Scientific Research Publishing

Engineering, 2019, 11, 231-239
http://www.scirp.org/journal/eng
ISSN Online: 1947-394X
ISSN Print: 1947-3931

Analysis and Interpretation of Steering Geometry of Automobile Using Artificial Neural Network Simulation

Pramod N. Belkhode

Department of General Engineering, Laxminarayan Institute of Technology, Nagpur, India
Email: pramodb@rediffmail.com

How to cite this paper: Belkhode, P.N. (2019) Analysis and Interpretation of Steering Geometry of Automobile Using Artificial Neural Network Simulation. *Engineering*, **11**, 231-239.
https://doi.org/10.4236/eng.2019.114016

Received: January 27, 2019
Accepted: April 8, 2019
Published: April 11, 2019

Abstract

Vehicle dynamics is the one of the most important factors in the analysis and predicting the steering behavior of automobile. The paper details the evaluation of the Artificial Neural Network (ANN) structures to estimate the steering geometry parameters of four wheel vehicle. One of the aspects of vehicle performance is performance of steering geometry. Steering geometry parameters kingpin inclination angle, caster angle, camber angle, toe angle, scrub radius, toe in and toe out are measured using alignment techniques and caster/camber gauges. Suspension system components pivot upon a rubber bushing which is compressed between an inner and outer metal sleeve. Excess clearance developed in the joints of suspension system in turn causes changes in steering geometry. This is obviously essential for any automobile for a major challenge in terms of operation, performance, servicing and maintenance. ANN models applicable to each of these steering parameters were developed. Steering geometry is evaluated through the independent and dependent variables of front suspension. Dependent variables such as steering geometry parameters kingpin inclination angle, caster angle, camber angle, toe angle, scrub radius, toe in and toe out are determined with the help of independent variables. These dependent variables are validated through ANN simulation. The result obtained through ANN is in close agreement to the experimental observation.

Keywords

Steering Parameters, ANN, Mathematical Model, Automobile, Suspension

1. Introduction

The present methods of observing the steering parameters are not suitable and have limitation in the measurements and predicting the behavior of front suspension of an automobile [1] [2]. Steering geometry parameters kingpin inclination angle, caster angle, camber angle, toe angle, scrub radius, toe in and toe out are measured using alignment techniques and caster/camber gauges. Steering parameters change from place to place. The analysis of parameters requires a mathematical model which can usefully to the observed variation and which then provides a basis for generalization, prediction and interpretation. Steering behavior is predicated by the by the experimental investigation. Artificial Neural Network (ANN) is generally the software systems that imitate the neural network of the human brain [3]. The complex relationship between the input and output is identifying by the powerful tool of neural networks. The study indicates that the expert systems such as ANN are efficient in simulating the complicated phenomena due to its non-linear structures [4]. The objectives of this study were to evaluate the accuracy of ANN for estimation of steering parameters. Artificial Neural Network technique is recently used in the entire field to evaluate the experimental or field data. Network is trained with known inputs and outputs. Once network is trained output is predicated based on the new inputs. Paper details the validation of the experimental data with the help of Artificial Neural Network.

2. Steering Geometry

Joint O1 and O2 are revolute joints and joints A and B are spherical joints as shown in Figure 1. The relative orientation of two links connected at joint can be recorded in terms of value of the angles measured by potentiometer and using electronic instrumentation. At four joints (two spherical and two revolute) of the RSSR mechanism six potentiometers are located. At revolute joints O1 & O2 the one included angle each of these joints and at spherical joints A & B the two included angles at each of these joints. Once these angles are measured and position of linkage of front suspension is decided, position of kingpin axis can be located. The included angles at the joints of front suspension mechanism are first decided by potentiometers. These measured angles are supplied to interfacing program which calculates the steering performance parameters such as Kingpin angle, Camber angle, Caster angle, Toe angle, Toe in, Toe out, Scrub radius. The experimental setup is formulated on which trial are recoded with varying speed

