NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A NEURAL NETWORK APPROACH FOR HELICOPTER AIRSPEED PREDICTION

by

Ozcan Samlioglu

March 2002

Thesis Advisor: Russell W. Duren
Assoc. Advisor: Monique P. Fargues

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2002	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Title (Mix case letters) A Neural Network Approach for Helicopter Airspeed Prediction			**5. FUNDING NUMBERS**
6. AUTHOR(S) Ozcan Samlioglu			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

Conventional pitot-static airspeed measurement systems do not yield accurate measurements when aircraft speed is below 40 knots. Recent studies have demonstrated that neural network approaches for predicting airspeed are quite promising. In this thesis, a back-propagation neural network is used to predict the airspeed of UH-60A and OH-6A helicopters in the low speed environment. The input data to the neural networks were obtained using the FLIGHTLAB flight simulator. The results obtained by flight simulation were validated by comparison to results of a previous study of the UH-60A helicopter based on actual flight data. The results of the work performed for this thesis show that at sea level the UH-60A low airspeed can be predicted with an accuracy of ± 0.71 knots and ± 0.88 knots for out of ground effect and in ground effect conditions respectively. OH-6A analyses were performed at two pressure altitudes. At sea level the OH-6A airspeed can be predicted with an accuracy ± 0.75 knots when the aircraft is out of ground effect and ± 0.88 knots when the helicopter is in ground effect. At a pressure altitude of 6000 feet OH-6A airspeed can be predicted with an accuracy of ± 0.64 knots for both flight conditions.

14. SUBJECT TERMS Airspeed measurement, Neural Networks, Back-propagation			**15. NUMBER OF PAGES** 137
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

A NEURAL NETWORK APPROACH FOR HELICOPTER AIRSPEED PREDICTION

Ozcan Samlioglu
1st Lieutenant, Turkish Army
B.S., Turkish Army Academy, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2002

Author: Ozcan Samlioglu

Approved by: R. W. Duren, Thesis Advisor

M. P. Fargues, Assoc. Advisor

Max F. Platzer, Chairman
Department of Aeronautics and Astronautics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Conventional pitot-static airspeed measurement systems do not yield accurate measurements when aircraft speed is below 40 knots. Recent studies have demonstrated that neural network approaches for predicting airspeed are quite promising. In this thesis, a back-propagation neural network is used to predict the airspeed of UH-60A and OH-6A helicopters in the low speed environment. The input data to the neural networks were obtained using the FLIGHTLAB flight simulator. The results obtained by flight simulation were validated by comparison to results of a previous study of the UH-60A helicopter based on actual flight data. The results of the work performed for this thesis show that at sea level the UH-60A low airspeed can be predicted with an accuracy of ± 0.71 knots and ± 0.88 knots for out of ground effect and in ground effect conditions respectively. OH-6A analyses were performed at two pressure altitudes. At sea level the OH-6A airspeed can be predicted with an accuracy ± 0.75 knots when the aircraft is out of ground effect and ± 0.88 knots when the helicopter is in ground effect. At a pressure altitude of 6000 feet OH-6A airspeed can be predicted with an accuracy of ± 0.64 knots for both flight conditions.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

My special thanks goes to Dr. Russell Duren for his assistance, endless effort and patience throughout the whole process of this thesis. I am truly grateful to Dr. Duren for his friendship and generosity during my education in NPS. I also wish to thank Dr. Monique Farques for her valuable guidance, time and patience. Her insight and knowledge helped me to facilitate this effort.

I owe many to thanks to LT. Gregory Ouellette, USN for his endless help, support and valuable friendship. His help was crucial to the completion of this thesis.

I would like to extend my appreciation to Jan Goericke and Dr. Dean Hoff for their kindness and help.

I am greatly appreciative of the opportunity provided by the Turkish Armed Forces Command to study for this degree.

Most of all, I would like to thank my family. I dedicate this work to my parents and my brother Ercan, for their infinite love and support over the years and to my dear wife Hamra and my lovely daughter Bersan for their never-ending love and understanding. Without their support none of this would have been achieved.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Since the 1970s there has been a significant concern to increase the accuracy of airspeed measurement systems because current measurement systems are ineffective at low airspeeds. The focus of this thesis is to develop a neural network (NN) model to estimate helicopter airspeed in the low speed environment using the NeuralWorks Professional Plus/II software. A flight simulator provides the required data that is used as inputs to the neural network model.

The thesis is organized into five chapters. Chapter I is the introduction chapter, which presents research results currently available in the area of neural network and aircraft speed measurement systems. Chapter II introduces the main concepts behind NN implementations and the basic structure of the Back-Propagation Neural Network (BPNN). Chapter III describes a specific implementation using the NeuralWare Professional Plus/II software and presents the FLIGHTLAB simulation tool. The simulations performed as part of this thesis are discussed in Chapter IV. Finally, Chapter V summarizes results obtained and discusses avenues for further research.

A. BACKGROUND

Since 1995, the Naval Postgraduate School (NPS) has conducted several research projects using OH-6A "Cayuse" helicopters after receiving two such helicopters from the Massachusetts Army National Guard. One of these projects involves the development of a Vortex-Ring State Warning System (VRSWS). The vortex-ring state is a condition of powered flight where the helicopter settles into its own downwash. The helicopter will increase its rate of descent very rapidly as the lift is lost when entering the vortex-ring state, and any further application of collective, a flight control mechanism for helicopters, tend to reduce rotor efficiency. In this state the rotor experiences a very high vibration level and loss of control. The consequences of the vortex-ring state when the helicopter is close to the ground might be extremely dangerous because loss of control at low altitudes often results in aircraft crash. Development of VRSWS requires better airspeed measuring systems than these currently available with most avionics systems.

Last year, NPS signed a Cooperative Research and Development Agreement (CREDA) with Advance Rotorcraft Technology, Inc. (ART) to reciprocally develop advancements in rotorcraft technology. Based on this agreement, NPS provided ART with equipment for a flight simulator, including the sticks and grips, seats and avionics suite of an OH-6A for integration into the control loader platform. ART provided their helicopter modeling suite with advanced visual rendering equipment to produce a fully functional stationary open platform flight simulator. In return for this equipment, ART provided the flight modeling software to NPS to develop rotorcraft models for future use which includes the OH-6A and V-22 "Osprey" models. NPS was able to obtain flight parameters such as rotor RPM, engine torque, roll rate, etc., using simulation techniques through its cooperation with ART. LT Gregory Ouellette is currently conducting research about FLIGHTLAB, a flight simulation tool provided by ART. The required flight parameters for the NN model of this thesis were provided by LT. Ouellette, USN [Ref.19].

