ATLANTIS PRESS

Advances in Engineering Research, volume 123

2nd International Conference on Materials Science, Machinery and Energy Engineering (MSMEE 2017)

Simulation Analysis of Automobile ABS based on Artificial Neural Network

Banglu Chen

College of Automobile Engineering, Wuxi Institute of Commerce, Wuxi, Jiangsu ,China, 214153

email:cblu@sohu.com

Keywords： ANN; ABS; the relative slip ratio

Abstract. An Artificial Neural Network (ANN) was developed to analyze the automobile Anti-locking Brake System (ABS) simulation model. The linear velocity of wheel and the angular velocity could be obtained through some applications, through them we could calculate the relative slip ratio, the relative slip ratio and the braking time were input variables, the braking press and the braking distance were output. The data used to train and verify the ANN were obtained from simulation models. Trained ANN can be used to predict output.

Introduction

Vehicle safety question is always the matter people concentrate on. But many accidents have something with braking, so ABS was developed to avoid these. The system adopted the braking press with the change of the relative slip ratio and the braking time to avoid the wheel locking.

Artificial Neural Network (ANN) is a data processing system based on the structure of biological neural system. Analysis with ANN is not like modeling and simulation, But by learning from the data generated experimentally or simulation models, ANN differ from conventional programs in their ability to learn about the system to be modeled without priori knowledge of the process variables relationship.

The objectives of this paper was to develop an ANN to analyze the simulation model using the simulation data, so that corrections can be made in braking process. This will assist in developing better process controls.

Model Development

Mathematic model of vehicle braking system include three components: the whole vehicle model, tire—road braking force model, brake model. For the convenience to study, I used a quarter of whole vehicle model.

The Whole Vehicle Model. In this model, the drag forces were neglected. I got the equations as following.

$$M\dot{v} = -F_x$$

$$I\dot{\omega} = RF_x - T_b$$

Where M is a quarter of the vehicle mass, v is the linear velocity of the wheel, F_x is the road braking force, I is the wheel moment of inertia, ω is the angular velocity, R is the radius of the wheel, T_b is the braking moment.

Tire—Road Braking Force Model. The road braking force is decided by friction coefficient between tire and road (φ_b) and vertical force between tire and road (F_z). The friction coefficient is the function of the wheel slip ratio (S).

$$F_x = F_z \cdot \varphi_b$$

$$\varphi_b = f_{(S)}$$

Brake Model. Braking moment has something with brake's size, brake hose press and the friction coefficient of friction material.

$$T_b = (p_l - p_0)A_c \eta_m (BF) r\rho$$

Where p_l is the press in the brake chamber, p_0 is the press enough to conquest the drag force of the coil, A_c is the area of brake chamber, η_m is the mechanical efficiency of the brake, r is the radius of the brake drum, ρ is the ratio of the brake chamber and the brake shoe.

Simulation model is shown as Fig.1.

