CI-PASM – Computational Intelligence Based Prognostic Automotive System Model

Denis Todd Vollmer
Idaho National Laboratory
Idaho Falls, Idaho USA
denis.vollmer@inl.gov

Milos Manic
Department of Computer Science
University of Idaho at Idaho Falls
Idaho Falls, Idaho USA
misko@uidaho.ed

***Abstract*—In an ideal case physically oriented vehicle models can reduce the required practical knowledge of a vehicle designer. These types of models are effective cost reducing tools used in industrial development cycles. There are many variables that can be used as input both internal and external to model automobile performance. The focus of this paper is on those external variable factors such as environment conditions that are not controllable by a human but are instantaneously measurable and affect performance. This paper presents CI-PASM, A Computational Intelligence Based Prognostic Automotive System Model. Initial feature reduction was accomplished by a human expert. Principal Component Analysis was performed to further reduce the input set. Using expert chosen features, the CI-PASM algorithm produced results having an error at worst in the hundredths of a second. These output results were compared against a support vector machine implementation and were shown to be superior. The CI-PASM mean error was half that of the support vector machine error. Results from using PCA attributes and a support vector machine indicated that these are relevant alternative methods given different requirements.**

Index Terms—Neural Networks, Support Vector Machines, Road Vehicles, Regression

I. Introduction

Automotive systems analysis and prognostics present a pinnacle point of interest in many industrial and engineering engine calibration and development systems [1],[2]. However, many aspects of these applications are problem specific [3]. For example, software models of diesel engines have been used to adjust parameters to reduce carbon emissions [4]. This paper focuses primarily on environmental, i.e. external conditions, which are difficult for humans to control but occur in real life applications outside a carefully controlled laboratory. This problem will be investigated on an extreme case study of drag racing, with understanding that the calibration problems will assume only a subset of the presented problem.

Predicting the elapsed time of an automobile in a National Hot Rod Association (NHRA) drag race is one of the most important capabilities to support a victory in competition. Although commercial simulation packages that provide estimates can be found, academic information appears to be sparse. As the commercial entities practice a closed source code and they maintain a competitive advantage by keeping their process secret, it is not possible to determine the algorithms without some form of reverse engineering. Therefore, in order to form a baseline, this paper will explore the use of supervised learning techniques to provide time estimates.

A drag race is an acceleration contest between two vehicles over a predefined distance. The goal is to traverse the distance and cross a finish line first. In a simplification of the actual process, the winner is then subsequently paired against other winning drivers. This sequence repeats until only one driver is left. The distance measure is usually either a quarter mile (402 meters) or an eighth mile (198 meters).

Two different official measures are tracked for each performer, elapsed time and speed. Elapsed time (ET) is the more important measure as a handicap procedure is used to ensure a fair competition between two cars of differing performance. Each driver, in advance of a run, provides an anticipated ET value. The difference of these two values is computed and used to inhibit the start of the faster car. To discourage drivers from seeking an unfair advantage by giving slower ET values, they are disqualified if their actual time is better than the estimated time. This condition is referred to as a breakout. The ability to predict an accurate ET is vitally important to the success of a racing team.

II. State of the Art

The application of computational intelligence techniques to automotive system models can be found in literature such as fuzzy controllers, adaptive neural networks and proprietary solutions. In particular, engine performance simulations using experimental engine conditions have been successfully conducted using an adaptive neuro-fuzzy inference system [5]. The experimental system made use of standard engine output measurements such as power, torque and CO2 values to predict engine input measurements.

In [6] the authors developed evolving artificial neural networks to warn drivers of dangerous situations. This showed that in principle the use of neural networks to predict the affect of driver and automobile interactions is feasible. The project at the University of Texas made use of an evolving neural network method called NeuroEvolution of Augmenting Topologies (NEAT) [7]. NEAT evolves the network topology, in addition to the weights, developing an arbitrary sized network. The resulting test systems output an estimated time to

978-1-4244-2800-7/09/$25.00 ©2009 IEEE

crash used to evaluate danger levels. An average under 0.04 mean squared error in predicted time to crash showed a strong promise for the accuracy of such a system.