B. TECHNOLOGICAL PROBLEM

The aerodynamic velocity in the plane of a helicopter's rotor disc significantly affects the control characteristics of that helicopter. The pilot easily feels this effect during flight and especially in hover, low speed and transition conditions. Small changes in these flight regimes are rapidly followed by substantial changes in control sensitivities and trim positions. Therefore, the pilot's job is very difficult when performing tasks during these specified conditions, and it is possible for these conditions to occur in any flight mission. Pilots are capable of successfully handling the control helicopter with training but in IFR flight, where ground references are no longer accessible, the pilot's job becomes increasingly more difficult. The need to extend military operations during poor weather and at night has been inevitable for military effectiveness throughout history. Therefore, many military missions have been performed during poor visibility and in a low speed environment. However, operations under these conditions endanger flight safety. The roles of search and rescue, submarine detection, mine countermeasures, front line supply, air-to-ground attack, and reconnaissance all demand a low speed, poor weather, day or night capability. Thus, several technological improvements have been made to aircraft to conduct missions without endangering the lives of the aircrew. Today, military helicopters perform a wide variety of tasks in conditions ranging from hot and dry to cold and wet, windy and low visibility weather. Accurate low speed velocity sensing devices are essential because aircraft velocity and position information are what pilots need to perform the aforementioned tasks.

Conventional methods to measure airspeed for aircrafts have been used for over 60 years. Pitot-static systems are still commonly used since they offer simple, low cost and reliable enough solutions to measure airspeed. In this system, airspeed is derived by measuring the difference between the total pressure occurring at the forward facing pitot-probe and the static pressure measured at a static vent [Ref. 12]. Since helicopters fly at relatively lower speeds than aircrafts, and the cruise speed of a helicopter is less than 0.3 Mach, helicopters are used to fly in incompressible subsonic flow conditions where Bernoulli principles pertain. In such flow regimes, airspeed is obtained simply by taking the square root of this pressure difference and multiplying the scale factor.

The flight of a helicopter occurs in the two distinct regimes of hover/low speed flight up to 45 knots, including vertical maneuvering, and mid/high speed flight up to V_{ne} –never exceed velocity- where V_{ne} is the maximum airspeed permitted under any circumstances [Ref. 17]. It is defined as a function of altitude, temperature and gross weight. For example, using flight manual of UH-60A, V_{ne} is found to be 186 knots at -20^o C, 4000 feet and when the gross weight of helicopter is18000 lb. The low speed regime is very much unique to the helicopter as an operationally useful regime. No other flight vehicles are as flexible and efficient at maneuvering slowly close to the ground and at avoiding obstacles. Therefore, the low speed regime is a significant portion of helicopter flight time. In fact, the maneuvers done in this regime make the helicopter invaluable. Although low speed is very critical for helicopters during take-off, landing or hovering, measuring airspeed and wind direction is generally lacking in this regime [Ref. 3]. During low speed flight, the current airspeed measurements systems are inaccurate due to the rotor downwash and limitations of the pitot-static system. The conventional pitot-static sensor is ineffective at airspeeds below 40 knots and does not function at all during rearward flight [Ref. 8]. John Carter explains in Ref. 12 why it does not function properly.

> *The pressure difference in pitot-static system becomes very small at low airspeeds. This creates a practical difficulty in that presently available pressure measuring techniques which are suitable for aircraft application are poor at measuring pressure differentials of less than 1/1200 atmospheres which corresponds to an airspeed of 20 knots.*

Moreover, rotor downwash inevitably enters the pitot probe, which causes a fast error or enters the static port resulting in a slow error. Flow patterns developing around the aircraft during sideward and rearward flight may bring erroneous airspeed indications [Ref. 13].

In low speed flight, required engine power increases due to the difficulty of maneuvers. Due to these high engine and tail rotor anti-torque requirements, extra attention must be paid to directional control margins. Moreover, vibrator loads can occur in some maneuvers, which can result in fatigue damage accumulation in flight critical components [Ref. 4]. The pilot is often required to fly this type of maneuver, and ground references, if available, are mostly used instead of instruments. However, in a combat environment, ground references might not be available which increases the need for accurate measurement systems.

Many developments have been completed since the 1950's to increase the accuracy of measurement systems. One study suggested moving the probes above the rotor hub in order to protect the pitot system from the down-wash effect. Another study suggested using a swivel device mounted above the rotor hub that can measure true airspeed magnitude and direction [Ref. 3]. In this design, two venturi tubes were mounted on opposite ends of a rotating arm to provide a differential measurement between the two sensors. These sensors were used to calculate airspeed and wind direction [Ref. 4]. However, such a system requires a slip ring assembly or a similar means of transferring the data from the rotating system to the body of the helicopter. One other approach is based on the study of wake under the rotor and using a sensor mounted under the rotor to determine the airspeed of the helicopter. In this approach, a 360^0 rotating pitot-static probe was used to measure the true airspeed and wind direction [Ref. 4].

In the late 1970s and early 1980s, Faulkner and Buchner proposed a study based on the idea that the measurement of the control position can be used to estimate airspeed when airspeed substantially affects the control trim position. These researches built a model using simplified thrust and flapping equations to obtain longitudinal and lateral velocity components. However, their analysis requires knowledge of the main rotor flap angle, which is difficult to measure [Ref. 8].

Although some progress has been made because of these studies, none of them have received worldwide acceptance due to system complexity, reliability, and economic and aerodynamic issues.

McCool and Haas describe some other efforts to determine the airspeed analytically [Ref. 3]. One of these efforts is a neural network-based approach, which is the core of this thesis. In that study, first the flight parameters, such as rotor RPM, cyclic position, etc. were obtained during test flights of HH-60J helicopter and CH-46E helicopter and then these parameters were used as inputs into a NN model to predict airspeed and wind direction. By using fuselage parameters, the problem of transmitting a signal from the rotating system to the fuselage is eliminated [Ref. 3]. The results obtained from this study proved that NN could be used to predict airspeed with reasonable accuracy.

A NN implementation is chosen for several reasons. First, it is easy to use. Input parameters can be chosen as quantities, which are commonly measured on a flight data recorder. Second, the combined influence of inputs can be investigated by using multiple hidden layers. Third, a more flexible and empirical estimation can be obtained, as new data becomes available to retrain the network [Ref. 16]. Based on the aforementioned benefits, McCool and Haas showed that the neural network approach provides a mechanically simple and inexpensive alternative to current low airspeed measurement technology [Ref.3].

The primary objective of this thesis is to determine the airspeed of an OH-6A helicopter using an NN implementation with the input data provided from the helicopter simulator model FLIGHTLAB. Our intention is to increase the accuracy of the measurement systems in a cost-effective way by using the NN and a simulator model. Since simulator data is used as inputs to the system, first UH-60A data were obtained from the simulator and used as inputs to the NN. HH-60J and UH-60A helicopters are very similar and they are both variant of Skorsky basic Hawk series helicopters. Therefore a comparison of HH-60J helicopter and UH-60A helicopter results would provide the verification of using simulator data for the NN model. Once the model is validated for the UH-60A helicopter, the next step is to predict the speed of the OH-6A helicopter using the same approach.

II. NEURAL NETWORKS

A. INTRODUCTION TO NEURAL NETWORKS

The human brain is the world's most complex computing device. Thus, the brain's powerful thinking, remembering, learning and problem solving capabilities have been studied and modeled for digital computing. The brain is so complex that scientists have only just begun to understand how it works. Although little is known about the brain and the neural system, modeling the functionality of the brain in a very fundamental manner leads to the creation of neural networks.