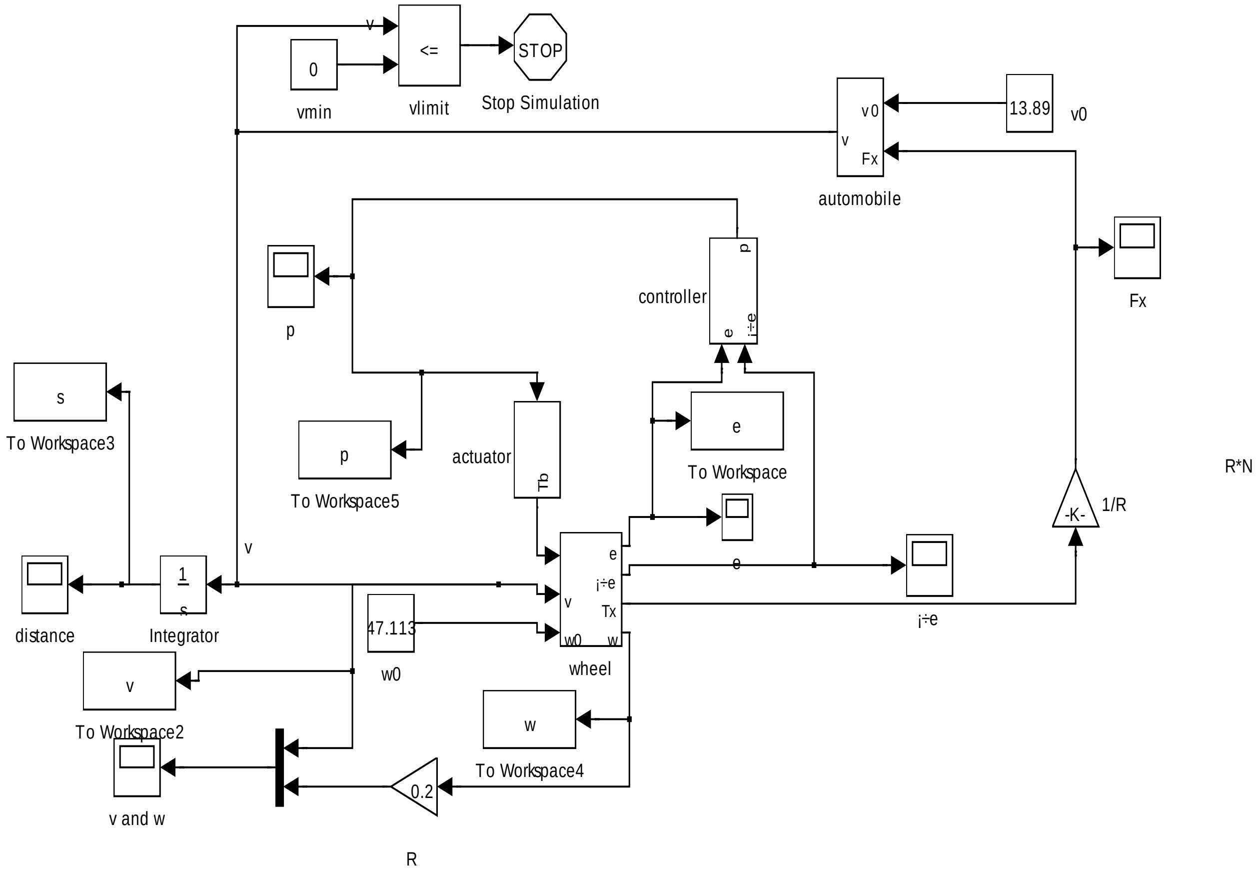

Fig.1 Simulation model

Data Generation

In the simulation model, I set the vehicle speed 50km/h, initial press 1200000pa, working time 0.001s. The data is shown as Table 1.

Table 1. The training and verifying data

W	*V*	*E*	*T*	*P*	*L*
47.113	13.89	-0.19992	0	1785720	0
37.97373	13.77702	-0.01256	5.76E-06	974029	0.263031
37.60694	13.63766	-0.01293	3.46E-05	974029	0.52347
37.22867	13.49847	-0.01306	0.000179	974029	0.781264
36.84643	13.35934	-0.01309	0.000898	974029	1.036413
36.46311	13.22023	-0.0131	0.001	974029	1.288919
36.07952	13.08112	-0.0131	0.001	974029	1.538782
35.69586	12.942	-0.0131	0.001	974029	1.786001
35.31218	12.80289	-0.0131	0.001411	974029	2.030578
34.92849	12.66378	-0.0131	0.001823	974029	2.272511
34.54481	12.52467	-0.0131	0.002234	974029	2.511802
34.16112	12.38556	-0.0131	0.00244	974029	2.748449
33.77744	12.24645	-0.0131	0.002491	974029	2.982453
33.39375	12.10734	-0.0131	0.002498	974029	3.213814
33.01007	11.96823	-0.0131	0.002499	974029	3.442532

