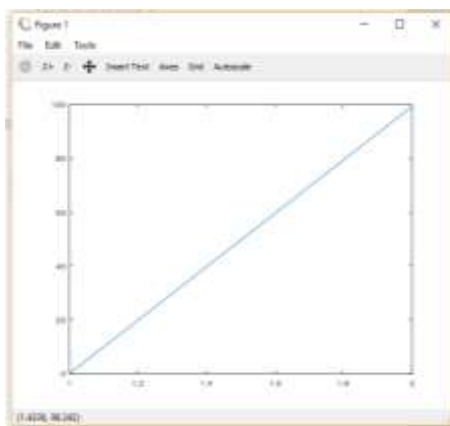
Introduce learning rate [v w]: 0

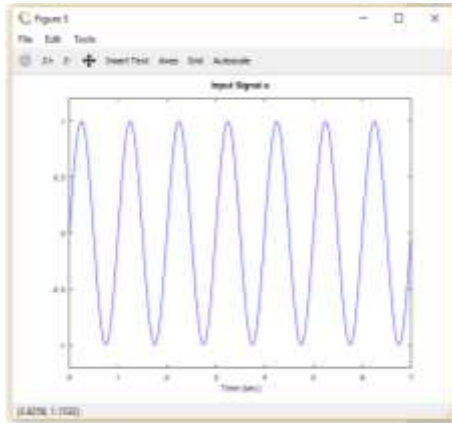
Introduce learning rate [c: sigmoid center]: 0

Introduce learning rate [a: sigmoid slope]: 0

Introduce maximum value of error function (percentage %): 7

Introduce number of iteration steps: 2





erreltotal = 0.10073

Conclusión:

De acuerdo a nuestras observaciones experimentales reducir el 'learning rate' a valores menores a 0.08 benefician una convergencia más rápida (menor número de iteraciones antes de converger) y que los valores en c: sigmoide center y a: sigmoid slope entre menores sean (tomando en cuenta también el % de error agregado) el error total obtenido disminuye.