Numerous commercial applications exist to monitor and/or predict automobile performance [8]-[10]. These applications require several inputs and appear to base their predictions upon the physics of automobile behavior. Most of the vendors' publicly available information provided inadequate details to make an objective evaluation of this observation. However a review of one vendor's patented application for monitoring and controlling wheel traction does provide supporting evidence for the supposition [11]. This patent was filed in support of a system created by Davis Technologies.

Davis Technologies offers a patented two stage system that applies a retarding timing correction signal to the ignition system. The term retarding means that the system is effectively reducing engine output power to the transaxle by means of engine timing adjustments. Their system controls wheel slip by sensing over rev and a few unmentioned signals. This information is then used to throttle back the engine. Wheel slip information is sampled and a feedback signal is applied at up to 400 times per second to the ignition system. This information is learned by running trial runs and then used for predicting future race car performance. The claim behind their system is that unlike other systems their product learns from previous trial runs and applies that information to the present run, their system learns from previous runs as well as using instantly acquired information to adjust the traction control. The patent description of learning is different than that typically applied to computational intelligence techniques like neural networks.

III. Background

This section provides background on two machine learning techniques artificial neural networks with back propagation and support vector machines. These algorithms are used in the CI-PASM and comparison routines respectively. In addition the concept of input vector preprocessing is presented.

This paper presents a solution that can utilize similar instantaneous forms of information as those mentioned section II. The information is used to generate an acceleration profile for a given vehicle regardless of track location. The algorithm described uses measured external forces to compensate for driver and environmental variations acting on the vehicle chassis during a drag racing event. Environmental temperature and barometric pressure are two examples of several variables that are monitored and used as input. The algorithm learns overall system behaviors and then adjusts for varying conditions. After the algorithm has been trained on a series of recorded runs, it functions in a prediction mode. In this mode, the algorithm can perform predictions of future run times based on measured environmental conditions and learned knowledge acquired from past performance.

Preprocessing of input data is one of the most important steps in development of a neural network solution [15]. In some cases the data set may be missing or its meaning obscured by excessive number of attributes. This is referred to as the curse of dimensionality. In addition, the numerical values of the data are normalized to help equate the strength of the variables. Encoding of features such as weather conditions is performed. This data point is a categorical discrete value that has no natural ordering. Bishop in [15] suggests using a 1-of-c coding for this kind of input. The data points that most affect the solution are optimal candidates for inputs while others are discarded. If too much information is removed, the resulting prediction ability will be affected.

A standard multi-layer feed forward Error Back Propagation network, or EBP for short, is at the heart of the CI-PASM algorithm. EBP networks are another well researched supervised learning method. The learning vectors are presented to the network and the results are fed forward through the network. Results are calculated and compared to the desired output producing an error measurement. The power of a multilayer neural network lies in its ability to train the network to model multidimensional nonlinear problems.

Back propagation solves the problem known as the *credit assignment problem* i.e. the determination of a specific node to be adjusted in a multilayer network with a hidden node layer. If the error function such as a simple sum-of-squares equation,

$$\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 \tag{1}$$

is used then this is a differentiable function of the weights. The derivatives of the error with respect to the weights can be performed and applied to find weight values that minimize the error function.

Multiple approaches can be taken to adjust the neural network output. One method is to produce an output, based on historical data of an entire event, and adjust engine output power by adding or subtracting a single set value to the entire relevant event. Another approach is to break the profile up into smaller segments of time and allow the network to make adjustments within each time segment. Considering the initial attempt to predict an overall time without corrective feedback during the run, the former is followed in this solution.

For comparison purposes a regression version of the Support Vector Machine [17] was implemented. It has been shown to have good empirical performance and has a strong research history [18]. The SVM was originally proposed by Vladimir Vapnik in 1963 as a linear classifier. In 1992 the idea was extended to be a non-linear classifier by Vapnik, Boser and Guyon. A non-linear kernel function was added. Vapnic and others extended the SVM for regression (SVR) in 1996 [19]. The idea behind an SVM is to maximize the distance of a decision surface between examples in a training set. For example, given a set of two points the midpoint of the euclidean distance would be the maximum distance between them. As pointed out by Rychetsky (2001), classical learning systems like neural networks suffer from their theoretical weakness, e.g. back-propagation usually converges only to locally optimal solutions. SVMs can provide a significant improvement in this regard. For this reason an SVR implementation was chosen as a comparison algorithm to EBP.

