

**THE UNIVERSITY OF TEXAS AT ARLINGTON, TEXAS  
DEPARTMENT OF ELECTRICAL ENGINEERING**

**EE 5327 - 001**

**SYSTEM IDENTIFICATION & ESTIMATION**

**HW # 5**

**ASSIGNMENT**

**by**

**SOUTRIK MAITI**

**1001569883**

**Presented to**

**Prof. Michael Niestroy**

**Nov 21, 2017**

Problem 1:

a)MATLAB Code for preparing data for training the Neural Net.

clc

clear all

close all

w=0;

wk=[0.5 1.0 1.5 2.0 2.5 3.0];

t=0;

i=0;

nnin=zeros(1001,3);

nnout=zeros(1001,1);

for i=1:6

w=wk(i);

[t,y]=sim('Hw5P1',[0 10]);

if i==1

nnin=NN\_In;

nnout=NN\_Out;

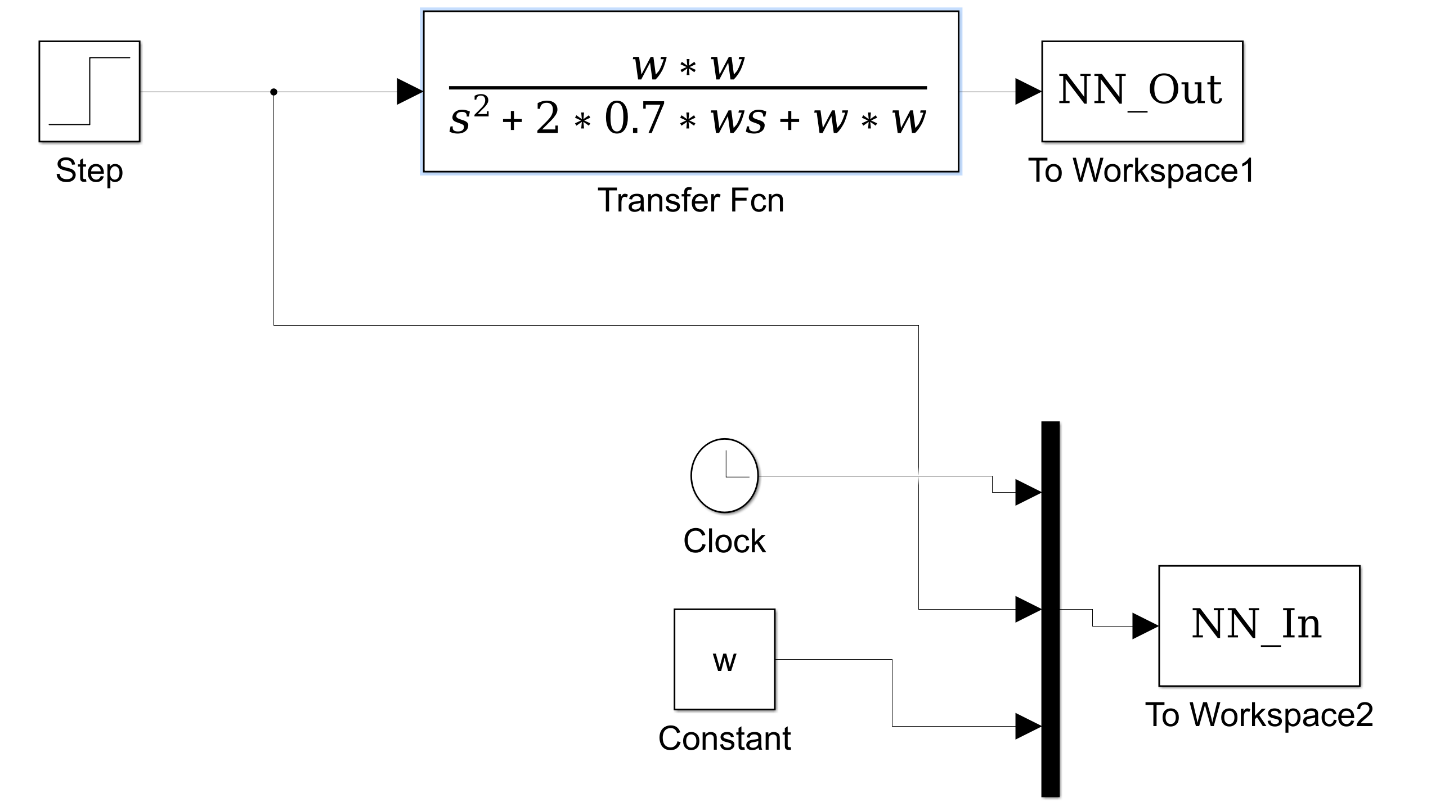
else

nnin=[nnin;NN\_In];

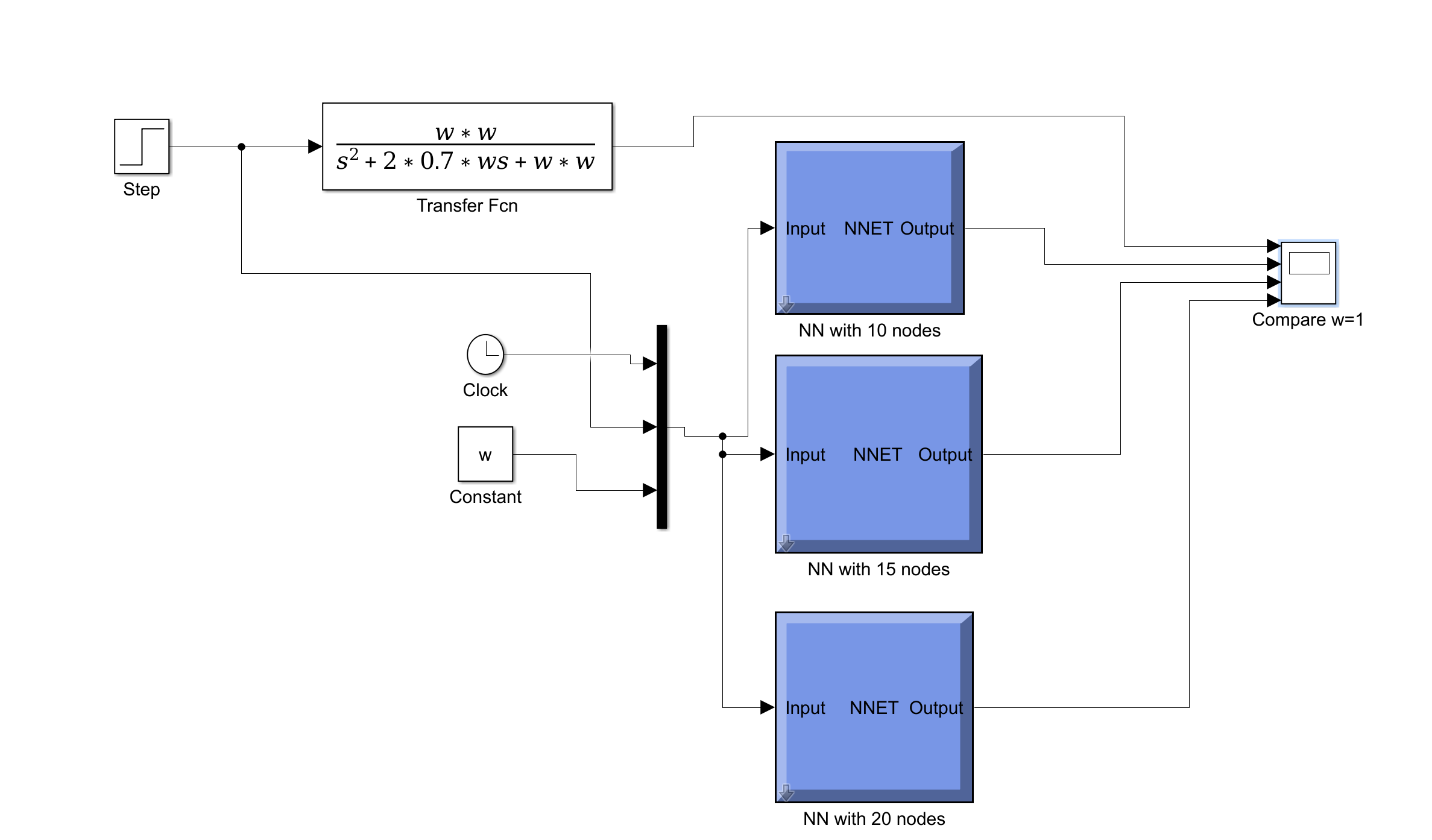
nnout=[nnout;NN\_Out];