32.62638	11.82912	-0.0131	0.0025	974029	3.668607
32.2427	11.69001	-0.0131	0.0025	974029	3.892039
31.85901	11.5509	-0.0131	0.0025	974029	4.112827
31.47532	11.41179	-0.0131	0.0025	974029	4.330973
31.09164	11.27268	-0.0131	0.0025	974029	4.546475
30.70795	11.13357	-0.0131	0.0025	974029	4.759335
30.32427	10.99446	-0.0131	0.0025	974029	4.969551
29.94058	10.85535	-0.0131	0.0025	974029	5.177124
29.5569	10.71624	-0.0131	0.0025	974029	5.382054
29.17321	10.57713	-0.0131	0.0025	974029	5.584341
28.78952	10.43802	-0.0131	0.0025	974029	5.783985
28.40584	10.29891	-0.0131	0.0025	974029	5.980986
28.02215	10.1598	-0.0131	0.0025	974029	6.175344
27.63847	10.02069	-0.0131	0.0025	974029	6.367059
27.25478	9.881578	-0.0131	0.0025	974029	6.55613
26.8711	9.742468	-0.0131	0.0025	974029	6.742558
26.48741	9.603358	-0.0131	0.002501	974029	6.926344
26.10372	9.464248	-0.0131	0.002502	974029	7.107486
25.72004	9.325137	-0.0131	0.002504	974029	7.285985
25.33635	9.186027	-0.0131	0.002508	974029	7.461841
24.95267	9.046917	-0.0131	0.002515	974029	7.635054
24.56898	8.907806	-0.0131	0.00253	974029	7.805624
24.1853	8.768696	-0.0131	0.00256	974029	7.973551
23.80161	8.629586	-0.0131	0.00262	974029	8.138835
23.41792	8.490476	-0.0131	0.002682	974029	8.301475
23.03424	8.351365	-0.0131	0.002814	974029	8.461473
22.65055	8.212255	-0.0131	0.002946	974029	8.618827
22.26687	8.073145	-0.0131	0.003076	974029	8.773538
21.88318	7.934034	-0.0131	0.003184	974029	8.925607
21.4995	7.794924	-0.0131	0.003417	974029	9.075032
21.11581	7.655814	-0.0131	0.003476	974029	9.221814
20.73212	7.516704	-0.0131	0.003491	974029	9.365953
20.34844	7.377593	-0.0131	0.003503	974029	9.507448
19.96475	7.238483	-0.0131	0.003503	974029	9.646301
19.58107	7.099373	-0.0131	0.003518	974029	9.782511
19.19738	6.960263	-0.0131	0.003533	974029	9.916077
18.8137	6.821152	-0.0131	0.003564	974029	10.047
18.43001	6.682042	-0.0131	0.003624	974029	10.17528
18.04632	6.542932	-0.0131	0.003745	974029	10.30092
17.66264	6.403821	-0.0131	0.003988	974029	10.42391
17.27895	6.264711	-0.0131	0.004473	974029	10.54426
16.89527	6.125601	-0.0131	0.005193	974029	10.66197
16.51158	5.986491	-0.0131	0.005914	974029	10.77704
16.1279	5.84738	-0.0131	0.006621	974029	10.88946
15.74421	5.70827	-0.0131	0.008962	974029	10.99924
15.36053	5.56916	-0.0131	0.009644	974029	11.10637
14.97684	5.43005	-0.0131	0.010326	974029	11.21086
14.59315	5.290939	-0.0131	0.011083	974029	11.31271
14.20947	5.151829	-0.0131	0.013295	974029	11.41192
13.82578	5.012719	-0.0131	0.015506	974029	11.50848
13.4421	4.873608	-0.0131	0.019	974029	11.6024

