Estimating Ignition Timing from Engine Cylinder Pressure with Neural Networks

Bryan Willson and Jeff Whitham

Department of Mechanical Engineering
Colorado State University
Fort Collins, CO 80523
(303) 491-6085
FAX: (303) 491-1055

Charles Anderson

Department of Computer Science
Colorado State University
Fort Collins, CO 80523
(303) 491-7491
e-mail: anderson@cs.colostate.edu

Abstract

A study was conducted to determine the ability of neural networks to extract high level control information from cylinder pressure data. Various experiments were performed using neural networks for pattern recognition on a series of data files consisting of cylinder pressure versus crank angle. The goal of these experiments was to estimate spark timing based on the cylinder pressure signature -- all other engine parameters were held constant during the data collection process. Test results indicate that an approximate spark timing value can be obtained using cylinder pressure data as the inputs to a neural network and spark timing as the output.

1. Introduction

In-cylinder pressure data provides one of the most direct measures of combustion quality in an internal combustion engine. Cylinder pressure data has been used for design and diagnostic purposes since the IC engine was developed. Cylinder pressure and volume data can be used to calculate engine torque, indicated mean effective pressure (IMEP), indicated efficiency, bulk temperature, burn rate and heat release. Statistical analysis of the same data can provide information about combustion variability. Recently, more attention has been given to the use of cylinder pressure for real-time engine control [Pes89, KSS88, AP87, GP89, HA86].

Pressure-based engine control techniques require that information be extracted from cylinder pressure data. Examples of pressure based engine control include:

- The location of peak pressure can be monitored to set the optimal spark timing [AP87, Pes89] for power or fuel economy.
- A/F ratio can be estimated from pressure data [GP89]. This could conceivably be used for feedback fuel control.
- Indicated mean effective pressure (IMEP) can be calculated from the pressure data and maximized using a hill-climbing or gradient descent technique [DL51, Kal58, Bla62, Flo64, KPF89, SW90] to achieve maximum torque.

Pressure-based engine control has not been practical for two reasons. The piezoelectric pressure sensors currently used are expensive, fragile, temperature sensitive and require a close-coupled charge amplifier to produce a high level signal. From a signal processing standpoint, calculation of anything more than peak pressure has been too slow for use in real time control.

Recent developments in sensor technology offer some hope on the hardware side of the problem. A compact piezoelectric sensor with a charge amplifier and temperature compensation built in has been developed [AP87, Pes89] although, there are no immediate plans to commercialize the sensor. Piezoelectric washers which are installed between the spark plug and cylinder head have been shown to provide qualitative pressure data at relatively low cost and are reportedly in use in at least one production vehicle for knock sensing. Within the last few years, there have been advancements in the development of low cost fiber optic pressure sensors which are suitable for engine work, and could be mass-produced at low cost.

In many ways, hardware developments appear to be outpacing developments in the area of signal processing. Notable exceptions include the adaptation of digital signal processors (DSP) for real-time calculation of IMEP from pressure data. An investigation was therefore undertaken to assess new signal processing techniques for use in pressure-based engine control.

2. The Role of Neural Networks

It is our premise that advanced signal processing techniques being developed in other fields may be adaptable for pressure-based engine control. As a first step in this investigation, a study was conducted to determine the potential of neural networks to discern information from cylinder pressure data. Rapid advancements in the development of neural network ICs suggest the possibility that they could be cost-effective in selected engine applications in the near future.

The specific focus of this study was to demonstrate the ability of neural networks to extract ignition timing from a cylinder pressure wave form. It is understood that there are easier ways to determine ignition timing. However, this study represents a first step in the development of neural networks which can recognize ignition timing, fuel/air ratio, and other parameters. The eventual goal of the project is to correlate engine data (cylinder pressure data, manifold pressure, engine temperature, engine speed) with engine emissions, and use this information in a neural network based engine controller.