IV. PROBLEM DESCRIPTION

It is well known that several external environmental measures factor into a vehicles performance [12],[13]. Over a given day as temperatures increase and other environmental factors vary so does a vehicles elapsed time. The data acquired for this paper shows this as well. The effectiveness of this system is based in part on the ability for a given expert driver to perform in a predictable manner time after time. The more a driver's behavior departs from an optimal winning behavior, the more of an effect it has on vehicle performance. The better drivers in turn will require little compensation by the system. In this way an overall drag racing system and driver become tuned together for optimal performance. In this case optimal performance means getting to the finish line as close as possible to a predefined time without going over.

Race teams generally maintain meticulous log books recording track and automobile conditions for each race. In support of this project, a log book for an NHRA record setting Ford Mustang was obtained that covers the 2003 and 2004 race seasons. The logged measures are shown in Table I. Other than time values and exhaust temperature, measurements were taken prior to an individual race.

For a vehicle to move the energy generated by the engine is used to overcome two forces that act in resistance. Rolling friction is the resistance of the tire in contact with the track surface. This friction force is nearly independent of car speed. For drag racing vehicles this value is generally much higher due to special tires, vehicle weight distribution and pavement conditions [14]. For instance, track crews spray a special chemical compound on the starting line area to improve tire traction. The second force is air resistance force, and it increases rapidly with speed

The power from an engine must supply sufficient forward force to the driven wheels in order to overcome these two resisting forces. The density of air and friction coefficient can change with variations in atmospheric density and air temperature. These and other values such as shock settings and wind can affect the performance of a race car. Power is required to apply force at a rate fast enough to do the work of accelerating the vehicle.

TABLE I. MEASURES

Time of Day	Reaction Time
Light Conditions	Air Temperature
60 ft. Time	330 ft. Time
660 ft. Time	660 ft. Miles Per Hour
990/1000 ft. Time	
1320 feet Time	1320 feet Miles Per Hour
Launch RPM	Shift RPM
Fuel Pressure	Tire Pressure Front/Rear
Weight	Shock Settings Front/Rear
Humidity	Barometric Pressure
Exhaust Temperature (Post race)	

A vehicles power rating measured in an uncontrolled environment with varying environmental conditions will vary greatly based on the data presented in this paper. As the atmospheric temperature, barometric pressure and other factors vary over a given day so does car performance. Both tire-to-track friction and engine performance are affected. If all the forces acting on the frame of the racer are understood, then accurate methods of controlling and predicting race care acceleration and time to finish line can be developed.

V. CI-PASM ALGORITHM

The conjecture of this paper is that automobile performance can be accurately modeled based on historical measurements of atmospheric, geographical and car performance. This prediction process is formulated as a regression analysis computation. Regression problems can be viewed as a specific case of function approximation. The outputs should approximate the conditional averages of the target data.

In order to reduce the dimensionality of the independent data sets and choose those that are most appropriate two methods were employed: human expert analysis and Principal Component Analysis (PCA). Initially the expert chosen categories were used for input to provide a baseline for future feature reduction effectiveness. Even though this solution provided the best results, feature set reduction was still considered. Computation performance is improved with fewer points and a relatively accurate answer is produced.

Information gathered from a specific run down the track, as depicted by Fig. 2, by a vehicle is represented as a feature vector $\mathbf{F} = \{f_1, f_2, ..., f_n\}$ where f_i is a feature in an independent feature set. Given a set $\mathbf{FS} = \{\mathbf{F}_1, \mathbf{F}_2, ...\mathbf{F}_k\}$ that contains all instances of available recorded feature vectors $\mathbf{F}_i$ we want to predict the elapsed time value t_i where set $\mathbf{T} = \{t_1, t_2, ...t_x\}$.