The neuron is the fundamental cellular unit of the nervous system and, in particular, the brain [Ref. 6]. The neuron functions like a microprocessing unit, which receives and combines signals from many other neurons. The brain consists of 10^{11} number of neurons. They are connected to each other by approximately 10^4 connections per element or 60 trillion connections total [Ref. 1]. The input path of a cell body is called "dendrites". The output path is called "axon". The axon of a neuron splits up and connects to the dendrites of other neurons through a connection called a "synapse". The cell body, which is also called "soma", sums incoming signals when sufficient inputs are received. It is generally thought that all functions are stored in the neurons and in the connections between them. Learning occurs when new connections are established or existing connections are modified. The structure of a neuron is depicted in Figure 1 [Ref. 10].

In an artificial neural network (NN), the same principles are used to simulate and capture some of the power of the brain and the neural system. In a NN, the unit corresponding to a neuron is called the "processing element (PE)". A PE has many input paths and combines the values of these input paths. The combined input is then modified by a transfer function.

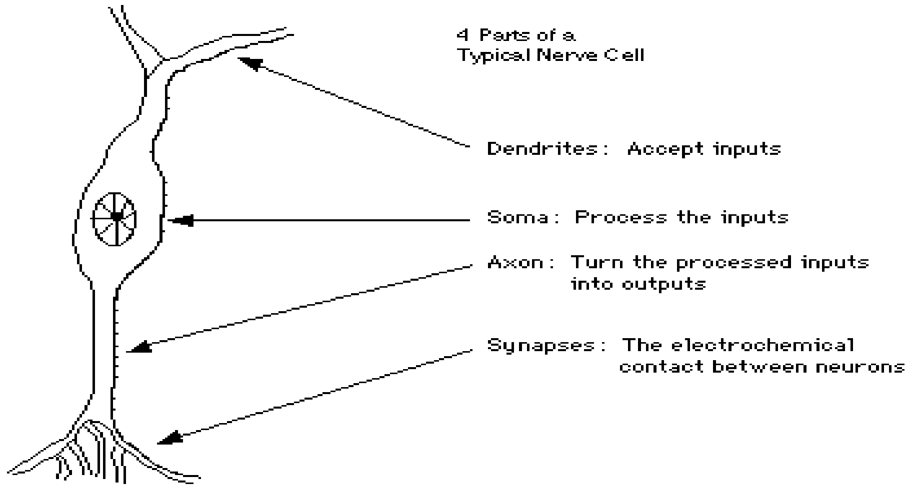


Figure 1. Structure of a Neuron [From Ref. 10].

The transfer function is also referred to as an "activation function". This transfer function can be either a threshold function or a continuous function of the combined input [Ref. 6]. When the transfer function is a threshold function, as when the combined activity level of PE reaches a certain level, the information passes or otherwise is filtered. The output path of the transfer function is passed directly to the output path of PE. The PE output is connected to the other PE input paths through connection weights, as in a neural system.

The NN is trained to perform a particular function by adjusting the values of strength of connections (weights) between elements. The procedure to modify weights and biases of a network is called a learning rule, where the bias corresponds to a weight with a constant input. Note that, biases are sometimes added to add more flexibility to the network configuration. There are two types of learning:

1) Supervised Learning, where sets of inputs and desired outputs (target values) are presented to the network, and the NN configures itself to achieve desired input/output mapping; 2) Unsupervised Learning, where only inputs are shown to the network and NN organizes itself internally so that each PE responds strongly to different sets of inputs

In this thesis, only supervised learning schemes are considered. In that type of learning, a NN generates its own rules by learning from shown examples. Learning is achieved through a learning rule that makes necessary modifications to weights and biases in response to network output and target values. Figure 2 shows how such a NN system works.

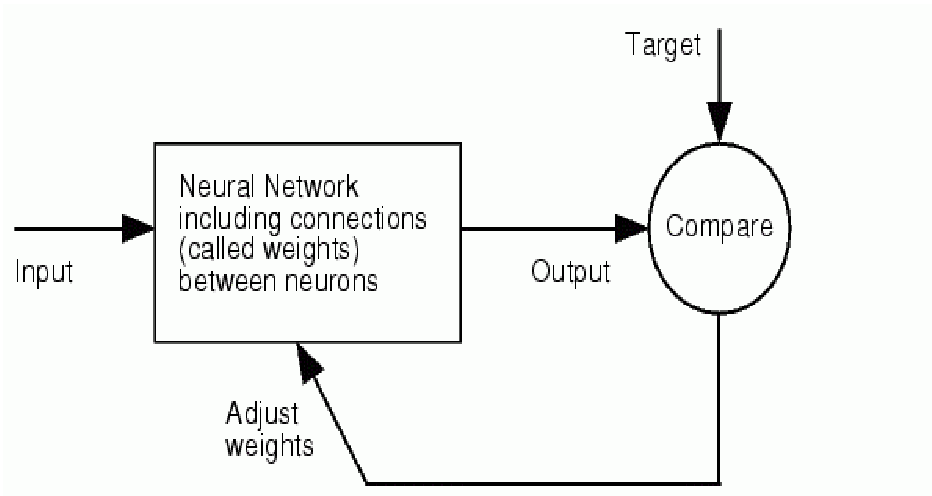


Figure 2. System Diagram of NN [From Ref.2].

A NN consists of many PEs grouped together called layers. A layer is defined as a group of neurons having connections to the same inputs and sending outputs to the same destinations [Ref. 2]. Figure 3 depicts the layer structure of a NN.

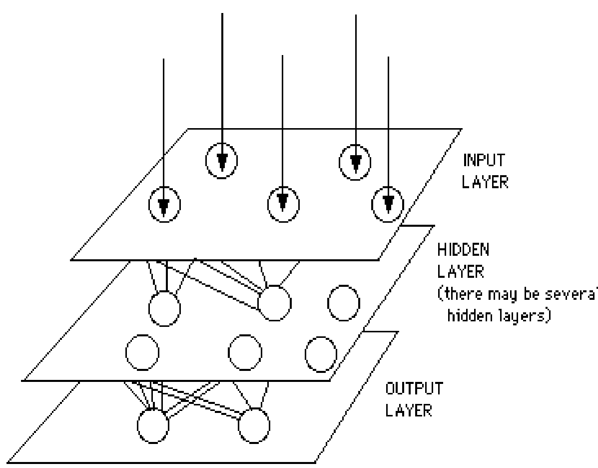


Figure 3. Layer structure of NN [From Ref. 10].

A general NN usually consists of an input layer, one or two hidden layers, and one output layer. The typical structure of a layer with one neuron is presented in Figure 4.

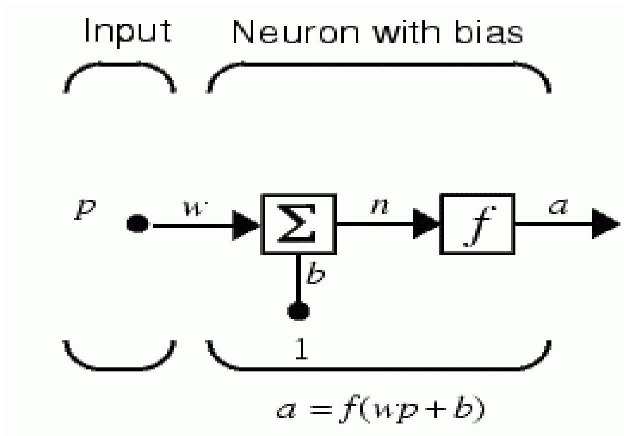


Figure 4. Structure of a layer [from Ref.2].