3. Cylinder Pressure Data

Cylinder pressure data for the pattern recognition experiment was obtained from SuperFlow Corporation. The pressure traces were generated on a 350 in.^{3} Chevrolet engine at 4000 RPM. The data was captured with SuperFlow's Engine Cycle Analyzer (ECA). The original files contained cylinder pressure (measured in "bar") at crank angles from 180^o before top dead center (BTDC) to 180^o after top dead center (ATDC). Each file contained 720 data points at 0.5 degree crank angle spacing. Data was taken at various different spark timings from 5^o to 35^o BTDC in 5^o increments. In addition, data was taken at a spark timing of 22^o BTDC to provide a test case for the pattern recognition experiments. The engine speed, air/fuel ratio, and throttle setting were all held constant throughout the data collection period. Thirty independent sets of data for each spark timing were taken over a period of approximately 45 seconds. The total elapsed time to collect all the data, changing only the spark timing between sets of data, was 27 minutes.

The cylinder pressure data away from the top dead center region was nearly identical for all variations of spark timing (see Figure 1). Therefore, the cylinder pressure data was condensed from the original 720 data points to just 37 data points, all within $\pm 40^{o}$ of top dead center (see Figure 2). Crank angle resolution was varied throughout this region, with the smallest resolution (1^{o}) from TDC to 20^{o} ATDC. All files from each set of spark timing data were condensed in exactly the same manner.

4. Neural Network Pattern Recognition

Standard back-propagation neural network architectures were chosen to perform the pattern recognition task on the spark timing data. Different network configurations and different combinations of training and testing data were used in the experiments. The experiments were run on a 33Mhz 80486 computer. A commercial software package was used in the experiments, "BrainMaker 2.1" from California Scientific Software. BrainMaker has a spreadsheet type of user interface for manipulating the data. The condensed cylinder pressure data files (37 points in each file) were used as inputs to the neural network. The only output of the neural network was the spark timing value for that particular input data file. A single hidden layer consisting of 4, 8, 16, or 37 neurons was used in each of the various experiments (see Figure 3). BrainMaker inserts an additional "threshold neuron" in both the input and hidden layers of the network. This neuron does not have any inputs and its output activation level is always the maximum value of the neuron transfer function. The "weight" connecting the threshold neuron to each neuron in the following layer is trained using back-propagation just like every other weight in the network. This provides a convenient method of assigning different activation thresholds to each neuron in the hidden and output layers of the network.