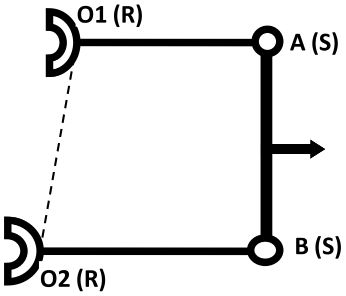

Figure 1. Front suspension of an automobile.

and breakers height. The steering geometry parameters such as link lengths, clearance at the joints, joints angles, breakers height, velocity and wheel diameter are recorded with the help of measuring instruments. Joints angles are measured by the potentiometer and position are joint A and B is located. Position of joint A and B further decided the position of kingpin inclination. Kingpin inclination is used for finding the steering geometry such Kingpin angle, Camber angle, Caster angle, Toe angle, Toe in, Toe out, Scrub radius.

3. Estimation of Performance Parameters

Table 1 shows the comparison of values calculated as per experimental observations and ANN simulations [5] [6]. The term such as π_{D1}, π_{D2}, π_{D3}, π_{D4}, π_{D5}, π_{D6}, π_{D7} indicates angles of Kingpin, Camber, Caster, Toe, Toe in, Toe out, Scrub radius respectively.

4. Procedure for Formulation of ANN Model

The experimental data based modeling has been achieved based on experimental data for the seven dependent pi terms. In such complex phenomenon involving non-linear kinematics where in the validation of experimental data based models

Table 1. Values computed by experimental observation and ANN simulation.

Sr.N	Values calculated as per experimental observations							Values calculated as per ANN simulation						
	Π_{D1}	Π_{D2}	Π_{D3}	Π_{D4}	Π_{D5}	Π_{D6}	Π_{D7}	Π_{D1}	Π_{D2}	Π_{D3}	Π_{D4}	Π_{D5}	Π_{D6}	Π_{D7}
1	6.31	0.24	1.17	0.24	59.89	120	120	6.31	0.24	1.10	0.23	59.89	120.07	119.97
2	6.65	0.61	2.10	0.54	59.73	119	121	6.65	0.61	2.01	0.54	59.73	119.21	120.91
3	6.80	1.30	4.20	0.91	59.44	119	121	6.8	1.30	4.08	0.90	59.44	119.11	121.00
4	6.99	2.08	6.29	1.12	59.10	118	122	6.99	2.08	6.30	1.09	59.10	118.01	122.00
5	7.20	2.12	7.44	1.31	59.09	116	122	7.20	2.12	7.44	1.11	59.09	118.00	122.00
6	7.47	2.87	8.59	1.23	58.77	118	122	7.47	2.87	8.58	1.20	58.77	118.00	121.98
7	5.93	0.46	1.87	0.43	59.80	120	120	5.93	0.46	1.84	0.43	59.79	120.01	119.99
8	6.61	1.25	2.80	0.89	69.46	119	121	6.61	1.25	2.79	0.87	59.46	119.03	120.98
9	7.03	1.62	4.90	1.02	59.30	119	121	7.03	1.62	4.90	1.00	59.30	118.99	120.99
10	7.20	2.16	6.75	1.13	59.07	118	122	7.20	2.16	6.77	1.09	59.07	117.99	121.99
11	7.63	2.27	7.44	1.15	59.02	118	122	7.65	2.26	7.52	1.13	59.02	117.99	122.02
12	7.81	3.08	8.82	1.25	58.67	117	123	7.85	3.08	8.83	1.24	58.67	117.00	123.02
13	6.38	0.79	1.17	0.67	59.66	119	121	6.39	0.79	1.15	0.69	59.66	118.97	120.99
14	6.98	1.89	3.50	1.08	59.19	118	122	6.98	1.88	3.49	1.06	59.18	117.97	121.99
15	7.52	2.32	5.82	1.16	59.00	118	122	7.52	2.31	5.81	1.14	59.00	118.06	121.99
16	7.71	2.48	6.98	1.18	58.93	118	122	7.74	2.48	6.99	1.18	58.93	118.02	122.00
17	8.02	2.79	8.59	1.22	58.80	118	122	8.06	2.79	8.66	1.22	58.81	117.99	122.00
18	8.74	3.24	9.27	1.27	58.61	117	123	8.71	3.24	9.42	1.26	58.62	117.01	122.99

is not in close proximity, it becomes necessary to formulate Artificial Neural Network (ANN) Simulation of the observed data. Simulation consists of three layers. First layer is known as input layer. The input neurons in input layer are equal to the number of independent variables. Second layer is known as hidden layer. It consists of seven numbers of neurons. The third layer is output layer. It contains one neuron as one of dependent variables at a time. For the ANN multilayer feed forward topology is decided.