A layer consists of the following elements:

- An input vector (p),
- Weights (w) represent connection strength,
- Summation (Σ),
- Bias vector (b) represents a column vector of bias values for a layer of neurons,
- Transfer function (*f*),
- Output vector (a) is the output of the network.

The architecture of a NN depends on the number of layers the network possesses, the number of neurons in each layer, each layer's transfer function, and how the layers are connected to each other. Note that there is no unique architecture for any given problem, and the best architecture heavily depends on the data presented to the network, the number of neurons for each layer selected by the user, and the selected activation function. Most practical NNs have just two or three layers. It is certain that the greater number of neurons in the hidden layer, the more powerful the network may potentially be. Note that, however, adding more neurons makes the network more complex and complexity should be minimized since it increases training time. Furthermore, the NN may memorize the training pattern set, but not perform well on data outside of the training set when the ratio of neurons to training patterns is too large. This problem is referred to as overspecialization or overfitting.

B. NEURAL NETWORKS BACKGROUND

Although background studies of NN, extend back to the late 19^{th} and early 20^{th} centuries, modern NN implementations first appeared in 1943 with the works of Warren McCulloch and Walter Pitts. Their researches considered the concept of artificial neurons, which have the capability to compute arithmetic or logical functions. McCulloch and Pitts published watershed paper entitled "A Logical Calculus of Ideas Imminent in Nervous Activity" [Ref. 6].

In the late 1950s, Frank Rosenblatt proposed the perceptron network and the associated learning rule. In 1957, Rosenblatt published the first major research project in neural computing which included the development of the perceptron element. The perceptron is a pattern classification system, which could identify both abstract and geometric patterns. In addition, the perceptron can make limited generalizations and can properly categorize patterns despite noise in the input [Ref. 5]. This study showed the first practical application of NN by demonstrating how NNs can perform pattern recognition. In 1959, Bernard Widraw and Tedd Hoff proposed the Adaline (Adaptive Linear Element), based on simple neuron-like elements and used it to train adaptive linear networks. The Adaline and the two-layer Madaline version were used for a variety of applications including speech recognition, character recognition, weather prediction, and adaptive control. Widraw used the adaptive linear element algorithm to develop adaptive filters that eliminate phone line echoes, in the first real life NN application [Ref. 6].

In the mid-1960s, Marvin Minsky and Seymour Papert, considered the NN potential limitations. They showed that these already known networks could handle linearly separable problems only and were usually not appropriate for real life applications. As a result, NN research faded for a while.

In the 1970s, Kohonen, Grossberg and Anderson proposed the Kohonen Network and the Self–organizing Network. Kohonen introduced the concept of the competitive learning rule in which PEs compete to respond to an input stimulus and the winner adapts itself to respond more strongly to that stimulus. This type constitutes an unsupervised learning process and the internal organization of the network is governed only by input stimuli [Ref. 5]. Grossberg's contribution was a wealth of research towards the design and construction of neural model as he used neurological data to build neural computing models. Anderson developed a linear model, called a linear associator. Which is based on models of memory storage, retrieval and recognition. In addition, Anderson improved the model by combining it with a nonlinear post-processing algorithm, which is used to clean up spurious responses. This model is called Brain-State-in-a –Box [Ref. 5].

In the 1980's, NN became popular again with the back-propagation algorithm for training multilayer perceptron networks. The concept of back-propagation algorithm was presented by several researchers, such as David Parker, Yaun LeCun, David Rumelhart, James McClelland, and Geoffrey Hinton. While a perceptron network is only capable of solving linear problems, back-propagation network can solve more complex nonlinear problems. This significant capability made the back-propagation networks the most widely used networks.

In 1982, John Hopfield presented a paper describing his neural computing system called the "crossbar associative network" or known as the "Hopfield Model". This model represented a neuron operation as a thresholding operation and illustrated memory as information stored in the interconnections between neuron units. He also illustrated and modeled the brain's ability to call up responses from many locations in response to a stimulus. Thus, this model represents how a NN associates information from many storage sites for a given input [Ref. 6]. In the 1980s, the Bi-directional Associative Memory (BAM) network, Boltzman Machine, the General Regression NN, and the Learning Vector Quantization Network were developed, in addition to the back-propagation and Hopfield models.

Although the concept of NN has been around for about 50 to 60 years, most applications have appeared in the last fifteen years and the field is still developing very rapidly. NNs can be found in many fields ranging from aerospace to medicine, banking and robotics. Given the work done and range of applications, NN will most likely be a permanent fixture not only as a solution to everyday problems but also as a tool to be used in appropriate situations. It is certain that the more the structure of the brain is understood, the more advances there will be in NN.

C. INTRODUCTION TO THE BACK-PROPAGATION NEURAL NETWORK

Although in 1974 Paul Werbos described a similar algorithm to train multilayer network, Back-propagation (BP) was invented independently by David Parker, Yaun LeCun, David Rumelhart, Geoffrey Hinton and Ronald William in the mid 1980s.

Back-propagation is a very popular NN learning algorithm to train multilayer networks with differentiable transfer functions to perform function approximation, pattern association and classification. The algorithm is named for how it handles errors in the network. The derivatives of the network error with respect to the network bias and weights are computed starting from the very last layer up to the first layer in order to modify the weights and biases of the network. The process starts from the output layer and goes back to the first layer and is therefore called the "Back-Propagation Network".

Typically, back-propagation network has an input layer, one or two hidden layers and one output layer. Although there is no limit on the number of hidden layers, generally one or two hidden layered networks are selected due to the complexity of the resulting system. An example of a typical network is shown in Figure 5.

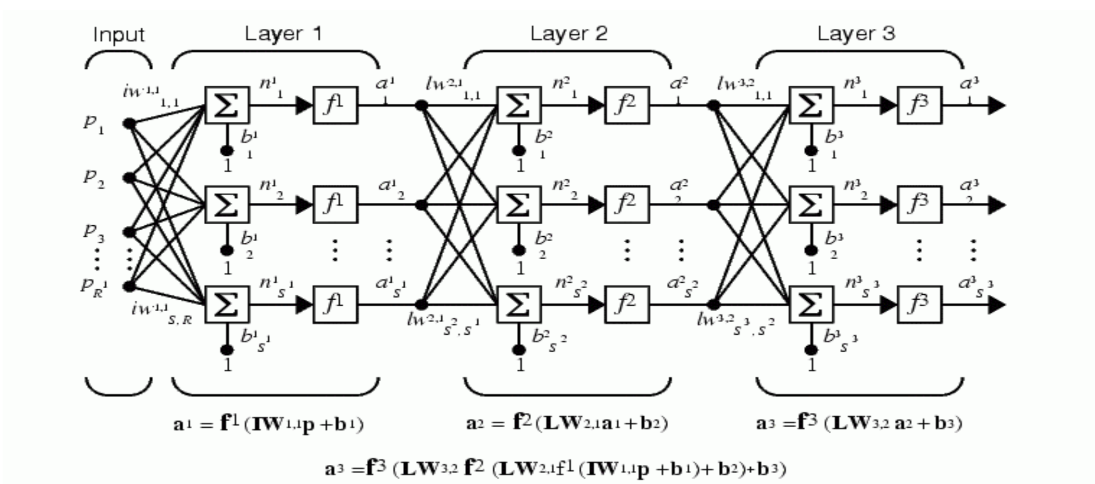


Figure 5. A Typical Back-Propagation Network [From Ref. 2].

