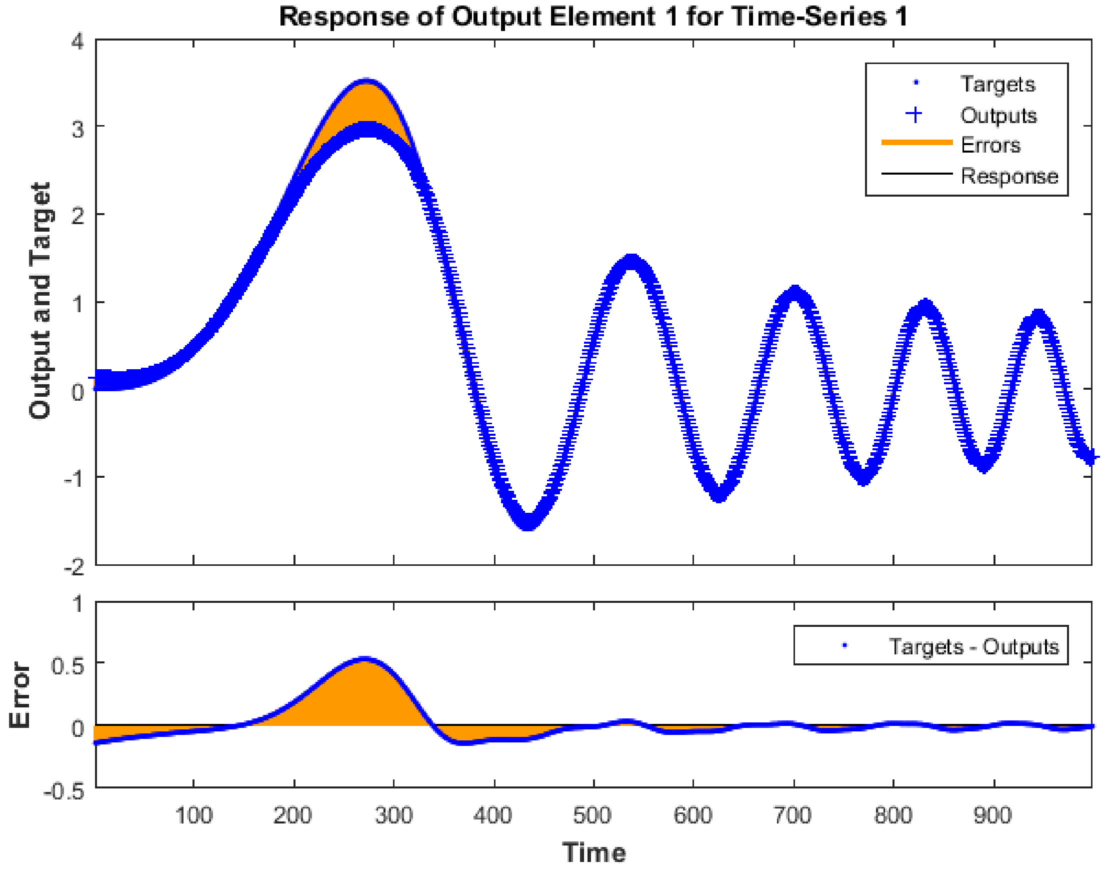
Transfer function	**Non-linear**	**linear**
Simulation time	10s	10s
Solver Options **- Type** **- Solver**	 Fixed Step auto	 Fixed Step auto
White noise **- Noise Power** **- Sample time** **- Seed**	 [0.01] 0.01 [23341]	 [0.01] 0.01 [23341]
Chirp **- Initial frequency (Hz)** **- Target time (secs)** **- Frequency at target time (Hz)**	 0.001 10 1	 0.001 10 1
NN	1:2, 2	1:2, 2
Epochs	250	250
Transfer Function	tansig (?)	purelin
Training:		
Stopped at iteration	250	37
Best Performance	1.1418e-07 (at 250 epoch)	3.6396e-19 (at 37 epoch)
Response of output error	-0.002459 – 0.00148 ($-2.459\text{x}10^{-3}$ - $1.48\text{x}10^{-3}$)	$-8.562\text{x}10^{-10}$ – $1.522\text{x}10^{-9}$
Testing:		
Response of output error	-0.1403 – 0.533	-0.1363 – 0.02471