PCA is a technique for combining inputs together to make a generally smaller subset [16]. The goal is to compute a basis to express as a representation of a noisy data set. In other words, what are the most important aspects of this data set and which are redundant or noise. Formally stated the process of mapping a given vector set x^n of dimension $\mathbf{D}$ $(x_1...x_d)$ to an alternative vector set y^n of a smaller dimension $\mathbf{M}$ where $\mathbf{M} < \mathbf{D}$. Exact details of this process are described in literature and are not

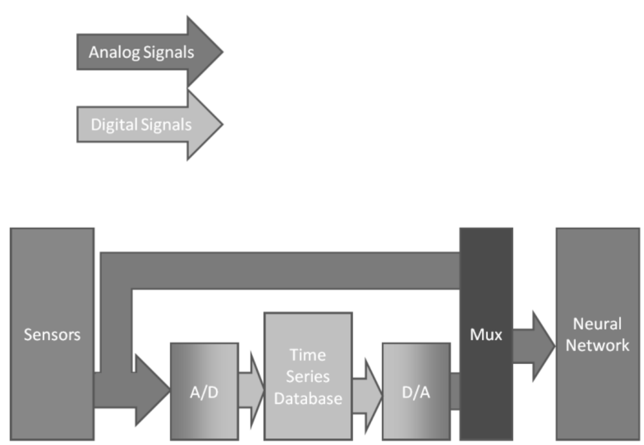

Figure 1. Data Flow Diagram

covered here [16],[17]. However the basic principles are outlined as follows: Organize the data into an *m* x *n* matrix where *m* is the measurement and *n* is the number of samples. Subtract off the mean for each measurement type. Calculate the eigenvectors and eigenvalues of the covariance. The eigenvectors corresponding to the largest eigenvalues are retained.

The ultimate goal is to predict elapsed time values for a given vehicle. This can be achieved by training a regression function g such that g($\mathbf{F}_i$)=>$\mathbf{T}_i$. An Error Back Propagation Neural Network (EBP) implementation is used to perform this regression analysis. A regression form of Support Vector Machine (SVR) was used for comparison purposed but is not a required portion of CI-PASM.

The algorithm steps are as follows:

Step 1: From the recorded historical information about the vehicle construct a sequence S_T of ordered attribute vectors $\vec{v}_{t_i}$. Each $\vec{v}_{t_i}$ should contain the elapsed time in a given position.

Step 2: Transform the set of attributes S_T into a new attribute set S_W by reducing the cardinality of each vector to a constant length. Feature selection is accomplished by a human expert followed by PCA over the ordered sequence of vectors $\vec{v}_{t_i}$. Compute a new attribute vector $\vec{w}_j$. Dimensionality reduction should be accomplished by choosing enough eigenvectors to account for a 95% variance in the original data.

Step 3: Create training and testing sets. T1 contains training feature vectors using human expert features and testing sets S1. T2 contains training feature vectors using a PCA features and test sets S2. The features should be split, with 2/3 for training and the rest for test.

Step 4: Train the EBP with both training sets T1 and T2. The feed forward neural network with back propagation of errors is constructed with three layers as can be seen in Fig. 2. The input layer consists of the twelve expert feature values. Six nodes reside in the middle hidden layer. These nodes feed into a single output node which is ultimately responsible for delivering the elapsed time estimate. The network is fully connected with all nodes in a layer connected to each node in the next layer. The connectivity depicted in fig. 2 is not complete to avoid cluttering the image.

The input signal of the hidden and output node is defined in (2) where *w* is the weight and *x* is the input value from the previous node and *m* is the number of connections including the bias term.

$$net = \sum_{i=0}^{m} w_i x_i \tag{2}$$

The output from each node is computed using (3). The *k* value is the gain and is defined as a set constant of 0.2.

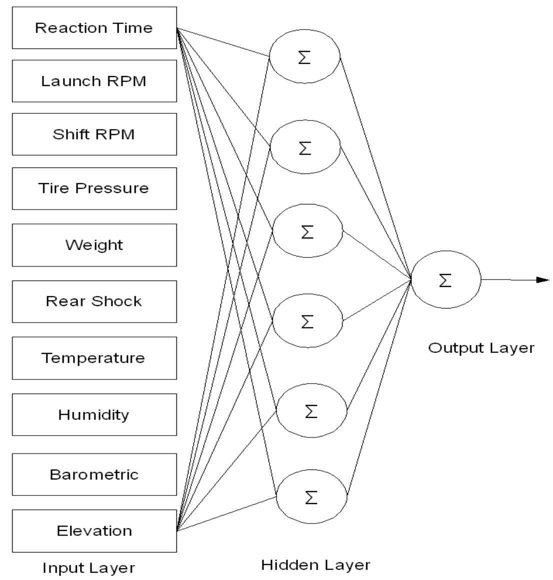

Figure 2. Network Diagram

$$o_j = \tanh(k \bullet net_j) \tag{3}$$

After the value for the output layer is produced the error is

computed and fed back through the hidden and output layers and the weights *w* are adjusted. Equation 4 provides the delta to apply to each weight, *np* is the number of patterns, d is desired value