end

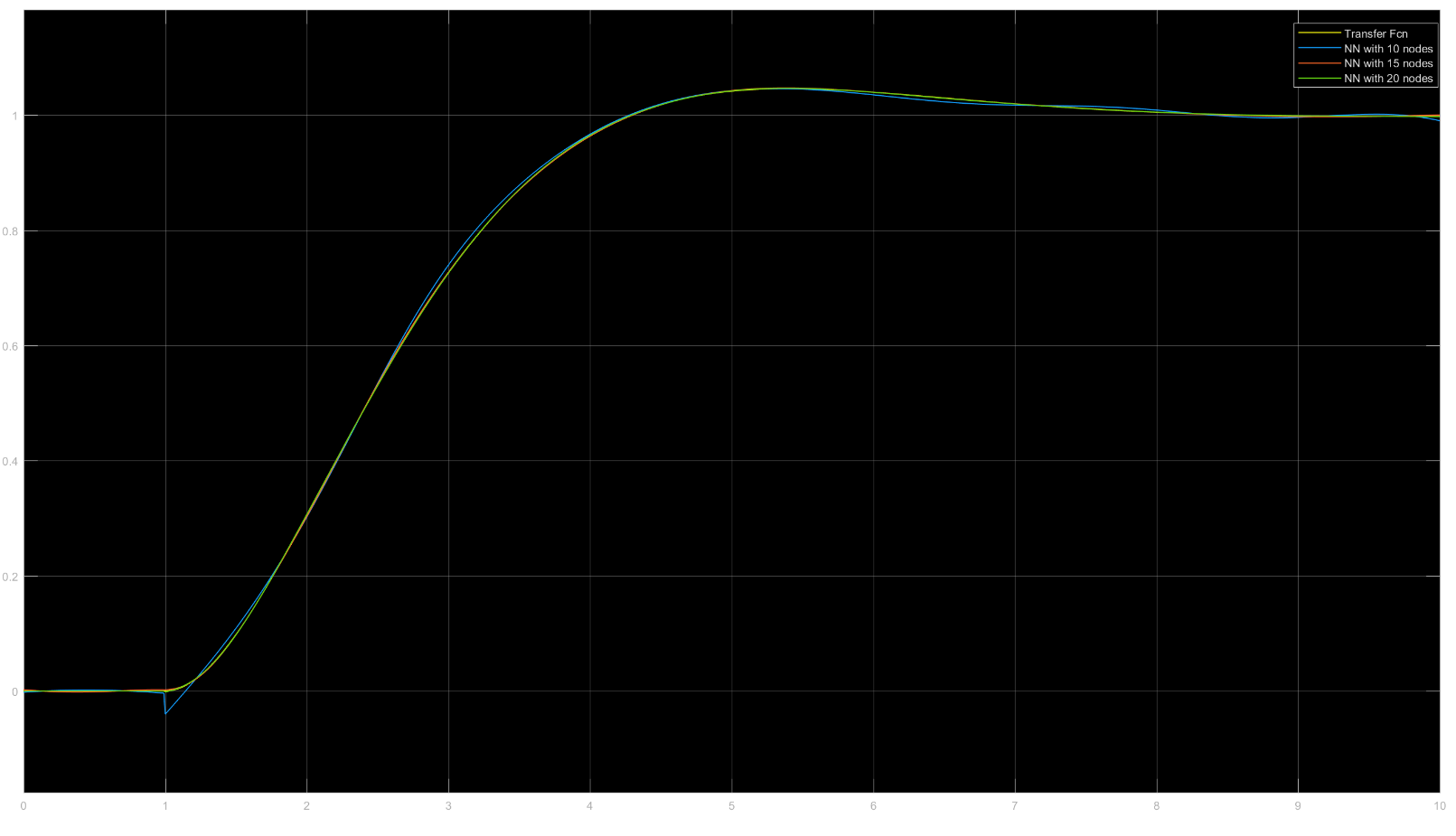
end



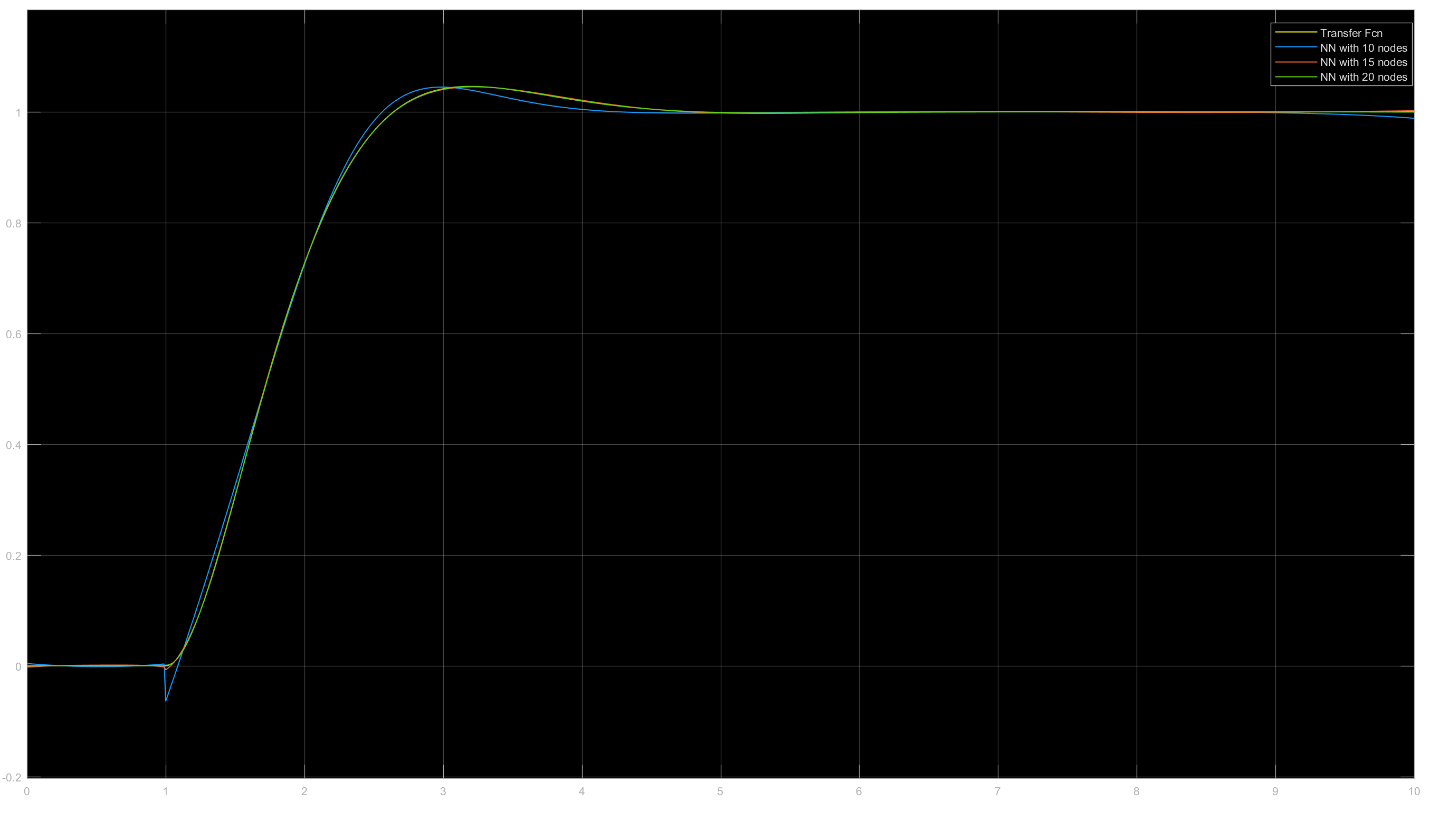
**Fig1.1 Simulink diagram while creating the training the data for the neural net.**