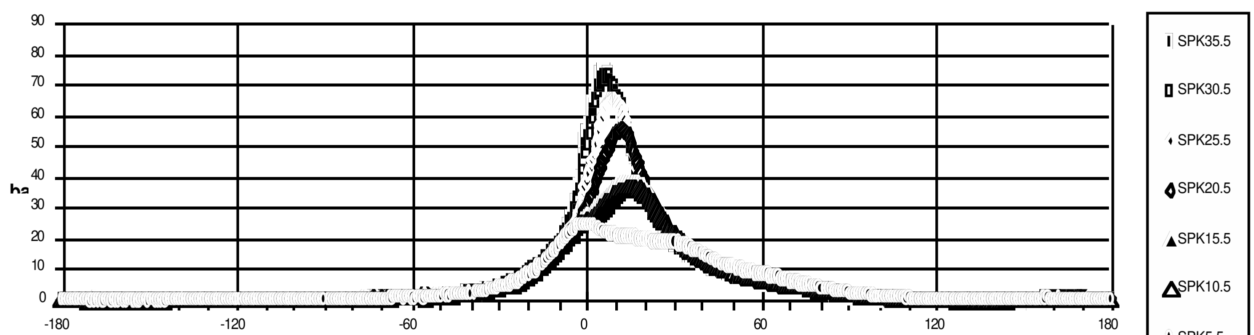


FIGURE 1 -- Cylinder Pressur

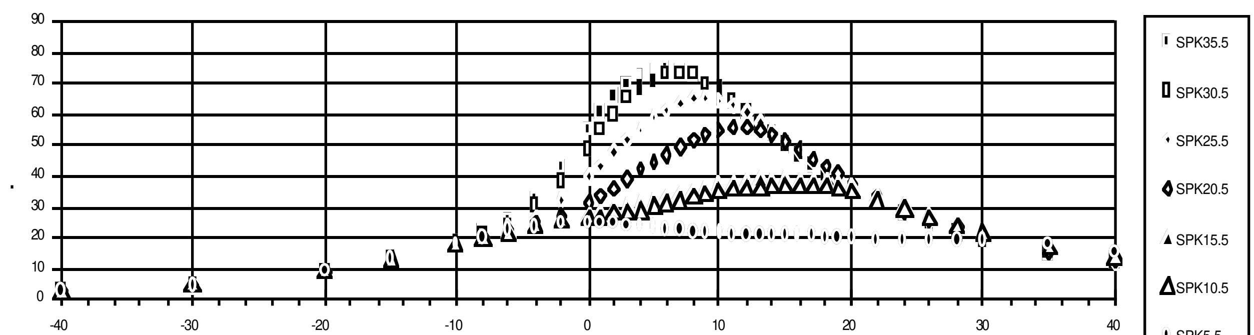


FIGURE 2 -- Condensed Cylinder Pres

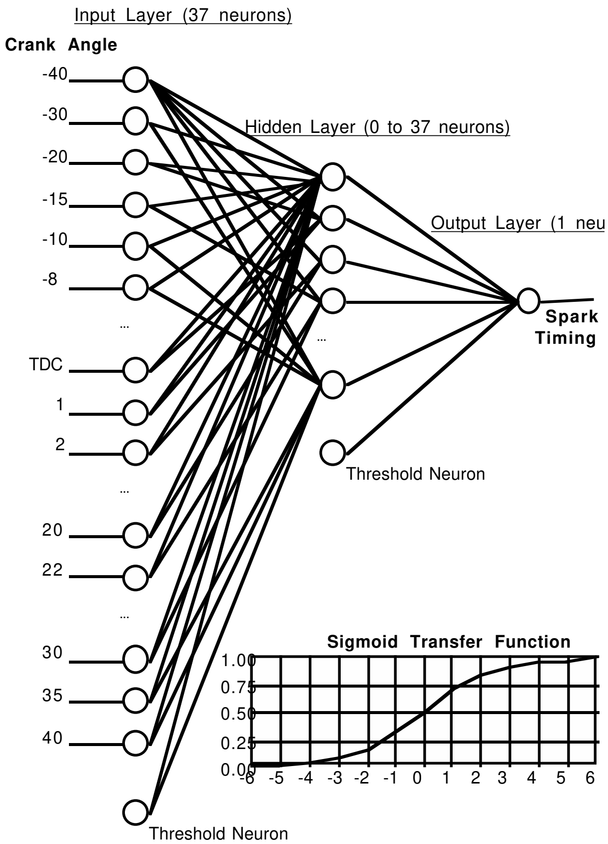


FIGURE 3 -- Neural Network Architecture

An input layer neuron normalizes each data point to a value between 0 and 1 using the minimum and maximum value for each individual point listed in the training data set. Thus, input neurons for cylinder pressure data points with low variance can still produce the same activation level for small input changes as neurons that encounter large pressure variations at that particular crank angle location. Likewise, the output neuron activation level is converted from a value between 0 and 1 to a meaningful spark timing output using the minimum and maximum values in the training data set. Both these normalization procedures use a linear scaling factor and an offset parameter.

The hidden neurons receive a signed weighted sum of the inputs, which is then passed through a sigmoid transfer function to produce an activation level between 0 and 1 (see Figure 3). The output neuron uses an identical process to calculate its activation level except that it receives inputs from the hidden layer instead of the input layer. The output neuron also uses the sigmoid transfer function (clamped between 0 and 1) before its activation level is converted to spark timing. Hence, the output can never go below the minimum or above the maximum spark timing value listed in the training data set. Consequently, spark timing data at 5^o and at 35^o BTDC is more difficult to train to the exact output value because of this nonlinear "squashing" function.

5. Experimental Setup

5.1 Experiment #1

Three different training and testing data sets were used with each of the four hidden layer network configurations described above. In addition, two of the data sets were used on networks with no hidden units. In the first set of experiments, five of the 30 cylinder pressure data files from each of the seven spark timing values were selected to train the networks (a total of 35 training examples). The networks did not train on any of the data files at 22^o BTDC. The remaining 25 data files at each spark timing, as well as all 30 data files at 22^o BTDC, were used to test the generalization capabilities of the networks after they had been trained (a total of 205 test patterns). These networks were labeled 1.0 - 1.4.