$$\Delta w_p = \alpha \bullet \sum_{o=1}^{1} \sum_{p=1}^{np} \left[\left(d_{op} - o_{op} \right) F'\{z_p\} f'(net_p) x_p \right] \tag{4}$$

After a complete set of updates for each training input pattern, the sequence of calculating outputs and feeding error information back through the network layers is reiterated 400 times the output error is acceptably small.

Step 5: Using the trained EBP to evaluate against testing sets S1, S2.

An important tool used for analysis and implementation of the algorithms is Weka from the University of Waikato [20]. Weka is a collection of machine learning algorithms implemented in Java and issued under the GNU General Public License. It is very flexible in its usage allowing for users to link against a library or make use of graphical user interface. Data can be imported and manipulated in comma separated data or in its own well defined ASCII format arff. Once the data is imported it is possible to make multiple runs with different algorithms. Results are then saved as a text file for later consumption. This software package was used exclusively to perform all computation during this project.

TABLE II. EXPERT ATTRIBUTES

Reaction Time	Rear Shock Setting
Launch RPM	Shift RPM
Tire Pressure Front	Tire Pressure Rear
Humidity	Barometric Pressure
Elevation	Weight
Temperature	1320 feet Miles Per Hour

VI. RESULTS

The data set used for training and testing consists of forty-five runs at five different geographic sites. Each site is located in the United States and has an elevation in the range 1,374 to 5,860 feet above sea level. After evaluating the raw data, it became evident that elevation affects performance. The average ET at 5,860 feet was 13.2 seconds while at 1,374 feet the average ET was 12.5 seconds. The physical aspects of the car such as engine parts, suspension and tires were not changed, other than simple adjustments, between track sites.

Data value instances of the expert chosen categories shown in Table II were normalized and all available sets were offered as input to the CI-PASM algorithm. This included running the principal component analysis process. The result was a reduction to six attributes. Both sets of data, expert and PCA derived, were used as input to the SVR implementation for comparison purposes.

Of the 45 instances of input vectors 39 were used to train and 16 were for test. Runs at all tracks were represented in both training and test sets. The 1320 foot elapsed time attribute is the output value the algorithm is trying to predict and consequently used to compare accuracy.

For comparison purposes, Table III presents the correlation coefficient and error results of four different combinations of algorithms. These include the two input selection methods and both machine learning methods of SVR and EBP that were presented in section 3. An implementation of Alex Smola and Bernhard Scholkopf's sequential minimal optimization algorithm with a polynomial kernel was used as the support vector regression model [20]. The EBP data was produced using a multilayer neural network that uses the method of error back propagation to adjust network node weights. A good reference for this method can be found in the book Neural Network Design [21].

TABLE III. SUPERVISED LEARNING RESULTS

	SVR Expert	SVR PCA	EBP Expert	EBP PCA
Correlation coefficient	0.979	0.975	0.995	0.970
Mean absolute error	0.063	0.089	0.029	0.079
Root mean squared error	0.089	0.124	0.038	0.100
Relative absolute error	16.44%	23.38%	7.95%	20.77%
Root relative squared error	21.83%	30.10%	9.96%	24.39%

TABLE IV. EBP EXPERT SAMPLE DATA

Actual	Predicted	Error	Actual	Predicted	Error
12.464	12.344	-0.12	…	…	..
13.295	13.239	-0.056	13.173	13.176	0.003
12.366	12.512	0.146	12.56	12.627	0.067
12.494	12.485	-0.009	13.184	13.183	-0.001
…	…	…	13.142	13.173	0.031

The graphs shown in Fig. 3 on the next page depict the predicted ET's with triangles overlaid with the actual ET's represented as squares. All four instances follow the general solution curve well. It can be observed that the different algorithms had accurate predictions at different data points. However A few run instances such as 8 and 14 caused accuracy problems for all the solutions and may indicate an issue those points.