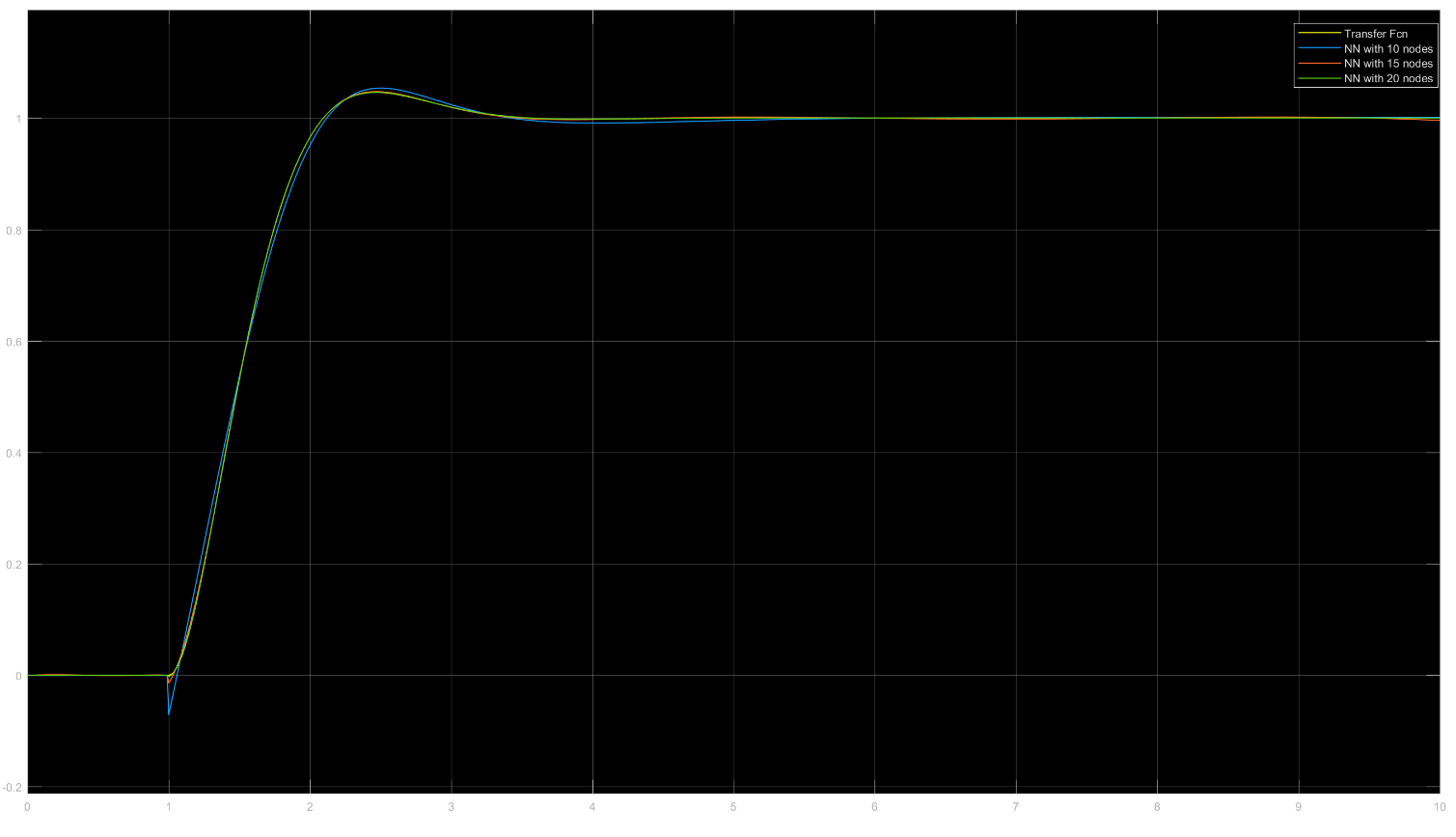
**Fig 1.2 Simulink diagram of the NNs with 10,15, and 20 nodes respectively.**