MATLAB software is selected for developing ANN simulation. The following steps are involved for developing the ANN algorithm is as under.

- The experimental data is separated into two parts viz. input data and the output data pi terms. The input data and output data are imported to the program respectively.
- The prestd function is used to read the input and output data and appropriately sized.
- The input and output data is normalized in preprocessing step using mean and standard deviation.
- The input and output data is then categorized in three categories viz. testing, validation and training. From the 18 observations, initial 75% of the observations is selected for training, last 75% data for validation and middle overlapping 50% data for testing.
- The data is then stored in structures for training, testing and validation.
- The feed forward back propagation is selected based on the data.
- Using the training data the network is then trained. The actual data and target data are compared and simulate the network.

The regression analysis and the representation are done through the standard functions. The values of regression coefficient and the equation of regression lines are represented on the seven different graphs plotted for the seven dependent pi terms [7] [8]. The detailed ANN program used for evaluation the steering geometry is provided in the **Appendix**.

ANN Model

Figure 1 shows the structure and basic elements for designing artificial neural network. The capability of the ANN model is to generalize unseen data dependent on several factors. These factors are appropriate selection of input output parameters, the distribution of the input and output dataset and the format of the presentation of the dataset to the neural network as shown in **Figure 2**. The output parameters of the model is then Kingpin angle, Camber angle, Caster angle, Toe angle, Toe in, Toe out, Scrub radius. Details of the input and output parameters of the proposed ANN model are illustrated. MATLAB is used for training the network architecture. **Figure 3** shows the optimal network architecture is formed by training. The network is trained with the help experimental results shown in the **Table 1**. The experimental results are imported to the trained network of ANN program as shown in the **Figure 3** ANN Topology. ANN

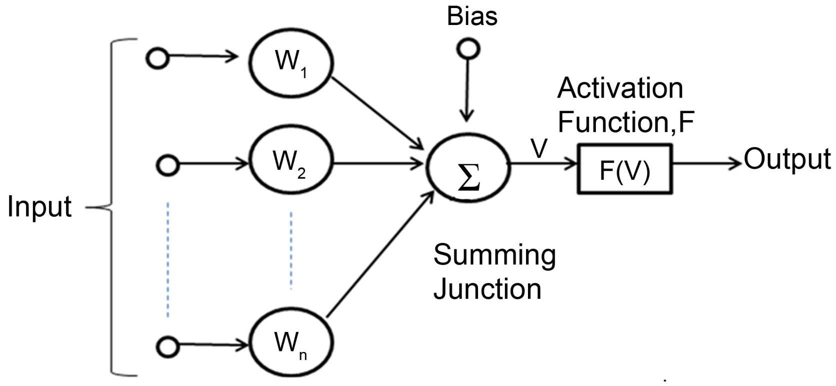

Figure 2. ANN neurons with its elements [3] [4].