5.2 Experiment #2

In the second set of experiments, the first 15 of the 30 cylinder pressure data files from each of the seven spark timing values were selected to train the networks (a total of 105 training examples). The networks did not train on any of the data files at 22^o BTDC. The last 15 data files for each spark timing, as well as all 30 data files at 22^o BTDC, were used to test the networks (a total of 135 test patterns). These networks were labeled 2.1 - 2.4.

5.3 Experiment #3

In the third set of experiments, moving averages were calculated on five consecutive data files for each of the 37 corresponding crank angle locations. Thus, data points in files numbered 1-5, 2-6, 3-7, and so on were averaged on a point by point basis, producing 37 "smoothed" inputs to the network for training and testing. Since

Network Number	# Hidden Neurons	Moving Average	Examples/ Spk Timng	# Epochs to train	Backprop steps	Noise	Training Tolerance	Learning Rate	# Test Patterns	# Good	# Bad	% Correct
1.0*	0	no	5	1000	35,000	.01	.1	.5	205	174	31	84.9%
1.1	4	no	5	482	16,870	.01	.1	1.0	205	146	59	71.2%
1.2	8	no	5	688	24,080	.01	.1	1.0	205	139	66	67.8%
1.3	16	no	5	460	16,100	.01	.1	1.0	205	163	42	79.5%
1.4	37	no	5	290	10,150	.01	.1	1.0	205	156	49	76.1%
2.1	4	no	15	4052	425,460	.01	.1	.5	135	99	36	73.3%
2.2	8	no	15	1644	172,620	.01	.1	.5	135	103	32	76.3%
2.3	16	no	15	536	56,280	.01	.1	.5	135	116	19	85.9%
2.4	37	no	15	814	85,470	.01	.1	.5	135	109	26	80.7%
3.0*	0	yes	5	208	7280	.02	.1	.5	173	155	14	91.7%
3.1	4	yes	5	311	10,885	.02	.05	1.0	173	153	20	88.4%
3.2	8	yes	5	288	10,080	.02	.05	1.0	173	157	16	90.8%
3.3	16	yes	5	344	12,040	.02	.05	1.0	173	171	2	98.8%
3.4	37	yes	5	154	5,390	.02	.05	1.0	173	145	28	83.8%

TABLE 1 -- Network Training and Testing Results

* not run with BrainMaker software

five data files were needed to produce a single moving average data file, only 26 sets of input data could be obtained from the original 30 data files for each spark timing (the first four data files did not have a moving average data file associated with them). Of the 26 moving average data files, five were selected from each of the seven spark timing values to train the networks (a total of 35 training examples). The networks did not train on any of the data files at 22^{o} BTDC. The remaining 21 moving average data files for each spark timing, as well as all 26 data files at 22^{o} BTDC, were used to test the networks (a total of 173 test patterns). These networks were labeled 3.0 - 3.4.

5.4 Method

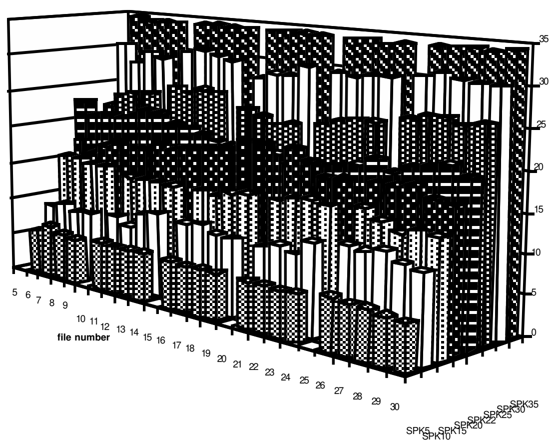
During each "epoch", the entire set of training examples was presented to the network; back-propagation learning took place after the presentation of each training example. An element of "luck" was involved in the actual training time and test results for any experiment because the initial network weights were chosen randomly. Only the best case results are shown in Table 1. A testing tolerance of 10% was used throughout the experiments to determine the number of correct test patterns. The training tolerance was chosen experimentally based on both the number of correct test patterns and whether the network would converge to a solution in a reasonable period of time. The noise parameter was chosen in a similar manner. The back-propagation Learning Rate was adjusted downward if the network nearly converged but then began oscillating close to a final solution for a prolonged period of time.