**Fig1.3 Comparison of the outputs when w=1**

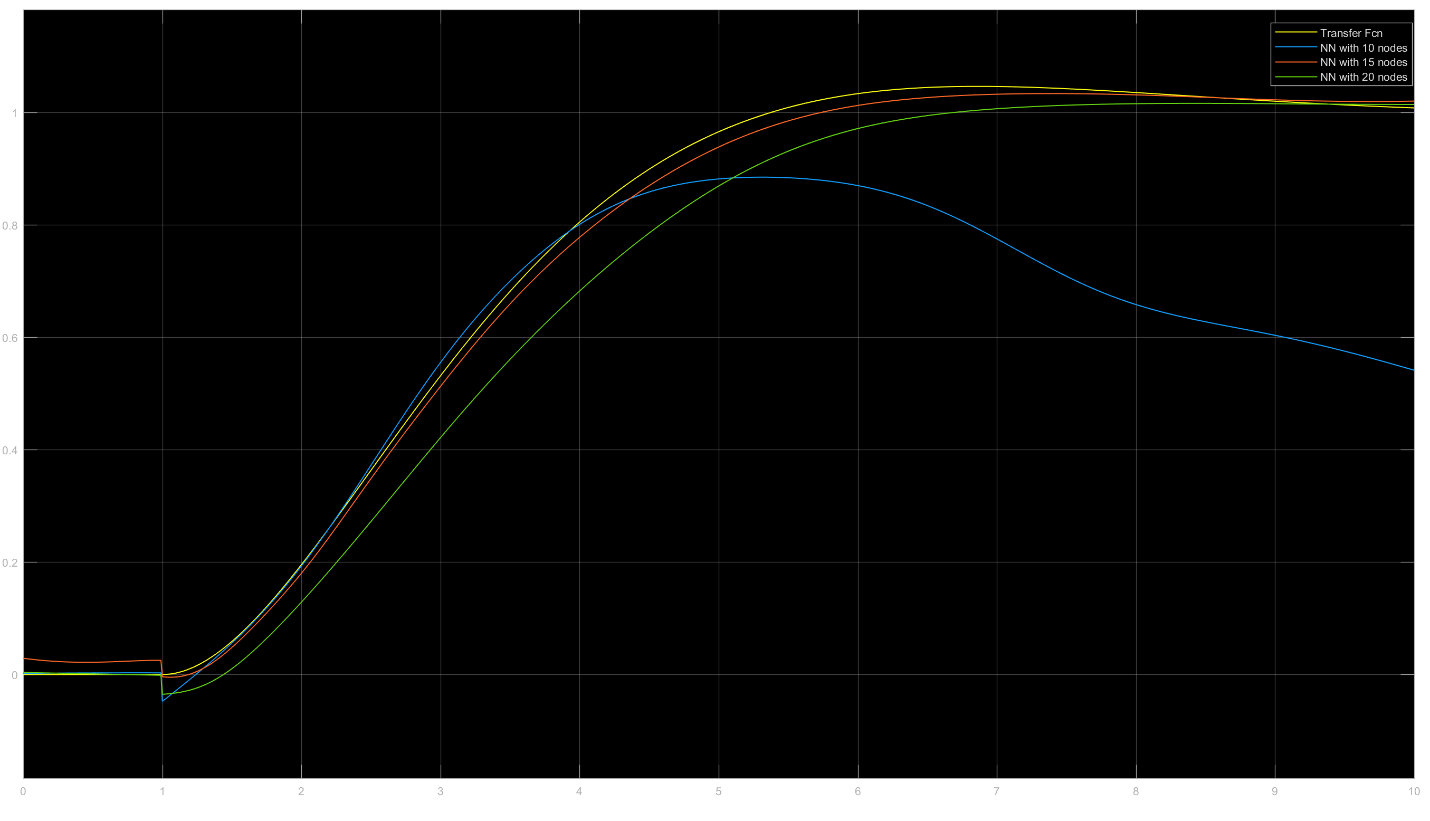


**Fig 1.4 Comparison of outputs when w=2**

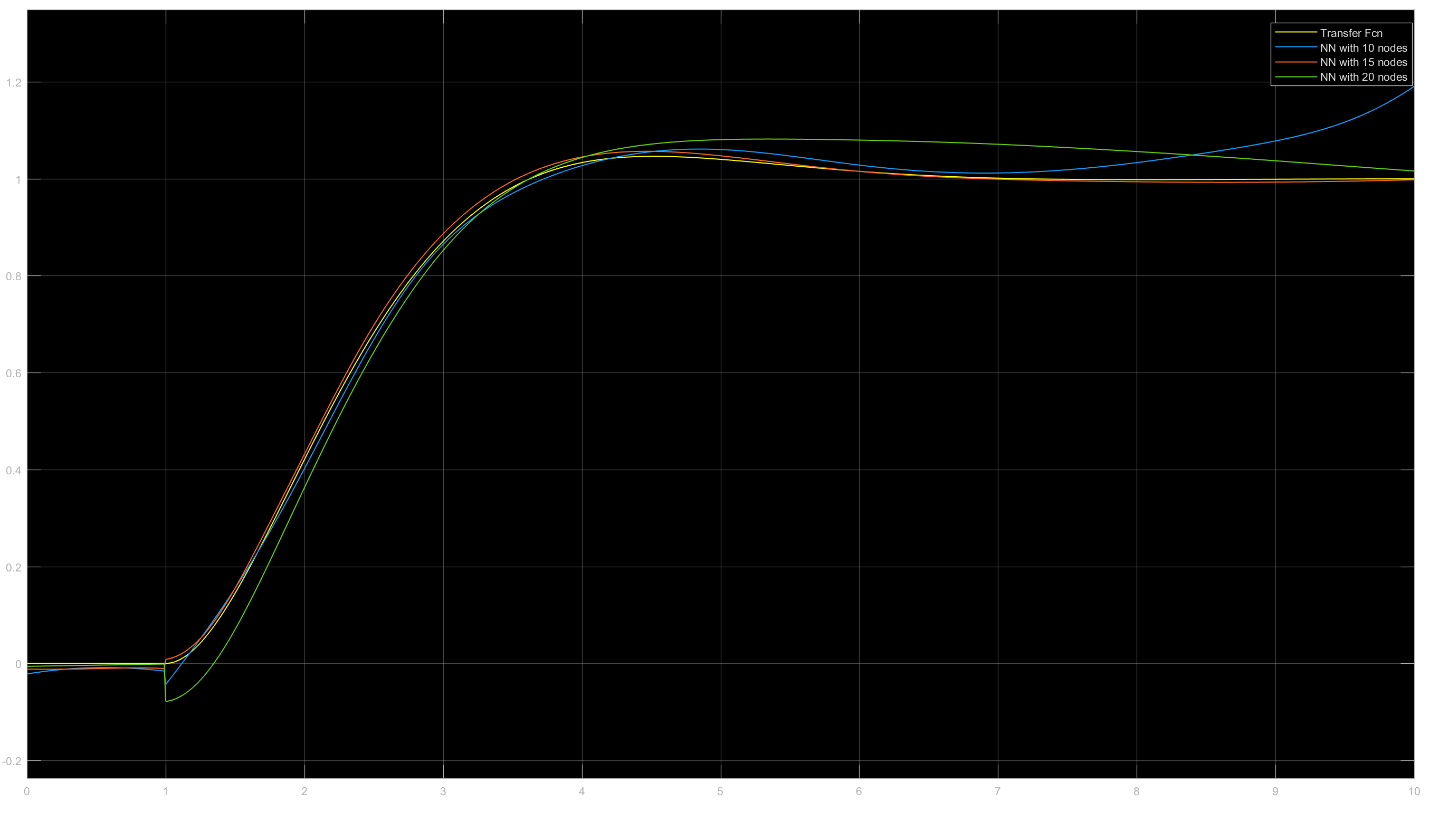


**Fig 1.5 Comparison of outputs when w=3**

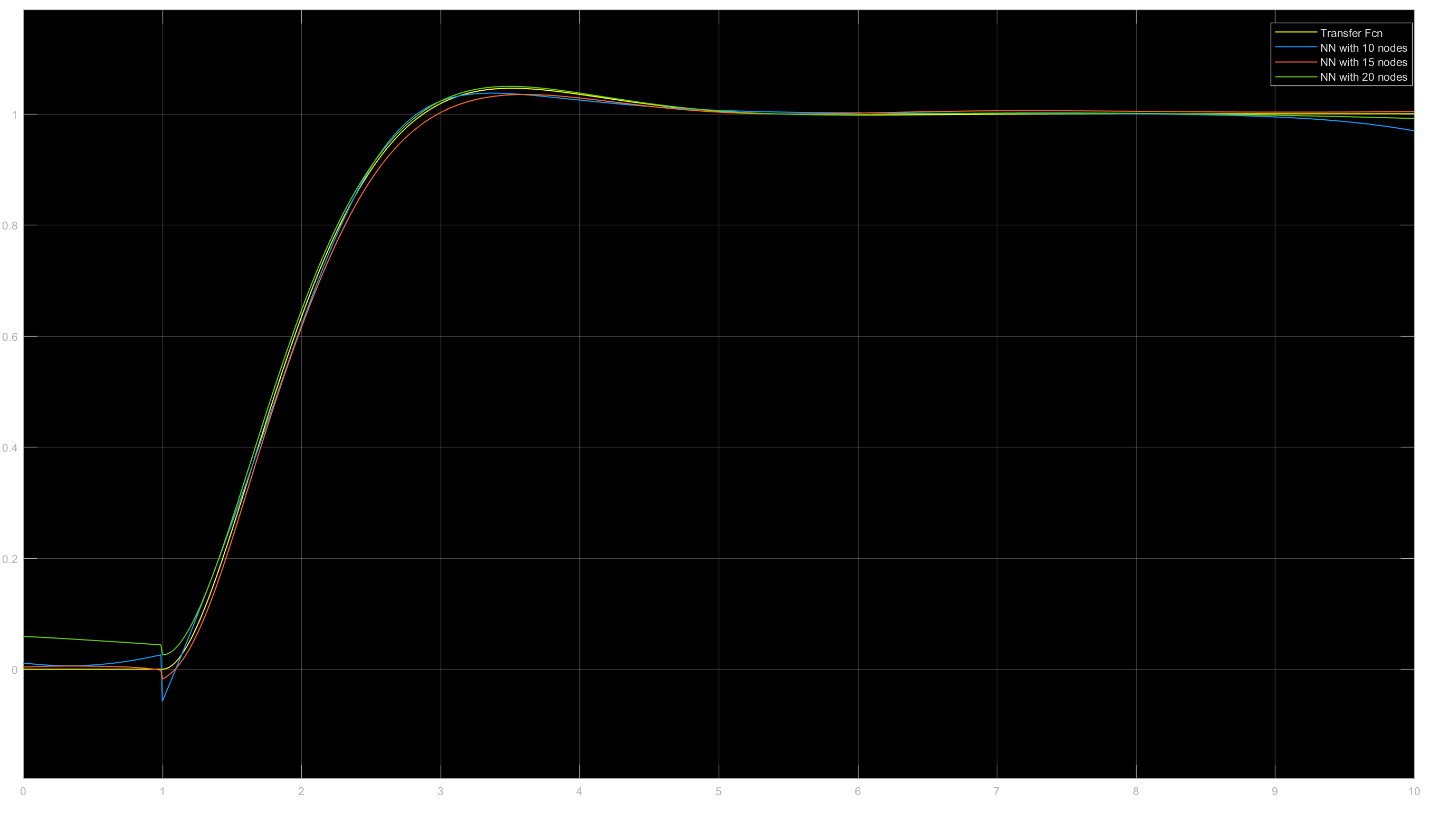
b)



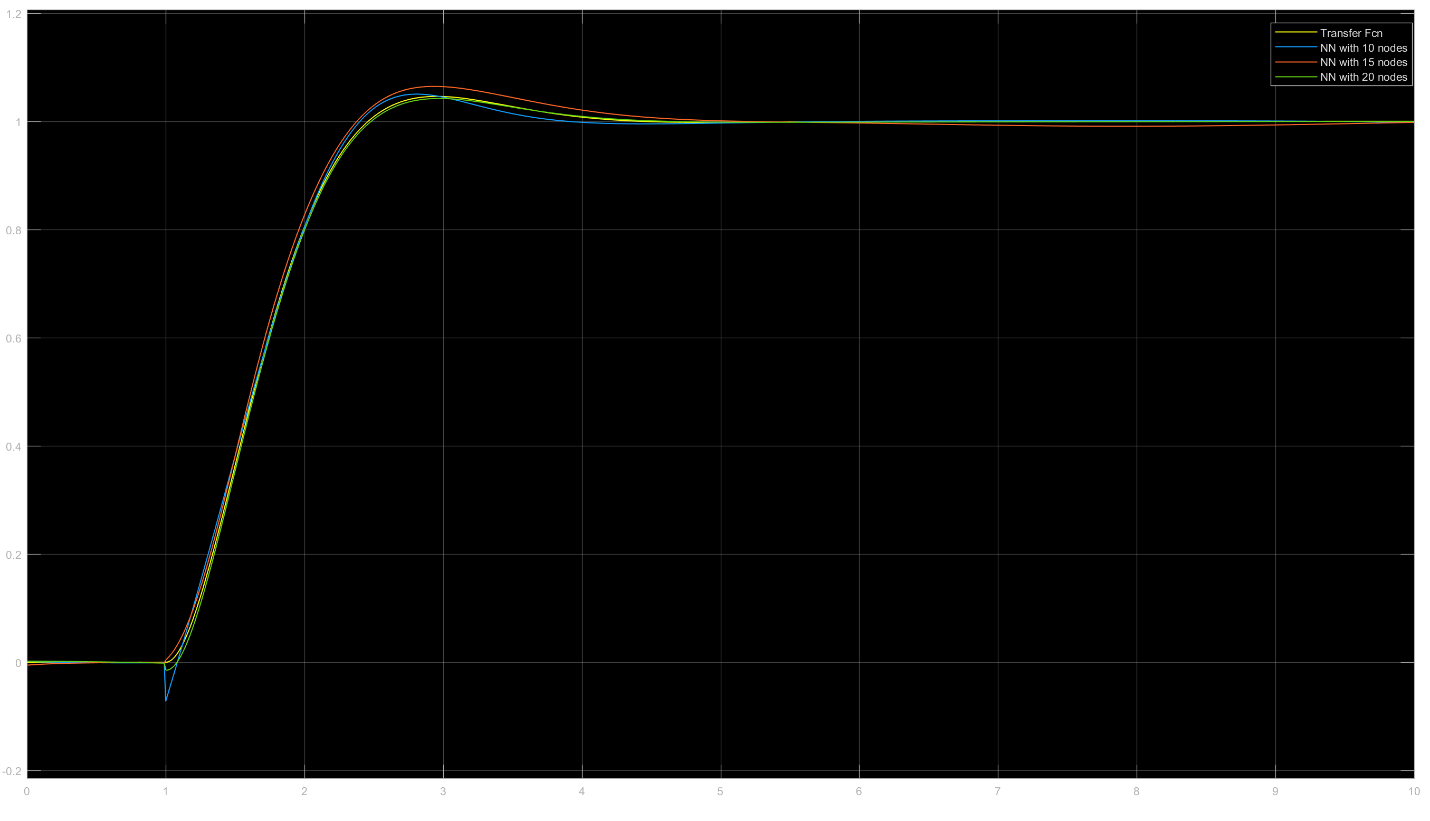
**Fig 1.6 Comparison of NN outputs with original TF output when w=0.75**