13.05841	4.734498	-0.0131	0.034497	974029	11.69368
12.67473	4.595388	-0.0131	0.038	974029	11.78231
12.29104	4.456278	-0.0131	0.057	974029	11.86831
11.90735	4.317167	-0.0131	0.076	974029	11.95165
11.52367	4.178057	-0.0131	0.095	974029	12.03236
11.13998	4.038947	-0.0131	0.114	974029	12.11042
10.7563	3.899837	-0.0131	0.133	974029	12.18584
10.37261	3.760726	-0.0131	0.152	974029	12.25861
9.988926	3.621616	-0.0131	0.171	974029	12.32875
9.605241	3.482506	-0.0131	0.19	974029	12.39623
9.221549	3.343396	-0.0131	0.209	974029	12.46108
8.837921	3.204284	-0.0131	0.228	974029	12.52328
8.453574	3.065184	-0.01304	0.247	974029	12.58284
8.078746	2.925943	-0.01396	0.266	974029	12.63976
7.701454	2.786738	-0.01471	0.285	974029	12.69403
7.302954	2.647847	-0.01308	0.304	974029	12.74566
6.923327	2.508676	-0.01358	0.323	974029	12.79465
6.558581	2.369287	-0.01606	0.342	974029	12.841
6.149757	2.230547	-0.01278	0.361	974029	12.8847
5.768603	2.0914	-0.01313	0.38	974029	12.92576
5.009706	1.813054	-0.01457	0.418	974029	12.99994
4.617313	1.674072	-0.0131	0.437	974029	13.03307
4.235045	1.534941	-0.01338	0.456	974029	13.06356
3.850629	1.395842	-0.01325	0.475	974029	13.0914
3.466427	1.256739	-0.01314	0.494	974029	13.1166
3.074026	1.117757	-0.01075	0.513	974029	13.13916
2.695431	0.978572	-0.01201	0.532	974029	13.15907
2.307435	0.839525	-0.01026	0.551	974029	13.17634
1.938964	0.70019	-0.01636	0.57	974029	13.19097
1.163485	0.42209	-0.01261	0.608	974029	13.21229
0.782539	0.282939	-0.01534	0.627	974029	13.21899
0.042244	0.012761	-0.1759	0.665	1785720	13.22446

Network Training and Testing

From the generated data (98 sets), 14 and 14 data sets were randomly selected as testing and production sets, rest 70 data sets were used for ANN training. Testing data were fed to test trained ANN after training 200 epochs. Testing errors were recorded. In the beginning of training, testing error decreased with training process. Training was continued until testing error did not decrease. If testing error was greater than the minimum testing error, the testing was continued until 200000 epochs training was reached. After training, 14 production sets were used to verify ANN performance.

Hidden Layers and Nodes. Decision of the use of number of hidden layer nodes is complex as it depends on the specific problem being solved using ANN. With too few nodes, the network may not be powerful enough for a given task. With a large number of nodes (and connections), computation is too lengthy. Sometimes an ANN may memorize the input training samples, such a network tends to perform poorly on new test samples, and is not considered to have completed learning successfully. Neural learning is considered successfully only if the system can perform well on test data which the system has not been trained. An ANN should have capabilities to generalize from input training sets, and not to memorize them. In my paper, learning rate and momentum were set 0.5, initial weights were set 0.3, the hidden nods were 10, then to chose the best number of hidden layers. The result was got as Table 2.

Table 2. Effect of numbers of hidden layers on ABS simulation errors

The number of hidden layers	1		2		3	
Output:	*P*	*L*	*P*	*L*	*P*	*L*
R squared:	1	0.9093	1	0.9324	0.9999	0.9332
r squared:	1	0.9129	1	0.9345	1	0.9356
Mean squared error:	5924.412	1.431	645160.1	1.067	764164.8	1.055
Mean absolute error:	51.304	0.877	344.438	0.72	472.365	0.677
Min. absolute error:	0	0	0	0	0	0.004
Max. absolute error:	284.25	3.367	3597.688	3.117	3832.25	3.194
Correlation coefficient *r*:	1	0.9554	1	0.9667	1	0.9673

Comparing with each other and analyzing the errors, I found that the number of hidden layer was better than any others.