In other neural network applications, researchers have discovered that adding noise to the inputs of the network as it trains can lead to better generalization capabilities on untrained data. The same conclusion was reached during these experiments. Guassian noise with a standard deviation of 1% of the input data point was added to the inputs during network training in experiments 1 and 2. In experiment 3, the noise level was raised to 2% because the networks training on moving average data could tolerate more noise without encountering the convergence problems at high noise levels discovered in the first two experiments.

6. Results

Table 1 summarizes the training and testing results for each network configuration and training data set. These are preliminary

FIGURE 5 -- Network Output on Testing Data with Movi

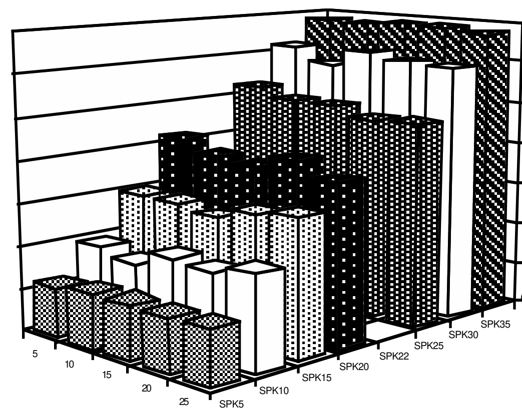


results and have not been averaged over different initial weight values for each network. Tests of statistical significance are required for valid comparisons.

Training time was generally longer for networks with fewer hidden neurons. Additionally, these experiments indicate that the moving average was an important input data transformation, allowing the networks to train more quickly and produce more accurate results on the test patterns.

The moving average networks were capable of training with a much tighter training tolerance and a higher level of input noise. However, the moving average network 3.3 (98.8% correct on the moving average test patterns) correctly classified only 80% of the non-moving average test patterns from the first set of experiments. Moreover, the non-moving average network 1.3 improved from 79.5% correct on non-moving average data to 87% correct on the moving average data. Likewise, the non-moving average network 2.3 improved from 85.9% correct on non-moving average data to 91.9% correct on the moving average data. Therefore, one may conclude that moving averages were important because they reduced cycle to cycle variations in cylinder pressure versus crank angle, permitting the networks to correctly predict spark timing on untrained patterns. Moving averages also allowed the networks to train more quickly while producing results on non-moving average data comparable with networks that were trained using non-moving average data.

FIGURE 4 -- Network Output on Training Data wit Averages



The output of network 3.3 on its 35 moving average training examples is shown graphically in Figure 4 (training tolerance was 5%). Figure 5 shows the output of the same network on the 173 moving average test patterns. Table 2 lists the average output and standard deviation on the test patterns for all networks with 16 hidden neurons.

7. Conclusion

Overall, the experimental results are very encouraging. They indicate that cylinder pressure versus crank angle data can be fed directly into a properly trained neural network producing a close approximation to actual spark timing. Results in Table 1 for the first set of experiments show that hidden units did not reliably increase the classification accuracy. This is probably due to over-fitting -- the training data was so accurately modeled that generalization to testing data is worse. The good performance of the network with no hidden units shows that a linear combination of the pressure data is sufficient to achieve roughly 90% accuracy. The moving average transformation reduces the chance that over-fitting will occur during training.