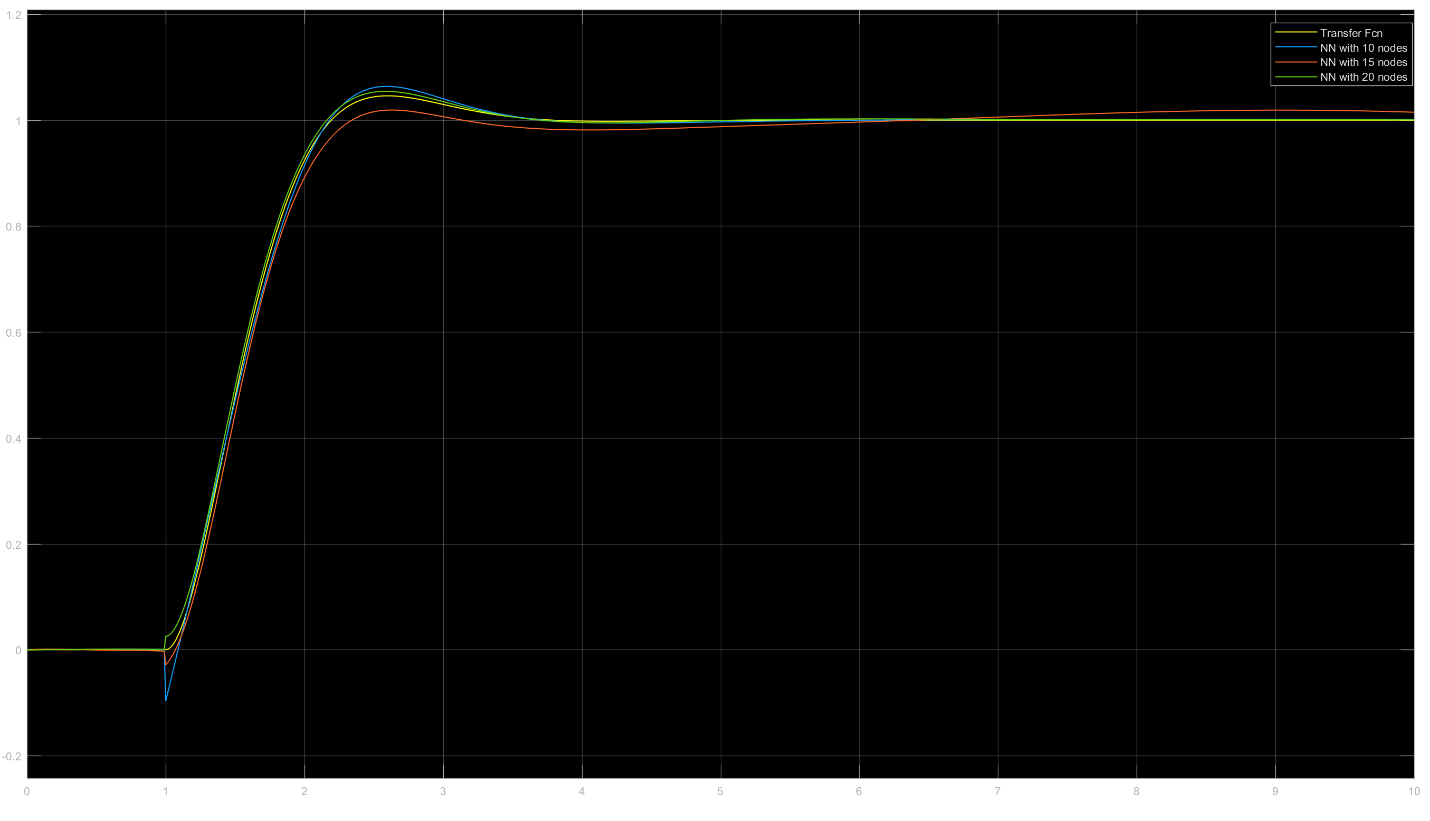
**Fig 1.7** **Comparison of NN outputs with original TF output when w=1.25**



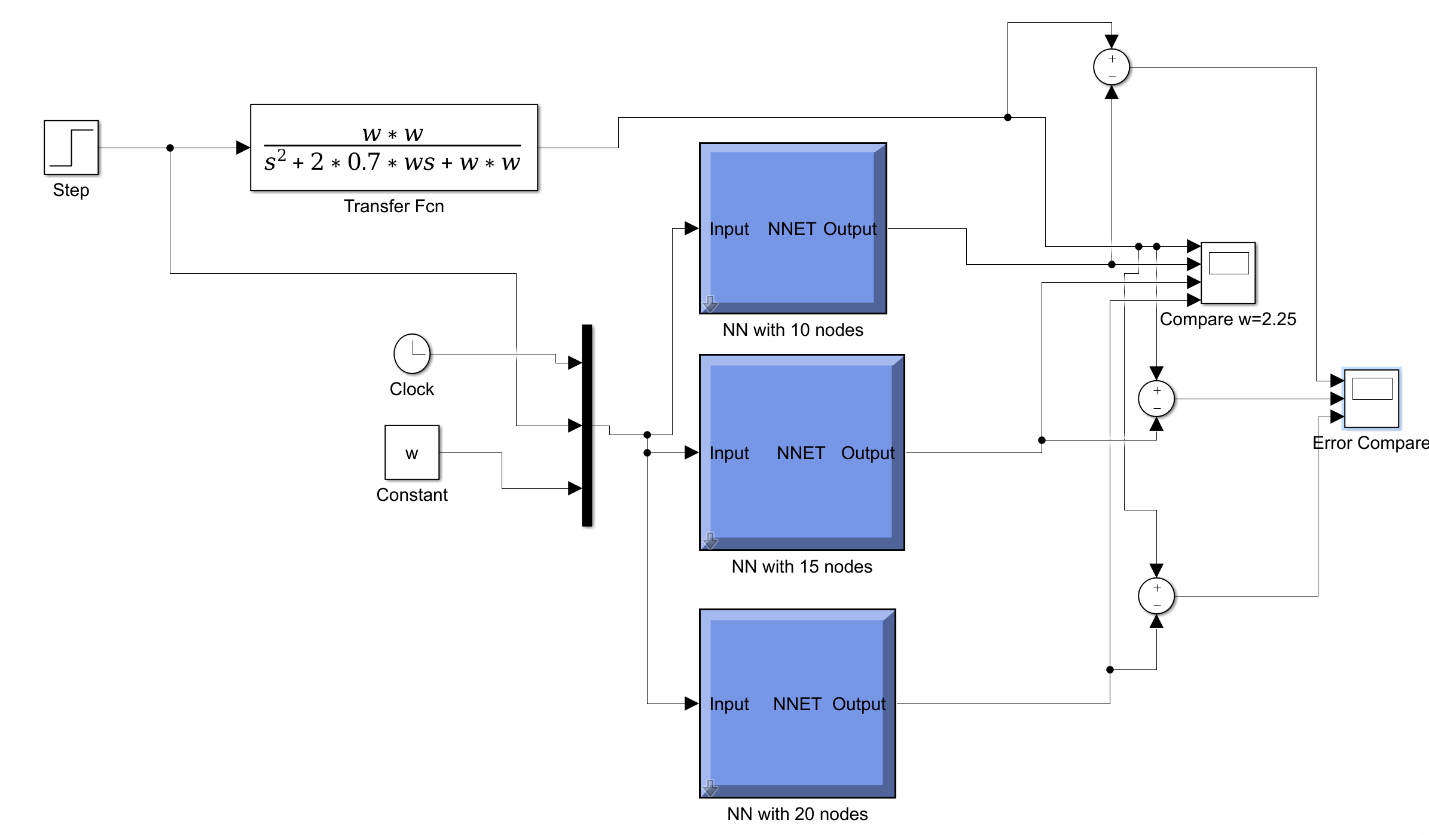
**Fig 1.8** **Comparison of NN outputs with original TF output when w=1.75**