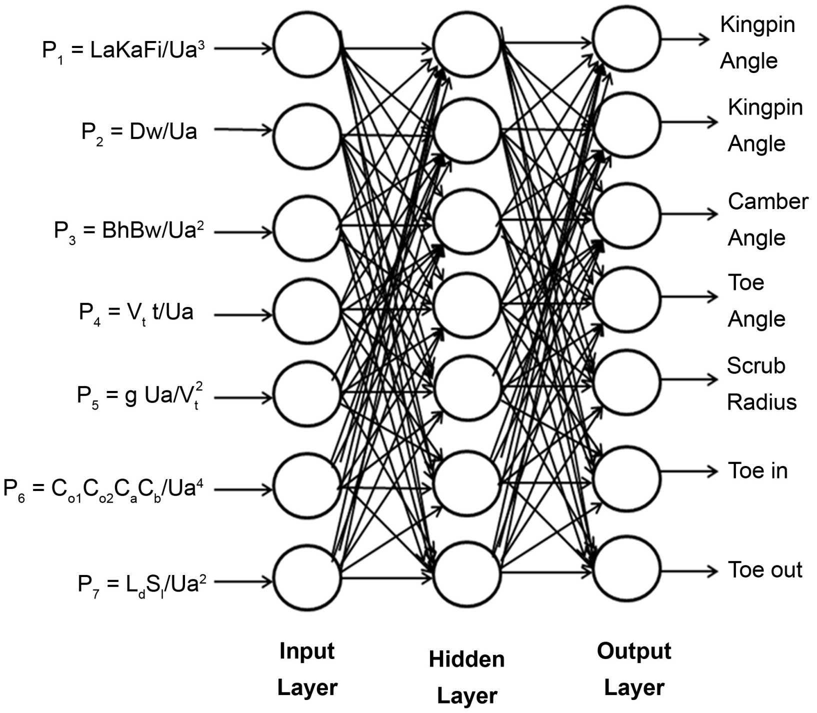

Figure 3. ANN topology.

program shown in Appendix is run on the MATLAB software. The ANN Outputs consists of all the steering parameters are shown in the Table 1.

5. Conclusion

An ANN model has been developed for predicating steering behavior. The model was proved to be successful in terms of agreement with actual values for experimentation. The feasibility and rationality of the ANN Model of the testing data which includes all the steering geometry Kingpin angle, Camber angle, Caster angle, Toe angle, Toe in, Toe out, Scrub radius is proved to be in close agreement. Table 1 shows the comparison of the steering geometry parameters evaluated through the experimental observation and ANN Program. It can be concluded that ANN model performs accurately to determine the optimal values.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Gillespie, T.D. (1992) Fundamentals of Vehicle Dynamics. Society of Automotive Engineers Inc., Warrendale.

[2] Suh and Redcliff (1978) Kinematics Design of Mechanisms. John Wiley and Sons, New York.

[3] Sivanandam, S.N., Sumathi, S. and Deepa, S.N. (2006) Introduction to Neural Networks Using Matalb 6.0. Tata Mcgraw-Hill Publishing Company Limited, New Delhi.

[4] Kartalopoulos, S.V. (2004) Understanding Neural Networks and Fuzzy Logic—Basic Concepts and Applications. Publication Prentice Hall of India Private Limited, New Delhi.

[5] Schenck Jr., H. (1961) Theory of Engineering Experimentation. Mc Graw Hill.

[6] Belkhode, P.N. and Modak, J.P. (2011) Comparison of Steering Geometry Parameter of Front Suspension of Automobile. *Internal Journal of Scientific and Engineering Research*, **3**, 1-3.

[7] Belkhode, P.N. and Modak, J.P. (2007) Kinematic Analysis of Front Suspension of an Automobile and Steering Behaviour. *Proceedings of* 12*th World Congress in Mechanism and Machine Science*, Besancon, 17-21 June 2007.

[8] Belkhode, P.N. (2017) Mathematical Modelling of Liner Piston Maintenance Activity Using Field Data to Minimize Overhauling Time and Human Energy Consumption. *Journal of the Institution of Engineers* (*India*): *Series C*, Springer, 1-9.