A BPNN is typically represented with the following notation: R-S^1-S^2-S^3, where R is the number of inputs and S^i is the number of neurons at layer i. In Figure 5, the input vector to the network is shown by P_i. Transfer function is represented by f^i and b^i represents bias. Depending on the selected transfer function, the BPNN can be used to solve linear or nonlinear problems. In order to solve a nonlinear problem, a nonlinear transfer function should be selected.

The first step in BPNN is to transfer the input forward through the network. The output of one layer becomes the input to the following layer. Thus, the output of a network can be defined as

$$a^{M} = f^{M}(w^{M} * a^{(M-1)} + b^{M}) \ , \tag{2.1}$$

where M = 1,…….., R and R is the number of layers in the network. In Figure 5, for the first layer, $a^{(M-1)}$ is $a^{(0)}$, which is the input of the network, and $a^{(3)}$ is the output of the last layer.

The second step is to propagate the errors, or in other words, sensitivities backward through the network from the last layer all the way to the first layer. The BPNN algorithm is a generalization of the LMS algorithm, which uses the mean square error as a performance index. As each input is applied to the network, the output, a, is compared with the associated target value, t. The algorithm then adjusts the network parameters in order to minimize the mean square error [Ref. 1].

Thus, given the error is defined as:

$$e = [t - a] \ , \tag{2.2}$$

and the expectation of mean square error is defined as

$$f(x) = E([e^{T}e]) = E[(t-a)^{T}(t-a)], \tag{2.3}$$

the expectation of the squared error at iteration k becomes [Ref. 1]

$$f(x) = E[(t(k)-a(k))^{T}(t(k)-a(k))]. \tag{2.4}$$

The algorithm adjusts weights and biases as follows [Ref. 1]

$$\mathrm{w}_{i,j}^{M}(k+1) = \mathrm{w}_{i,j}^{M}(k) - \alpha \frac{\partial f}{\partial \mathrm{w}_{i,j}^{M}} \ , \tag{2.5}$$

$$b_{i}^{M}(k+1) = b_{i}^{M}(k) - \alpha \frac{\partial f}{\partial b_{i}^{M}} \ ,$$

where α is the learning rate. This shows that the weights at any given iteration are equal to the weights at the previous iteration adjusted by some fraction, α, of the sensitivity of the error to that weight. In other words, the weights at each iteration are adjusted in a way that reduces the error at the previous iteration.

The selection of α is usually done by trial and error. Too large an α often leads to divergence of the learning algorithm while too small an α results in a slow learning process [Ref. 9].

Note that the above equations include partial derivatives. Since the error is an indirect function of the weights in the hidden layer, a chain rule is used to compute these partial derivatives. These partial derivatives are called sensitivities and defined as:

$$s_i^M = \frac{\partial f}{\partial n_i^M} \quad . \tag{2.6}$$

In the above equation s_i is called the sensitivity of f to changes at layer M in the i^{th} element of the net input.

The next step is to propagate the sensitivities backward

$$s^{M-1} = \dot{f}^{M-1} n^{M-1} (w^M)^T s^M , \qquad \text{for M = R,....,1} \quad . \tag{2.7}$$

The final step is to adjust weight and biases with respect to these sensitivities, which leads to

$$w^M (k+1) = w^M (k) - \alpha s^M (a^{M-1})^T , \tag{2.8}$$

$$b^M (k+1) = b^M (k) - \alpha s^M \; .$$

Although the BP network is a very powerful technique, it usually requires a long training time. However, using some tricks and heuristics can often help to obtain an efficient BPNN implementation [Ref. 9].

In summary, the BPNN is a commonly used technique for solving nonlinear estimation problems, provided that a nonlinear transfer function is selected. The BP algorithm is based on a LMS algorithm that minimizes the squared error, where the chain rule is used to compute the derivatives of the error with respect to weights and biases in the hidden layers. The methodology in the BP network is a three-step process. First, the input is propagated forward through the network and then the sensitivities are computed. Second, starting from the last layer, these sensitivities are propagated backward through the network. Finally, weights and biases are updated using these sensitivities.

III. DEVELOPMENT OF NEURAL NETWORK MODEL

A. NEURALWORKS PROFESSIONAL II/PLUS SOFTWARE

The model built for this analysis is derived from the NeuralWorks simulation software program presented by NeuralWare, Inc. NeuralWorks is a user-friendly program that makes it possible to not only select network parameters easily and quickly but also to present network results effectively. Many NN algorithms such as BPNN, Radial Basis Function NN, and LVQ, are included in the software. Once the algorithm type is chosen, the next step is to define the network architecture, which includes specifying the number of layers and the number of PEs associated with each layer, etc. Several types of learning rules and PE transfer functions are embedded in the program. NeuralWorks also allows the user to select learning rates and momentum terms. In addition, the extensive and powerful instrumentation and diagnostic package allows the user not only to monitor and adjust network parameters but also to display weights, errors, classification rates and confusion matrices in graphical formats. Moreover, users can display the networks through network or Hinton diagrams. These specifications make NeuralWorks Professional II/Plus useful in designing, building, training, testing and deploying neural networks to solve complex, real-world problems [Ref. 5].

The following picture shows a typical window of the Neural Works Professional II/Plus software:

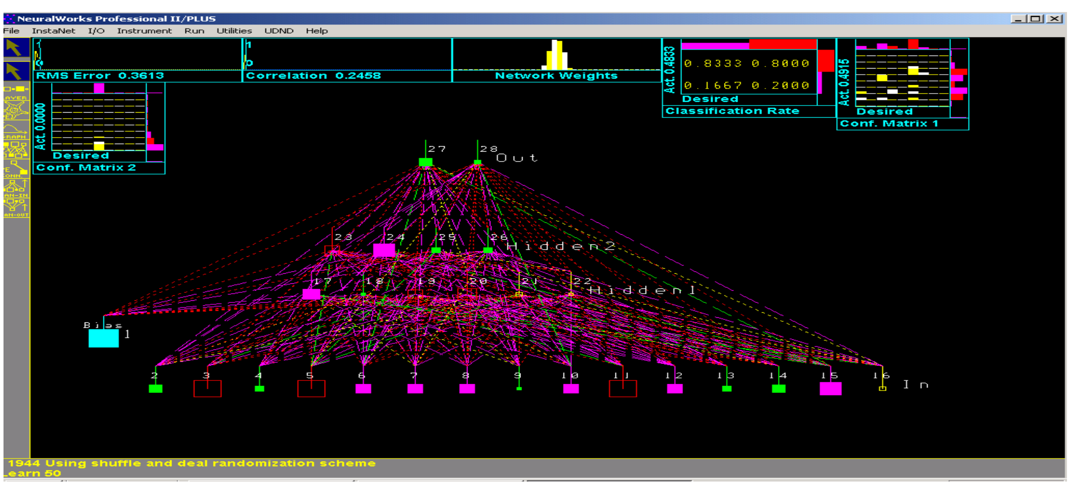


Figure 6. Typical Window of Neural Works Professional II/Plus Software.