After choosing the best number of hidden layer, I would choose the best nodes of hidden layer. Learning rate and momentum were set 0.5,initial weights were set 0.3, I got the result as Table 3.

Table 3. Effect of nodes in hidden layer on ABS simulation errors

Nodes of hidden layer	10		15		13	
Output:	*P*	*L*	*P*	*L*	*P*	*L*
R squared:	1	0.9093	1	0.9111	1	0.9235
r squared:	1	0.9129	1	0.9173	1	0.9276
Mean squared error:	5924.412	1.431	91631.67	1.403	225454.6	1.207
Mean absolute error:	51.304	0.877	88.623	0.841	138.424	0.771
Min. absolute error:	0	0	0	0	0	0
Max. absolute error:	284.25	3.367	2360.688	3.462	3913.438	3.368
Correlation coefficient r:	1	0.9554	1	0.9578	1	0.9631
Nodes of hidden layer	12		11		9	
Output:	*P*	*L*	*P*	*L*	*P*	*L*
R squared:	1	0.9236	1	0.9145	1	0.9211
r squared:	1	0.9276	1	0.9198	1	0.9259
Mean squared error:	66127.72	1.206	274268.2	1.349	54180.57	1.245
Mean absolute error:	75.571	0.771	228.425	0.817	83.862	0.772
Min. absolute error:	0	0	0	0	0	0
Max. absolute error:	1656.875	3.369	3617.5	3.454	1384.25	3.445
Correlation coefficient r:	1	0.9631	1	0.9591	1	0.9623

From the table, we could see that ANN with 10 hidden nodes provided minimum errors. The 10 hidden nodes ANN was chosen for further analysis.

Momentum and Weight. After building a frame of the network, I should decide the momentum and the weights. Value for momentum can be obtained adaptively, as in the learning rate. A well-chosen value of momentum can significantly reduce the number of iterations for convergence. Table 4 shows the effect of momentum and weights on simulation errors. Optimum combination of momentum and weight will improve the simulation. When the momentum and the weights were 0.3, the result was worse than anyone else. So the momentum and the weights were chosen as 0.7.

Table 4. Effect of momentum and weights on ABS simulation errors

Momentum-weight	0.5-0.5		0.7-0.7		0.9-0.9	
Output:	*P*	*L*	*P*	*L*	*P*	*L*
R squared:	1	0.9294	1	0.9312	1	0.9358
r squared:	1	0.9339	1	0.9356	1	0.9404
Mean squared error:	4768.34	1.115	3985.219	1.086	89494.51	1.013
Mean absolute error:	25.482	0.695	11.365	0.677	122.058	0.629
Min. absolute error:	0	0	0	0	0	0
Max. absolute error:	464.125	3.345	437.563	3.32	1400.313	3.244
Correlation coefficient *r*:	1	0.9664	1	0.9673	1	0.9697

Conclusion

In this paper I built an ANN in which there were one hidden layer, ten hidden nodes, the momentum and the weights were 0.7. Analyzing ABS simulation by this ANN using simulation data is a simple, convenient and accurate method. Errors could be reduced by selecting hidden layer, hidden nodes, appropriate combination of momentum and weights carefully. In this study, the errors could be controlled well, but the training time is too long to fit the need of braking. So I must choose better simulation model to accomplish the task.

References

[1] Zhou Kongkang, “theory of automobile”, the publish of arms industry,1996

[2] Liu Haiquan, “the study of fuzzy logic ABS control”, Jiangsu University degree paper,2000.1

[3] Zhang Jixian, “Artificial neural network”, Jiangsu University textbook

[4] Li Lin, “the study of automobile anti lock braking system based on fuzzy control”, Jiangsu University degree paper,2003.6

[5]Georg F. Mauer,Gerard F. Gissinger and Yann Chamalllard, Fuzzy Logic Continuous and Quantizing Control of an ABS Braking System, SAE 940830:1033-1042