Table IV contains a sample output of the CI-PASM algorithm using EBP with expert features. Measured actual ET's are shown next to that of the predicted values. Error is represented as a simple difference between the two time values. Overall the EBP with expert chosen features had the highest correlation coefficient and the smallest error. In all cases the expert picked attribute set performed better than that of the PCA attributes. However the error difference may be acceptable depending upon the solution requirements.

VII. CONCLUSIONS

Based on the results shown in this paper supervised learning algorithms implemented to model automotive performance is not only feasible but also very accurate. Current industrial practices leverage similar models to reduce cost and shorten development cycles. The presented CI-PASM algorithm demonstrated the use of a multi-layer error back propagation neural network and compared its performance with a support vector regression model. Two methods of input feature dimensionality reduction, human expert selection and PCA were presented as well. All showed promising results with the combination of human expert feature selection and EBP network performing best with a correlation coefficient of 0.995 and a mean absolute error of 0.029.

Human performance enhancement is an interesting topic and use of a CI-PASM system may find application elsewhere. The algorithm could be enhanced to include a mode where historical data is actively applied to instantaneous data and used to correct human behavior in real time. There may be an optimal mixing factor for such a hybrid system that allows the human to maintain control while allowing for fine accuracy adjustments by the artificial network. For example, a similar approach is typically used to increase stability and pilot control of highly maneuverable forward swept wing tests.

ACKNOWLEDGMENT

The authors are indebted to C. Pope for his meticulous tracking of the raw data provided for this paper. He is the owner and operator of the Ford Mustang used to provide the vehicle run data used in this paper.

REFERENCES

[1] A. Davare, Q. Zhu, M. Natale, C. Pinello, S. Kanajan, A Sangiovanni-Vincentelli, "Period Optimization for Hard Real-time Distributed Automotive Systems", DAC 2007, June4-8, 2007, San Diego, CA

[2] C. Atkinson, M Traver, T. Long, E. Hanzevack, (Jun. 2002). "Predicting Smoke", *ISA Virtual Sensors* URL from Nov. 2008. Available: http://www.isa.org/

[3] M. Traver, R. Atkinson, C Atkinson, "Neural Network-Based Diesel Engine Emissions Prediction Using In-Cylinder Combustion Pressure," West Virginia Univ., SAE 1999-01-1532, May 1999

[4] K. Satake, T. Monaka, S. Yamada, H Endo, M Yanagisawa, T. Abe, "The Rapid Development of Diesel Engines Using an Optimization of the Fuel Injection Control", Mitsubishi Heavy Industries, Ltd. Tech. Review Vol. 45 No. 3, Sep. 2008

[5] K. Tutuncu and N. Allahverdi, " Reverse modeling of a diesel engine performance by FCM and ANFIS", Int. Conf.on Computer Systems and Technologies, 2007

[6] K. Stanley, N. Kohl, R. Sherony, R. Miikkulainen, M. Samples, "Evolving a Real-World Vehicle Warning System", in *Genetic And Evolutionary Computation Conference*, Seattle, WA. 2006, pp. 1681-1688

[7] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99-127, 2002.

[8] QUARTER Pro, URL from Nov. 2008, Available: http://www.quarterjr.com/products.htm

[9] Engine Simulation Software, URL from Nov. 2008, Available: http://www.storesonline.com/site/573683/page/189725

[10] Stationary drag racing simulation system, by J. Teresi and S. Prince, (2003, Jan 14). U.S. Patent 6505503. available: www.patentstorm.us

[11] Traction control system, by S. R. Davis, (2001, Apr 30). U.S. Patent 6577944.

[12] J. Hand, *How to Build Max-performance Pontiac V-8s,* North Branch,MN,Car Tech Inc., 2004.