B. FLIGHTLAB SIMULATOR

Flight simulation tools have been used extensively not only for training and evaluation of aircrew members but also for the design and analysis of aerospace vehicles. The need to use flight simulation systems instead of real aerospace vehicles depends on the following reasons:

- The complexity of aircraft systems,
- High operating costs of aircrafts,
- The limitations of operating environment of aircrafts,
- Technological improvements in flight simulators.

In addition, flight simulators provide safe and effective conditions for training purposes as instruction, demonstration and practice of certain maneuvers and procedures that cannot be done during real flight conditions may easily be performed in a simulated environment. Finally, longer training periods can be tolerated due to the low operating cost of simulators. Therefore, modern training procedures benefit from simulation tools extensively.

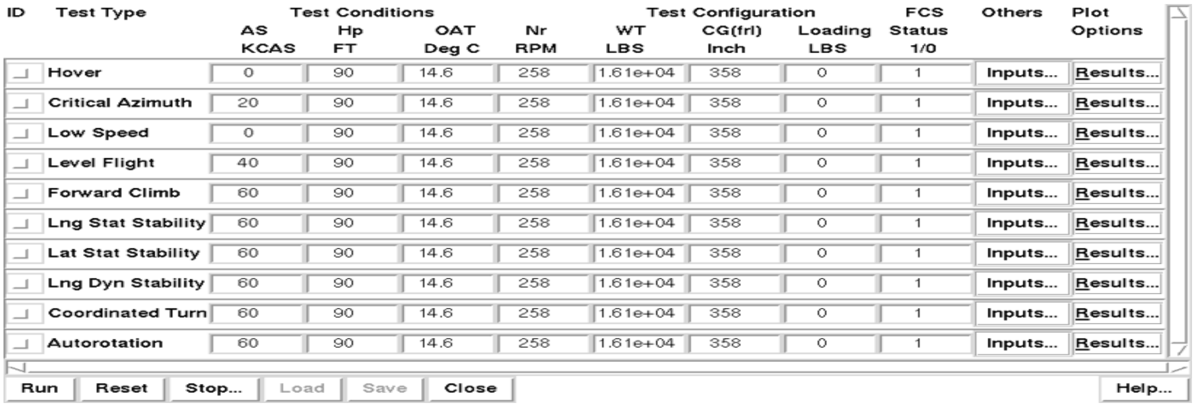
Simulators also play an important role in engineering processes such as design and analysis of aircraft. It is certain that the dynamics of rotary-wing aircraft is much more complex than those of fixed-wing aircraft. Consequently, helicopter simulation systems require high computational power to include a complex set of computer programs, very powerful and expensive computer systems and full motion based simulation devices. Recent improvements in computer and simulation technology make simulators able to produce data as correct as real helicopters. These improvements make it possible to make use of simulators for design, test and analysis purposes.

FLIGHTLAB is a commercial-off-the-shelf type of software product developed by Advanced Rotorcraft Technologies, Inc (ART) for the development and operation of flight vehicle dynamics models in simulation applications especially for helicopters. FLIGHTLAB is a tool that allows a user to build each section of a model separately and then combine the pieces under the same framework similar to building a model by using a finite element approach [Ref. 14]. This approach is very useful for engineering studies. In order to provide a desktop pilot interface for FLIGHTLAB models, ART has developed PilotStation, a turnkey executive for real time simulation operations that couples image generation and pilot control inputs with the FLIGHTLAB flight dynamics model to provide a low cost, pilot-in-the-loop simulation. PilotStation can also be used with FLIGHTLAB code generated models to provide a low cost, real time simulation capability using PCs [Ref. 7]. PilotStation, which can be run on UNIX or LINUX operated computers, processes the high fidelity rotorcraft model and generates the cockpit gauges and the window displays.

FLIGHTLAB uses a simulation language called Scope. This language is an interpretive language that uses the industry standard MATLAB syntax, coupled with new language constructs and a combination of C and Fortran computer languages to make the building, testing and solving of nonlinear dynamic rotorcraft simulation models easy [Ref. 7].

Another feature of the FLIGHTLAB Simulation System is Xanalysis. This tool makes it possible to design, test and analyze rotorcraft models by allowing the user to modify model parameters and perform a wide range of analyses on design alternatives. A set of predefined test scenarios carry specific rotorcraft analyses such as performance, stability and control, handling qualities and aerodynamic and structural loads. In addition to these features, the simulation can be automatically configured to interact with the test vehicle's configuration and test conditions while using the time history of the test flight's control inputs to implement the simulation [Ref. 7]. These features are the required characteristics necessary to implement the model built for this thesis and make use of data provided by a simulator instead of a real test flight.

The following figure depicts the X-Analysis ``Flight Test Utility'' [Ref. 7]:



ID	Test Type	Test Conditions AS KCAS	Hp FT	OAT Deg C	Nr RPM	Test Configuration WT LBS	CG(frl) Inch	Loading LBS	FCS Status 1/0	Others	Plot Options
	Hover	0	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...
	Critical Azimuth	20	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...
	Low Speed	0	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...
	Level Flight	40	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...
	Forward Climb	60	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...
	Lng Stat Stability	60	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...
	Lat Stat Stability	60	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...
	Lng Dyn Stability	60	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...
	Coordinated Turn	60	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...
	Autorotation	60	90	14.6	258	1.61e+04	358	0	1	Inputs...	Results...

Figure 7. Xanalysis Window [From Ref. 7].

C. SELECTING DATA AND BUILDING THE MODEL

In the early 1990s, Kelly M. McCool and David J. Haas at Naval Surface Warfare Center, Bethesda, MD, developed a study of UH-60 helicopter airspeed estimations using NN in a low speed environment. McCool and Haas built and implemented a BPNN with two hidden layers and 25 processing elements in each layer. The data used as input for the NN were obtained from actual test flights performed at the Naval Air Warfare Center, Aircraft Division, Patuxent River, MD. [Ref. 4]. Their study was used as a starting point for this thesis since similar analysis was performed but with different data sets. They used real flight test measurements as a source of data for their model. FLIGHTLAB simulator outputs were used for NN input for this thesis. In order to evaluate how well the simulator performs and how well the relationship between the simulator parameters and low airspeed is developed, the simulator was first run for the UH-60A helicopter to compare the results of the present study and the results of the actual test flights. The simulation data was analyzed using a similar network architecture and network parameters, to make a fair comparison. The goal of this approach is to determine whether using a simulator instead of a real test flight is feasible or not.

The simulator was run for UH-60A helicopter from hover to 50 knots with 5 knots intervals at various gross weights ranging from 16000 to 24000 lbs with 1000 lbs intervals and at various sideslip angles from 0 to 360^0 with 30^0 intervals to obtain the input data for the NN model. However, sideslip angles were varied from 300^0 to 60^0 for velocities 35 knots and above because of the limitations of the UH-60A helicopter since the helicopter cannot fly rearward or sideward when the speed is above 30 knots. The altimeter was set to 85 feet and AGL to obtain the results when the helicopter is out of the ground effect and at level flight. In a ground effect, analysis was performed by setting the altimeter to 20 feet AGL for the UH-60A helicopter. The wind was assumed to be zero for this analysis. Fifteen parameters were chosen as inputs for the neural network model based on the variables described above, and are shown in Table 1.