Appendix

ANN Program

```
clear all;
close all;
inputs3=[
]
  a1=inputs3
input_data=a1;
  output3=[
]
  y1=output3
size(a1);
size(y1);
p=a1';
sizep=size(p);
t=y1';
sizet=size(t);
[S Q]=size(t)
[pn,meanp,stdp,tn,meant,stdt] = prestd(p,t);
  net = newff(minmax(pn),[18 1],{'logsig' 'purelin'},'trainlm');
net.performFcn='mse';
net.trainParam.goal=.01;
net.trainParam.show=200;
net.trainParam.epochs=50;
net.trainParam.mc=0.05;
net = train(net,pn,tn);
an = sim(net,pn);
[a] = poststd(an,meant,stdt);
error=t-a;
x1=1:18;
plot(x1,t,'rs-',x1,a,'b-')
legend('Experimental','Neural');
title('Output (Red) and Neural Network Prediction (Blue) Plot');
xlabel('Experiment No.');
ylabel('Output');
grid on;
figure
error_percentage=100*error./t
plot(x1,error_percentage)
legend('percentage error');
axis([0 18 -100 100]);
title('Percentage Error Plot in Neural Network Prediction');
xlabel('Experiment No.');
ylabel('Error in %');
```

```
grid on;
for ii=1:18
xx1=input_data(ii,1);
yy2=input_data(ii,2);
zz3=input_data(ii,3);
xx4=input_data(ii,4);
yy5=input_data(ii,5);
zz6=input_data(ii,6);
xx7=input_data(ii,7);
pause
yyy(1,ii)
yy_practical(ii)=(y2(ii,1));
yy_eqn(ii)=(yyy(1,ii))
yy_neur(ii)=(a(1,ii))
yy_practical_abs(ii)=(y2(ii,1));
yy_eqn_abs(ii)=(yyy(1,ii));
yy_neur_abs(ii)=(a(1,ii));
pause
end
  figure;
plot(x1,yy_practical_abs,'r-',x1,yy_eqn_abs,'b-',x1,yy_neur_abs,'k-');
legend('Practical','Equation','Neural');
title('Comparision between practical data, equation based data and neural based data');
xlabel('Experimental');
figure;
plot(x1,yy_practical_abs,'r-',x1,yy_eqn_abs,'b-');
legend('Practical’,’ Equation');
title('Comparision between practical data, equation based data and neural based data');
xlabel('Experimental');
figure;
plot(x1,yy_practical_abs,'r-',x1,yy_neur_abs,'k-');
legend('Practical','Neural');
title('Comparision between practical data, equation based data and neural based data');
xlabel('Experimental');
error1=yy_practical_abs-yy_eqn_abs
figure
error_percentage1=100*error1./yy_practical_abs;
plot(x1,error_percentage,'k-',x1,error_percentage1,'b-');
legend('Neural','Equation');
axis([0 100 -100 100]);
title('Percentage Error Plot in Equation (blue), Neural Network (black) Predic-
tion');
xlabel('Experiment No.');
ylabel('Error in %');
meanexp=mean(output3)
meanann=mean(a)
meanmath=mean(yy_eqn_abs)
mean_absolute_error_performance_function = mae(error)
mean_squared_error_performance_function = mse(error)
net = newff(minmax(pn),[18 1],{'logsig' 'purelin'},'trainlm','learngdm','msereg');
an = sim(net,pn);
[a] = poststd(an,meant,stdt);
error=t(1,[1:18])-a(1,[1:18]);
net.performParam.ratio = 20/(20+1);
perf = msereg(error,net)
 rand('seed',1.818490882E9)
[ps] = minmax(p);
[ts] = minmax(t);
numInputs = size(p,1);
numHiddenNeurons = 18;
numOutputs = size(t,1);
net = newff(minmax(p), [numHiddenNeurons,numOutputs]);
[pn,meanp,stdp,tn,meant,stdt] = prestd(p,t);
[ptrans,transmit]=prepca(pn,0.001);
[R Q]=size(ptrans);
testSamples= 6:1:Q;
validateSamples=15:1:Q;
trainSamples= 1:1:Q;
validation.P=ptrans(:,validateSamples) ;
validation.T=tn(:,validateSamples) ;
testing.P= ptrans(:,testSamples) ;
testing.T= tn(:,testSamples)
ptr= ptrans(:,trainSamples) ;
ttr= tn(:,trainSamples);
net = newff(minmax(ptr),[18 1],{'logsig' 'purelin'},'trainlm');
 [net,tr] = train(net,ptr,ttr,[] ,[],validation,testing);
plot(tr.epoch,tr.perf, 'r',tr.epoch,tr.vperf, 'g',tr.epoch,tr.tperf, 'h') ;
legend('Training', 'validation', 'Testing',-1) ;
ylabel('Error') ;
an=sim(net,ptrans);
a=poststd(an,meant,stdt);
pause;
figure
[m,b,r] = postreg(a,t);s
```