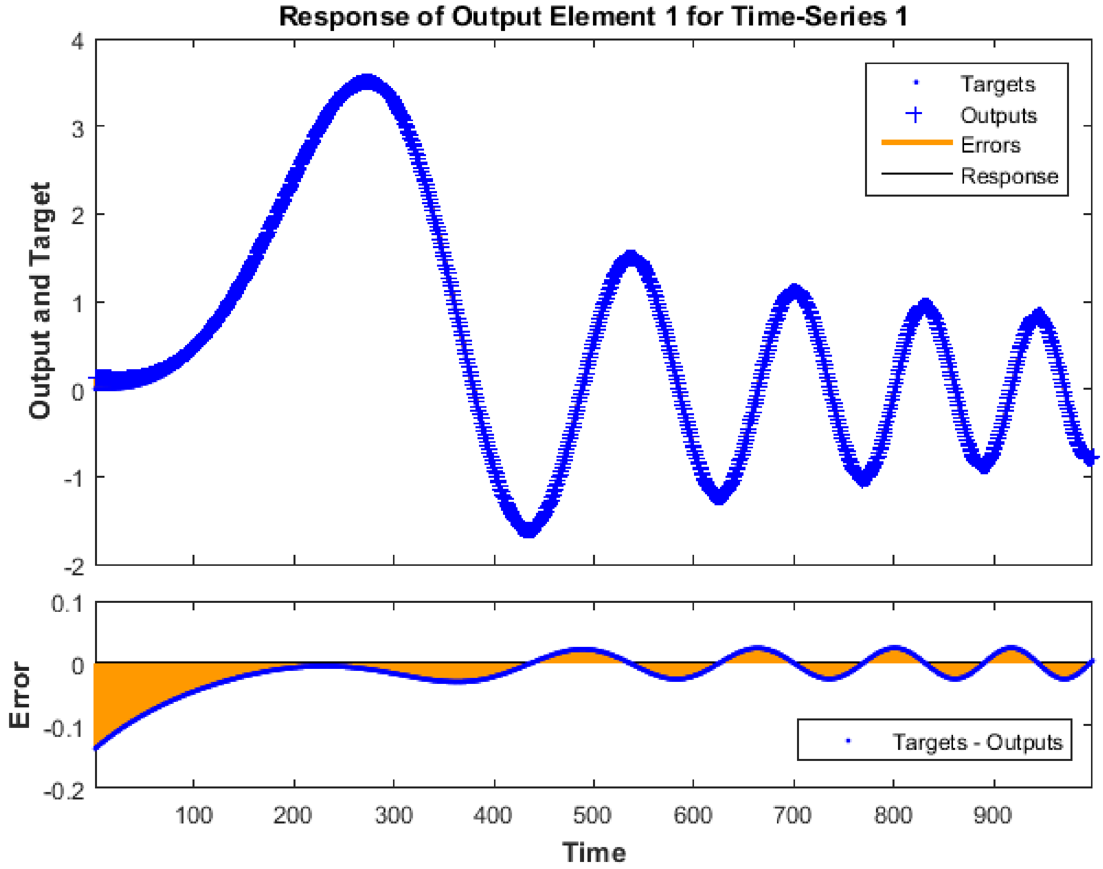
Linear/(nonLin) Code

```
%automatic save of plots and net
%training with white-noise and testing
with chirp

% ############ Prepare ##############
% view training signals
fig = figure('Name','Train In/Out');
subplot(2,1,1);
plot(input_signal);
% semilogx(input_signal);
% loglog(input_signal);
title('Input signal fed to the net');
subplot(2,1,2);
plot(output_signal);
% semilogx(output_signal);
% loglog(output_signal);
title('Output signal fed to the net');
% save figure
saveas(fig, 'net_trainInput', 'fig');

% prepare net
lrn_net = layrecnet(1:2, 2);
lrn_net.trainParam.show = 5;
lrn_net.trainParam.epochs = 250;
lrn_net.layers{1}.transferFcn =
'purelin';

% prepare signals
X = con2seq(input_signal');
T = con2seq(output_signal');
Xc = con2seq(input_chirp');
Tc = con2seq(output_chirp');
[Xs,Xi,Ai,Ts] = preparets(lrn_net,X,T);
[Xcs,Xci,Aci,Tcs] =
preparets(lrn_net,Xc,Tc);

% ############# Training #############
% train net
[lrn_net, tr] =
train(lrn_net,Xs,Ts,Xi,Ai);

% save various plots from training
% performance
fig = figure(2);
plotperform(tr);
saveas(fig,'train_plotperform','fig');

% training state
fig = figure(3);
plottrainstate(tr);
saveas(fig,'train_plottrainstate','fig')
;

% error histogramm
Y = lrn_net(Xs,Xi,Ai);
e = cell2mat(Ts) - cell2mat(Y);
fig = figure(4);
ploterrhist(e,'bins',20);
saveas(fig,'train_ploterrhist','fig');

% regression
fig = figure(5);
plotregression(Ts, Y, 'Training');
saveas(fig,'train_plotregression',
'fig');

% response
fig = figure(6);
plotresponse(Ts, Y);
saveas(fig,'train_plotresponse','fig');

% save trained net
save TP_net.mat lrn_net;

% view net output
Y = lrn_net(Xs,Xi,Ai);
fig = figure('Name','Net Train Output');
plot(cell2mat(Y));
title('Output signal from the net');
% save figure
saveas(fig,'net_trainOutput','fig');

% ############## Testing ##############
% view and save testing input
fig = figure('Name','Test Input');
plot(input_chirp);
title('Input signal fed to the net');
% save figure
saveas(fig,'net_testInput','fig');

% fed testing signal to net
Yc = lrn_net(Xcs,Xci,Aci);

% view test output signals
fig = figure('Name','Test Output');
subplot(2,1,1);
% semilogx(output_chirp);
plot(output_chirp);
title('Original output signal');
subplot(2,1,2);
% semilogx(cell2mat(Yc));
plot(cell2mat(Yc));
title('Net output signal');
% save figure
saveas(fig,'net_testOnput','fig');
% view output errors
fig = figure(10);
plotresponse(Tcs,Yc);
% save figure
saveas(fig,'test_plotresponse','fig');
```

Non-Lin:


Response of Output Element 1 for Time-Series 1
Output and Target
4
3
2
1
0
-1
-2
Targets
Outputs
Errors
Response
Error
1
0.5
0
-0.5
Targets - Outputs
100
200
300
400
500
600
700
800
900
Time

Lin:


Response of Output Element 1 for Time-Series 1
Output and Target
4
3
2
1
0
-1
-2
Targets
Outputs
Errors
Response
Error
0.1
0
-0.1
-0.2
Targets - Outputs
100
200
300
400
500
600
700
800
900
Time