**Fig 1.9** **Comparison of NN outputs with original TF output when w=2.25**

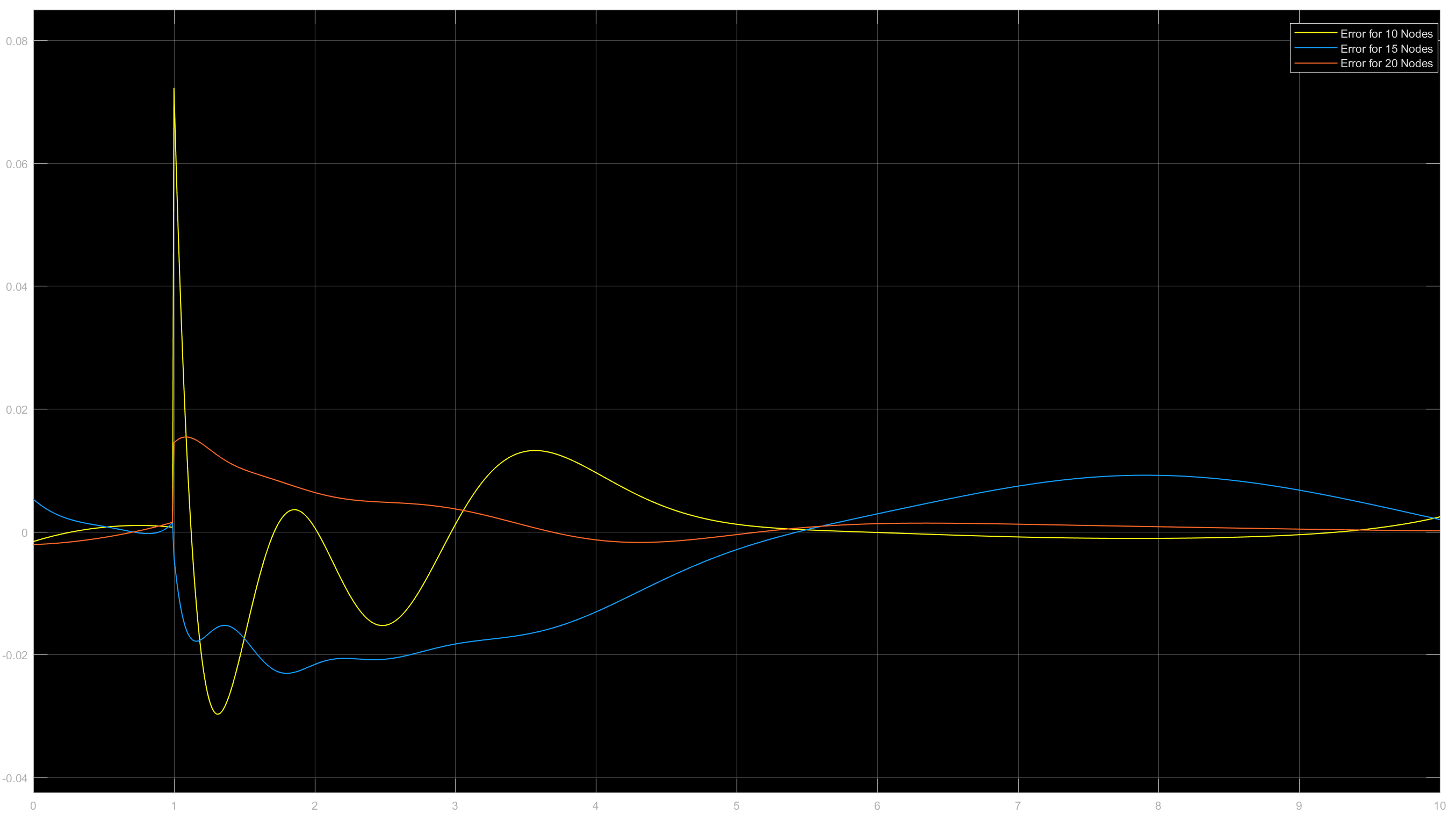


**Fig 1.10** **Comparison of NN outputs with original TF output when w=2.75**



**Fig 1.11** **Simulink diagram for error comparison**

c) From the figure below, we can see that the error for NN with 20 nodes is the least. Therefore for an appropriate value of w and larger number of nodes, the NN produces good result for this application.



**Fig 1.12** **Comparison of NN output errors with original TF**

Problem 2:

*MATLAB code for generating the training data for Neural Net*

clc

clear all

close all

w=0;

wk=[0.5 1.0 1.5 2.0 2.5 3.0];

zk=[0.2 0.4 0.6 0.8 1.0];

t=0;

i=0;

nnin=zeros(1001,4);

nnout=zeros(1001,1);

for k=1:5

z=zk(k);

for i=1:6

w=wk(i);

[t,y]=sim('Hw5P2',[0 10]);

if i==1 && k==1

nnin=NN\_In;

nnout=NN\_Out;

else

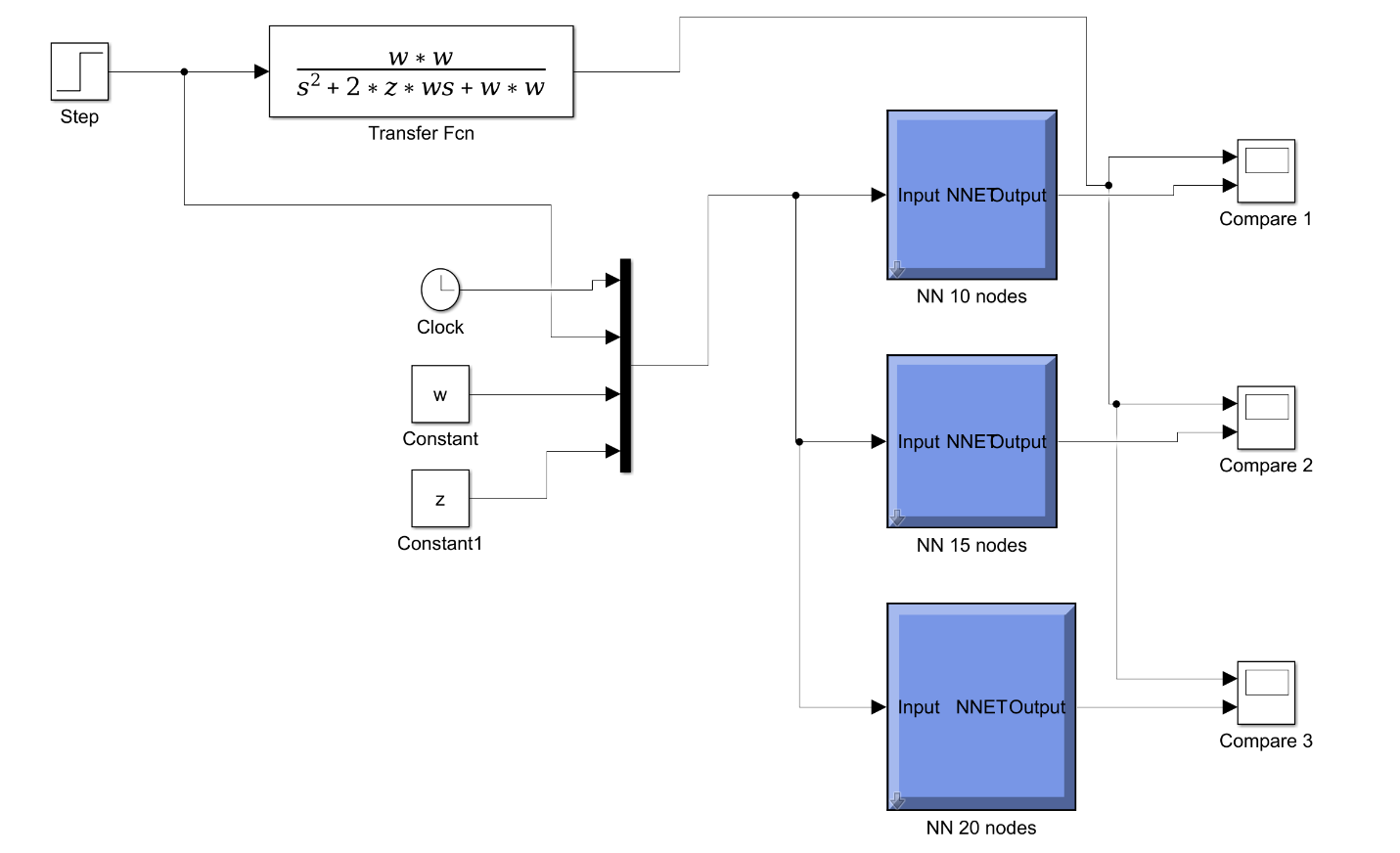
nnin=[nnin;NN\_In];

nnout=[nnout;NN\_Out];