[13] S. Soares and J. Sodre, (2002). Effects of atmospheric temperature and pressure on the performance of a vehicle. *J. of Automobile Eng*. Vol 216(6). pp 473-477 Available:http://journals.pepublishing.com

[14] R.C. Smith,"General Physics and the Automobile Tire", American Journal of Physics, Volume 46, Issue 8, pp. 858-859, 1978

[15] C. M. Bishop, *Neural Networks for Pattern Recognition*, New York:Oxford Press,1995

[16] J. Shlens, "A Tutorial on Principal Component Analysis", Salk Institute and University of California, San Diego, Dec. 10, 2005 Ver. 2

[17] A.J. Smola and B. b.Scholkopf. "A tutorial on support vector regression." NeuroCOLT2 Technical Report NC2-TR-1998-030, 1998

[18] K.R. Muller, A.J. Smola, G. Ratsch, B. Scholkopf, J.Kohlmorgen and V. Vapnik. "Using Support Vector machines for Time Series Prediction", 2000

[19] V. Vapnik, The Nature of Statistical Learning Theory, New York, NY, Springer, 1995

[20] I. .H. Witten and E. Frank, Data Mining: Practical machine learning tools and techniques, 2nd Ed., Morgan Kaufmann, San Francisco, 2005.

[21] M. Hagan, H. Demuth, M. Beal, "Neural Network Design.", PWS Publishing Company, 1995

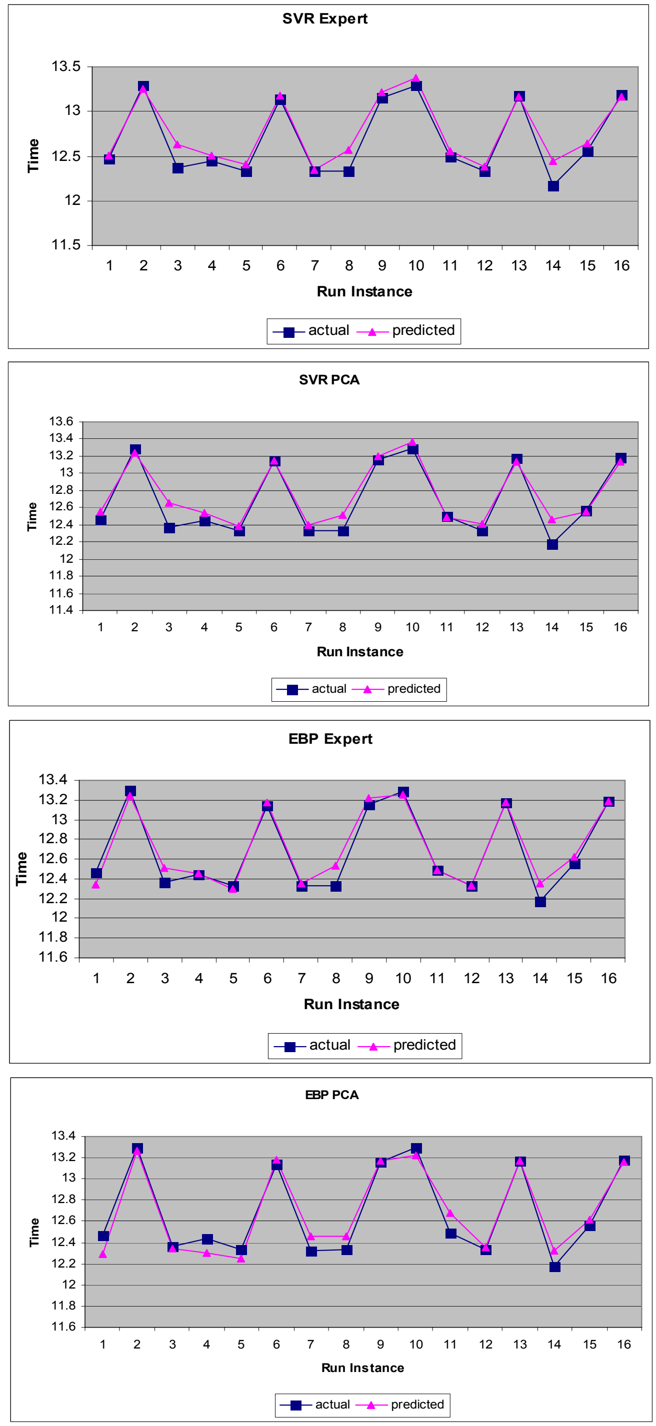

Figure 3. Graphs showing predicted outputs with actual ETs.