	Network 1.3		Network 2.3		Network 3.3	
	Avg	StDev	Avg	StDev	Avg	StDev
SPK5	6.85	0.37	6.67	0.56	6.13	0.28
SPK10	9.38	2.06	10.20	1.63	10.40	1.12
SPK15	14.96	2.11	15.77	1.39	15.02	0.64
SPK20	19.32	3.25	19.76	2.35	18.86	1.36
SPK22	24.88	2.98	24.75	2.38	20.58	1.13
SPK25	26.25	2.29	24.59	1.57	24.77	1.29
SPK30	31.60	1.63	31.13	0.98	29.92	0.86
SPK35	32.76	0.36	32.63	0.74	34.07	0.14

TABLE 2 -- Average Network Output & Standard Deviation on Test Patterns for Networks with 16 Hidden Neurons

8. Future Directions

8.1 Analysis

The preliminary results reported here demonstrate the feasibility of learning the mapping between pressure and crank angle. The next step is to analyze the data more thoroughly. As a first step, we have performed linear regression analysis of the data by removing all hidden units and slightly modifying the training algorithm. The magnitudes of the coefficients developed for the linear model suggest that sample points surrounding the maximum pressure samples are most informative. Such results may lead to further reductions in data without sacrificing performance.

We will also analyze the weights acquired by the hidden units. Statistical methods, such as clustering of weight vectors and tests of independence of hidden unit output values, will be used to identify which transformations extract the most relevant information from the pressure data.

8.2 Constructive Learning Algorithms

Recent studies have shown that learning algorithms that incrementally increase the complexity of the approximating function can learn much faster than error back-propagation in some cases. Such methods begin with a linear approximation and develop nonlinearities to remove residual errors. The progression is stopped once the improvement in error is below some criterion. Limiting the complexity of the approximating function results in better generalization to untrained data.

Cascade Correlation [FL90] is a constructive method that performs well for highly-nonlinear mappings. Sanger, et al.'s, [SSM92] method constructs sparse polynomials by estimating the utility of including each input term in a higher-order term of the polynomial, then adjusting the coefficients. This method is able to discern the relevant inputs and learn approximations that do not depend on irrelevant inputs. We will test both procedures on the pressure data.

9. Acknowledgments

The authors would like to thank Dr. Paul Puzinauskas and Mr. Harold Bettes of SuperFlow Corporation for providing the cylinder pressure data, acquired with the SuperFlow Engine Cycle Analyzer.

This work was funded in part by the Pipeline Research Committee of the American Gas Association.

References

[AP87] C. M. Anastasia, G. W. Pestana, "A Cylinder Pressure Sensor for Closed Loop Engine Control", SAE Paper 870288, 1987.

[Bla62] P. F. Blackman, "Extremum-Seeking Regulators, and an Exposition of Adaptive Control", Pergamon Press, 1962.

[DL51] C. S. Draper, Y. T. Li, "Principles of Optimalizing Control Systems and an Application to the Internal Combustion Engine", *ASME Publications*, 1951.

[FL90] S.E. Fahlman and C. Lebiere, "The Cascade-Correlation Learning Architecture", *Advances in Neural Information Processing Systems*, volume 2, pp. 524-532, Morgan Kaufmann, San Mateo, CA, 1990.

[Flo64] J. J. Florentin, "An Approximately Optimal Extremal Regulator", *J. Electronics and Control,* XVII, 2, p. 211, 1964.

[GP89] E. H. Gassenfeit, J. D. Powell, "Algorithms for Air-Fuel Ratio Estimation using Internal Combustion Engine Cylinder Pressure", SAE Paper 890300, 1989.

[HA86] Y. Hata, M. Asano, "New Trends in Electronic Engine Control - To the Next Stage", SAE Paper 860592, 1986.

[Kal58] R. E. Kalman, "Design of a Self-Optimizing Control System", ASME Transactions, 1958.

[KPF89] D. B. Kittelson, M. J. Pipho, M. L. Franklin, "Dynamic Optimization of Spark Advance and Air-Fuel Ratio for a Natural Gas Engine," SAE Paper 892142, 1989.

[KSS88] Y. Kawamura, M. Shinshi, H. Sato, N. Takahashi, M. Iriyama, "MBT Control through Individual Cylinder Pressure Detection", SAE Paper 881779, 1988.

[Pes89] G. W. Pestana, "Engine Control Methods Using Combustion Pressure Feedback", SAE Paper 890758, 1989.

[SSM92] T.D. Sanger, R.S. Sutton, C.J. Matheus, "Iterative Construction of Sparse Polynomial Approximations", *Advances in Neural Information Processing Systems*, volume 4, Morgan Kaufmann, San Mateo, CA, 1992.