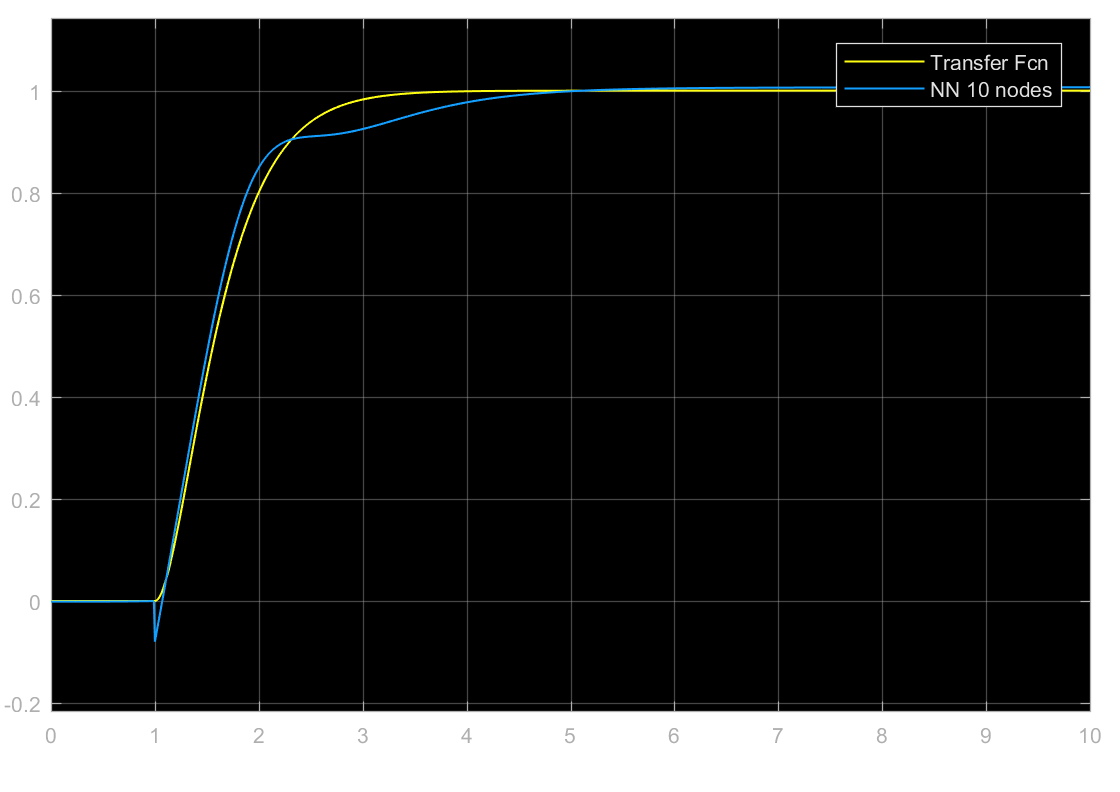
end

end

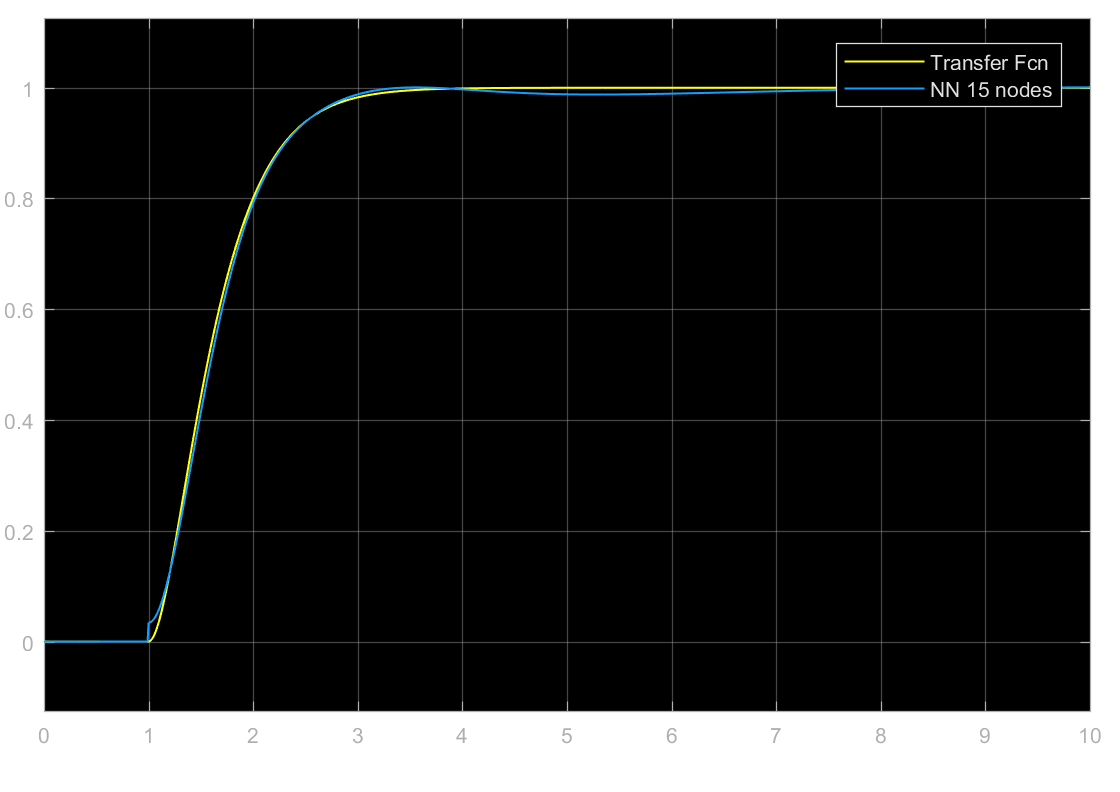
end



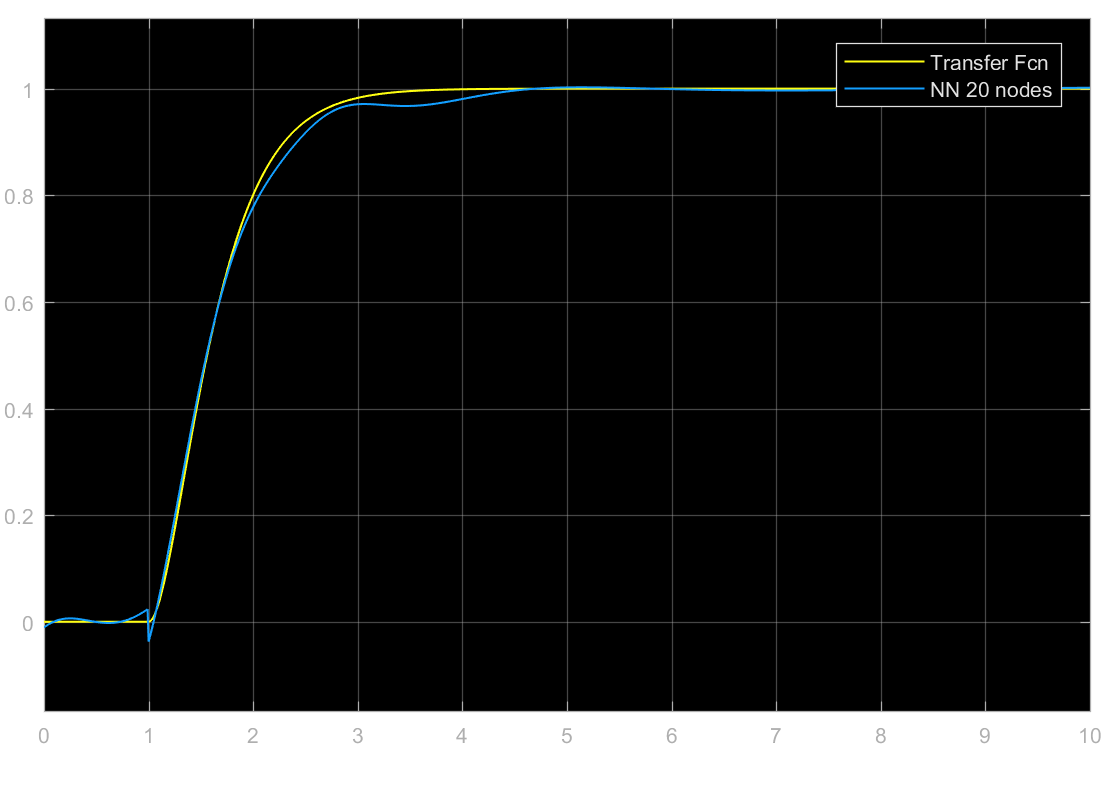
**Fig 1.13** **Simulink diagram for NN with 10,15, and 20 nodes.**