MODEL VARIABLES FOR UH-60A HELICOPTER	
Airspeed	0 - 50 Knots (5 Knots intervals)
Gross weight	16000 - 24000 lbs (1000 lbs intervals)
Sideslip angle	0- 360 degrees and 300 - 60 degrees (30 degrees intervals)
Altimeter (AGL)	20 ft (for in ground effect), 85 and (for out of ground effect)
Wind Speed	0
Pressure Altitude	Sea level
NN MODEL INPUT PARAMETERS	
1.	Cyclic position (Lateral)
2.	Cyclic position (Longitudinal)
3.	Collective position
4.	Pedal position
5.	Roll rate
6.	Pitch rate
7.	Yaw rate
8.	Pitch attitude
9.	Roll attitude
10.	Altimeter
11.	Climb rate
12.	Main Rotor Blade RPM
13.	Engine torque
14.	Gross weight
15.	Sideslip angle

Table 1. Neural Network Input Parameters.

These fifteen parameters, which define the condition of the related part of the helicopter at certain settings, are the outputs of the FLIGHTLAB simulator. The data was split into two sets, one for training the network (training data set) and the other one for evaluating the network performance (test data set). The first set used for training includes the data related to airspeeds at 0, 10, 20, 30, 40, 50 knots at all gross weights and sideslip angles. The second set used for testing was formed with the remaining input data.

The type of network was selected, after setting up training and test data sets. A Back-propagation approach was selected as it was used before with success [Ref. 3]. A Radial Basis Function Network (RBFN) was selected as an alternative network because it can be used in most applications where back-propagation approaches may be used. However the RBFN trains faster and leads to better decision boundaries than a BPNN in many classification and decision problems. The learning rule used in RBFN is an unsupervised learning rule. The results of these two models are presented in Chapter IV.

The next step in the BPNN scheme involved the selection of number of layers and neurons in each layer. First the model in Reference 3 was used as a starting point, since the first goal of this study was to compare the results of the model implemented by real flight data and the model by simulator data. Thus, we selected two hidden layers and 25 neurons in each hidden layer, resulting in a 15-25-25-1 BPNN structure. Several models, were also studied; 14-25-25-1, 15-18-25-1, 14-14-14-1, 14-14-12-1, and 14-14-10-1.

Next, the learning coefficient and momentum term were determined. Learning coefficients control the changes in size of weights and biases during learning. Setting an appropriate learning rate is significant because a small learning coefficient leads to very little learning, which increases the training time. However, a large learning coefficient may cause the performance index to diverge, meaning that no learning occurs. Therefore, a small learning rate is generally used to avoid divergence. Next, the momentum term is selected. The momentum term helps to obtain faster learning when using a low learning rate. In this study, the appropriate values for these two terms were determined by trial and error.

Finally, the learning rule and transfer function type were selected. There are several learning rules available in NeuralWorks, such as, the Extended Delta-Bar-Delta

(Ext. DBD), Normalized Cumulative Delta (NCD), Quickprop, Delta Bar Delta (DBD) and Delta Rule (DR). In this study, several types of learning rules were analyzed for the UH-60A helicopter and the results are presented in Chapter IV. Based on the results of the UH-60A helicopter model, only Ext. DBD and NCD learning rules were used for the analysis of the OH-6A helicopter. NCD is a learning rule, which is immune to changes in the epoch length, where an epoch is the number of sets of training data presented to the network between weight updates. The Ext. DBD rule takes momentum term into consideration. The transfer function is used to map the output of a neuron or a layer to its actual output [Ref. 2] and may be linear or nonlinear. The most commonly used transfer functions are depicted in Table 2 [Ref. 1].

NAME	INPUT/OUTPUT RELATION
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$
Linear	$a = n$
Log-Sigmoid	$a = \dfrac{1}{1+e^{-n}}$
Hyperbolic Tangent Sigmoid	$a = \dfrac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons

Table 2. Commonly Used Transfer Functions.

The sigmoid transfer function is commonly used in BPNN implementations because it is differentiable. NeuralWorks provides the hyperbolic tangent and the sine function as alternative functions. The hyperbolic tangent function is a bipolar version of the sigmoid function. The sigmoid function maps the output between 0 and 1 smoothly whereas the hyperbolic tangent function places it between –1 and 1. In this work, the hyperbolic tangent transfer function (Tanh) is used.

Once the feasibility of this approach had been verified, the same type of architectures and NN model parameters, were used for the OH-6A helicopter. These parameters were the same as those given in Table 1, except for the engine torque parameter. The only difference utilized for the second part of the analysis was the altimeter setting. The altimeter was set to 12 feet for in ground effect data and 100 feet for out of ground effect data for the OH-6A helicopter. Besides the altimeter setting, other parameters, such as gross weight and pressure and altitude, were adjusted for the OH-6A helicopter. Gross weight for OH-6A was defined ranging from 1500 lb. to 2500 lb. with 100 lb. increments. Pressure altitude was set to 90 ft for sea level and 6000 ft for high-level altitude. For both pressure altitudes, in ground effect analyses were performed at 12 ft, while out of ground effect analyses were performed at 100 ft.

IV. ANALYSIS AND RESULTS

The NN-based approach to determine helicopter low airspeed developed by Haas and McCool has proven to be quite promising. In their study, a BPNN configuration with the Extended Delta Bar Delta learning rule, 2 hidden layers and 25 PEs per layer was selected [Ref. 3]. The data used as input to the NN were obtained from actual test flights performed at the Naval Air Warfare Center, Aircraft Division, Patuxent River, MD. Based on the model they implemented, and using the methodology described in the previous section, in this thesis the NN model was first developed for the UH-60A helicopter by using simulator data. The first model architecture developed for this study was approximately the same model as that described in Ref. 3, since most of the network parameters such as number of layers and number of neurons are determined by trial and error. The first network selected was a 14-25-25-1 BPNN with the Ext. DBD learning rule, and did not include the sideslip angle parameter as input to the network. Various architectures were subsequently tried with different learning rules. The momentum function was varied from 0.2 to 0.6 with the best results obtained at 0.4. Therefore, all OH-6A analyses used a momentum function to 0.4. The NN output error airspeed was measured in knots at 1 σ and in terms of the root mean square (RMS) error, as reported by the NeuralWare software. The number of learning iterations was set to 50000 since the network stabilized after approximately 20000 iterations for most of the implementations.

A helicopter flying close to the ground requires less power than when it is flying far from the ground. The proximity of the ground to the rotor disk constrains the rotor wake and reduces the induced velocity at the rotor, which causes a reduction in the power required for a given thrust. This is called ground effect. A helicopter can hover in ground effect at a higher gross weight or altitude than when it is out of ground effect. However, in forward flight, the effect of the ground diminishes with the forward speed. Data sets of both helicopters were obtained by running the simulator for two flight conditions: in-ground effect (IGE) and out-of-ground effect (OGE). These data sets are called single condition data sets. After training and testing with the single condition data sets separately, the sets were combined for a baseline data set. The networks were retrained and tested with the combined data set.