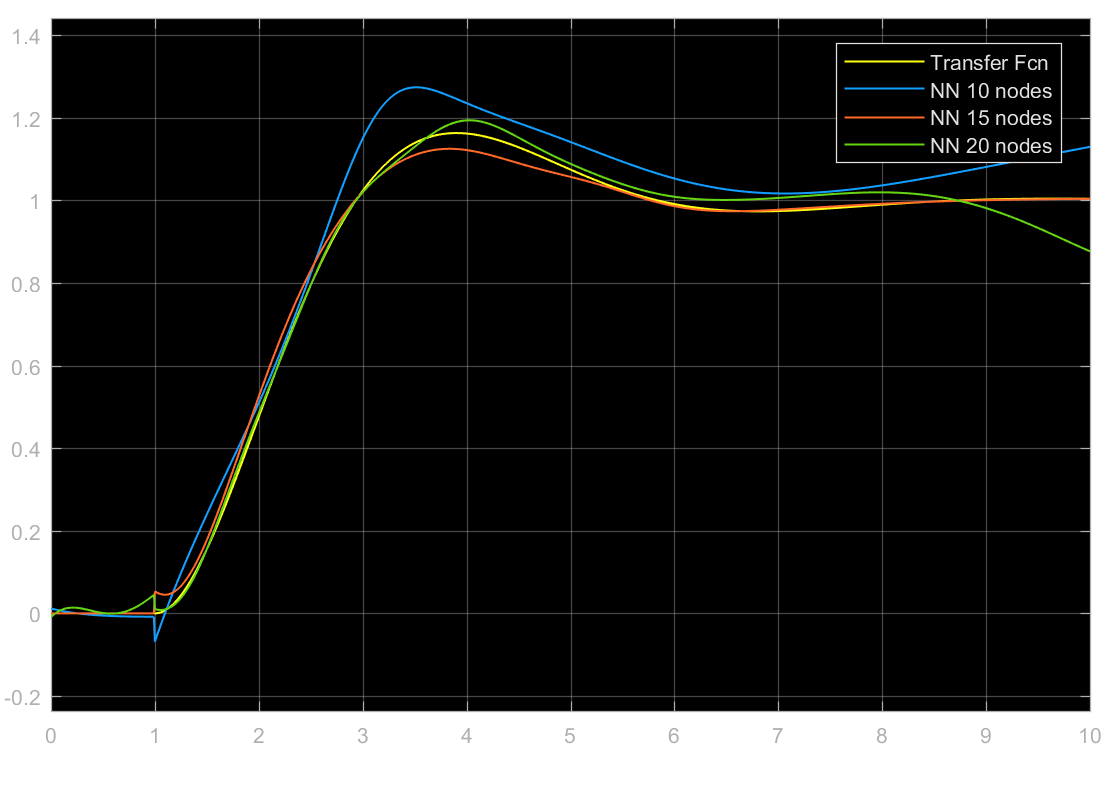
**Fig 1.14** **Comparison of NN with 10 nodes with original TF output when w=3 and z=1**



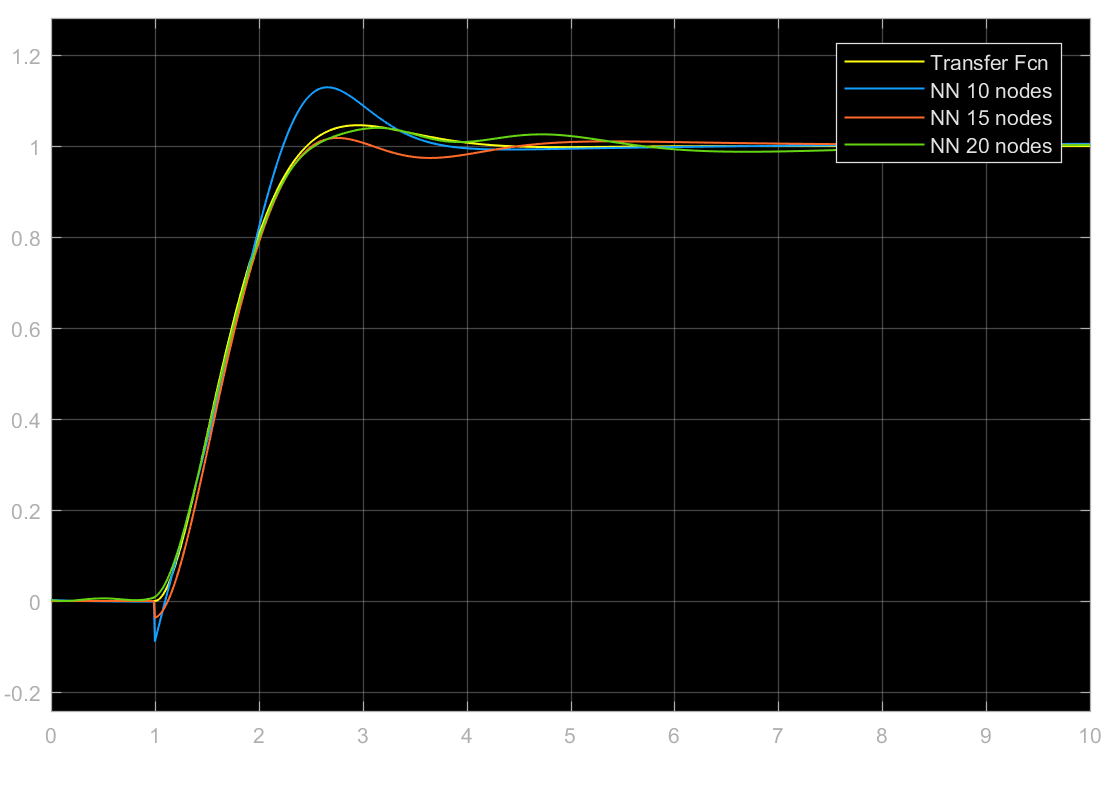
**Fig 1.15** **Comparison of NN with 15 nodes with original TF output when w=3 and z=1**



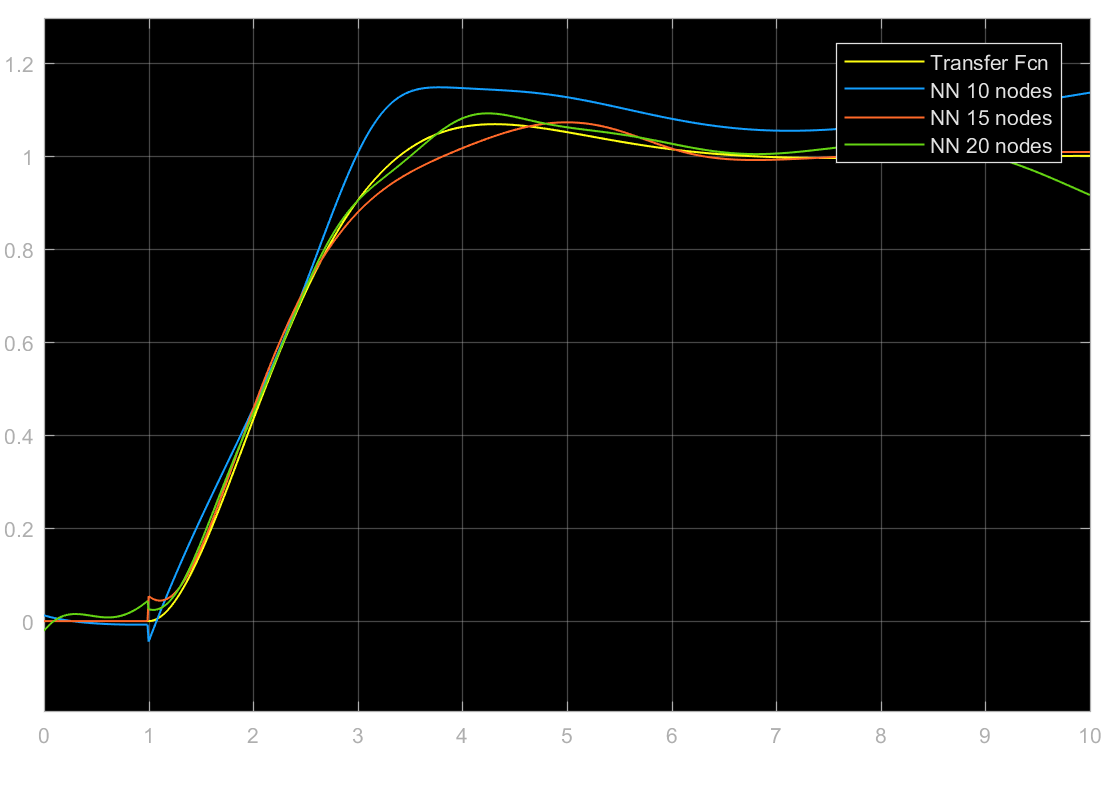
**Fig 1.16** **Comparison of NN with 20 nodes with original TF output when w=3 and z=1**



**Fig 1.17** **Comparison of NN outputs with original TF output when w=0.5 and z=1.25**



**Fig 1.18** **Comparison of NN outputs with original TF output when w=0.7 and z=2.25**



**Fig 1.19** **Comparison of NN outputs with original TF output when w=0.65 and z=1.25**

*Therefore, the neural network is able to approximate the function very well for larger values of w and also when the NN have larger number of nodes.*