The OGE single condition data set has a dimension of [1144x14] for the UH-60A. The IGE single condition data set has a dimension of [1144x15]. The difference is due to the fact that the UH-60A OGE data did not include the sideslip angle. Sideslip angle was included for the UH-60A IGE and combined data sets. For the OH-6A helicopter type, the sideslip angle was included in all data sets, but the engine torque was not available, so all OH-6A single condition data sets have dimensions of [1144x14]. The single condition data sets were split into training and testing sets. Each training set is a subset of its single condition data set with a dimension of [638x14] or [638x15]. Each testing set is made up of the remaining elements of the single condition set and has a dimension of [506x14] or [506x15]. Various networks were trained separately using these single condition data sets. Later, the single condition data sets were combined forming a baseline data set with a dimension of [2288x15] for the UH-60A and [2288x14] for the OH-6A. A baseline training set was formed as a subset of the combined set with a dimension of [1276x15 or 14]. The remaining elements of the baseline set formed the baseline testing set with dimensions of [1012x15 or 14]. The networks were retrained using these combined training sets. Combined and single condition, OGE and IGE, results were obtained again from each network that was trained with these baseline data sets. Finally, NN model results were exported to MATLAB to develop the following tables and figures.

Each figure consists of four windows. The first window shows the relationship between gross weight and the predicted speed related to that particular gross weight. The second window shows the range of the predicted speed for each actual speed. The third and fourth windows display the relationship between predicted speed and the sideslip angle. In the third window, the speed range is from 0 knots to 30 knots, whereas in the last window, the speed varies from 35 knots to 50 knots. In order to make the figures more readable and easier to distinguish the NN predictions from one another, adjacent speeds are illustrated with a different marker. NN outputs for 5 knots, 15 knots and 45 knots are presented with circles while outputs for 15 knots and 35 knots are demonstrated with triangles. The structure of the NN, learning rule, helicopter type, RMS error and altitude information about that figure is provided in the figure label.

Each table corresponds to a different NN type and consists of five columns. The first column presents actual airspeed and target values for the NN model. The second column gives the mean value of the NN outputs related to each target speed. The third column shows airspeed errors at 1 σ in terms of knots. This error shows the standard deviation (SD) around the mean and it is computed by the MATLAB function "std". The next column demonstrates the error percentage at 1 σ relative to the target speed. It is computed by the following formula:

$$\text{Percent Error} = \frac{\sigma_i}{V_i} * 100\,, \tag{4.1}$$

where σ_i is the error SD related to that speed and V_i is the actual speed. The last column shows the absolute maximum error of the NN predicted speeds.

Results obtained for analyzed architectures follow in the next section.

A. ANALYSIS OF THE UH-60A MODEL

1. Out of Ground Effect (OGE) Analysis

This section shows the results of OGE analyses by setting the altimeter of the simulator to 85 feet AGL. The network was trained using this single condition data set obtained for this altitude. The results of OGE analysis using the combined data set are shown in the UH-60A Baseline Analysis section.

a. 2-Hidden Layer BPNN

(1) 14-25-25-1 Ext. DBD. Results are shown in Figure 8 and in Table 3. The RMS error is 0.0593. Although the network performance on the training data is quite good, the test results are not close enough to the target speeds. Note that, the NN produced a maximum speed error of 3.51 and 3.86 knots at 5 knots and 45 knots respectively. A 0.7017 knots airspeed error SD was achieved by this setup. The percentage of error is worst at 5 knots, where it is 8.5%, while it is about 3 % for other speeds.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7017			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.3322	0.4249	8.4979%	3.5125
15	16.1533	0.4601	3.0675%	1.8307
25	23.8216	0.5045	2.0179%	2.2057
35	36.2752	1.0928	3.1222%	3.1262
45	46.2789	1.4228	3.1617%	3.8625

Table 3. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; Ext. DBD learning rule.

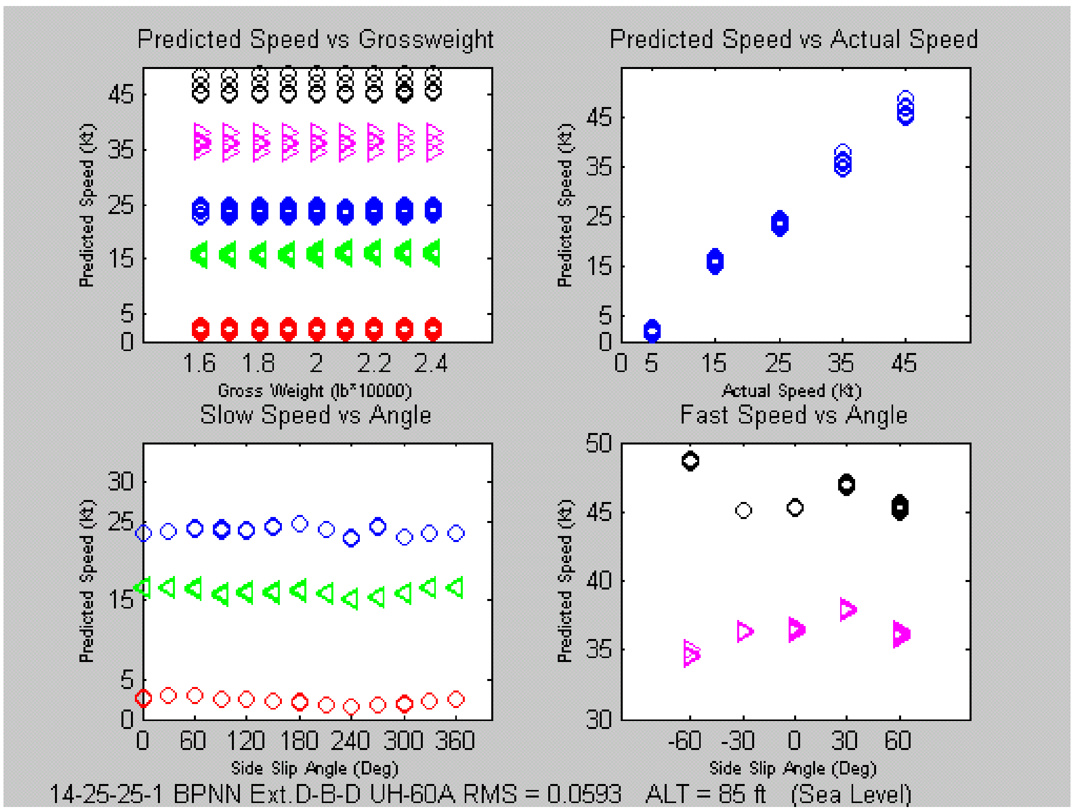


Figure 8. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; Ext. DBD learning rule.

(2) 14-25-25-1 NCD. Results for this configuration are shown in Table 4 and in Figure 9. The RMS error is 0.0658 with the NCD learning rule. The maximum value of the predicted speed error is ± 3.8584 knots. The speed prediction accuracy within ± 1 σ is 0.7506 knots. The airspeed error SD at speeds 25, 35 and 45 knots is about 1 knot or 3 %. For speeds equal to 5 and 15 knots, the error percentage increases to 6 %.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7506			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.7932	0.3221	6.4425%	2.7872
15	17.0481	0.8209	5.4726%	3.0500
25	23.3043	0.6470	2.5878%	2.5837
35	36.8160	1.0996	3.1417%	3.8584
45	46.2666	1.1273	2.5051%	3.0720

Table 4. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; NCD learning rule.

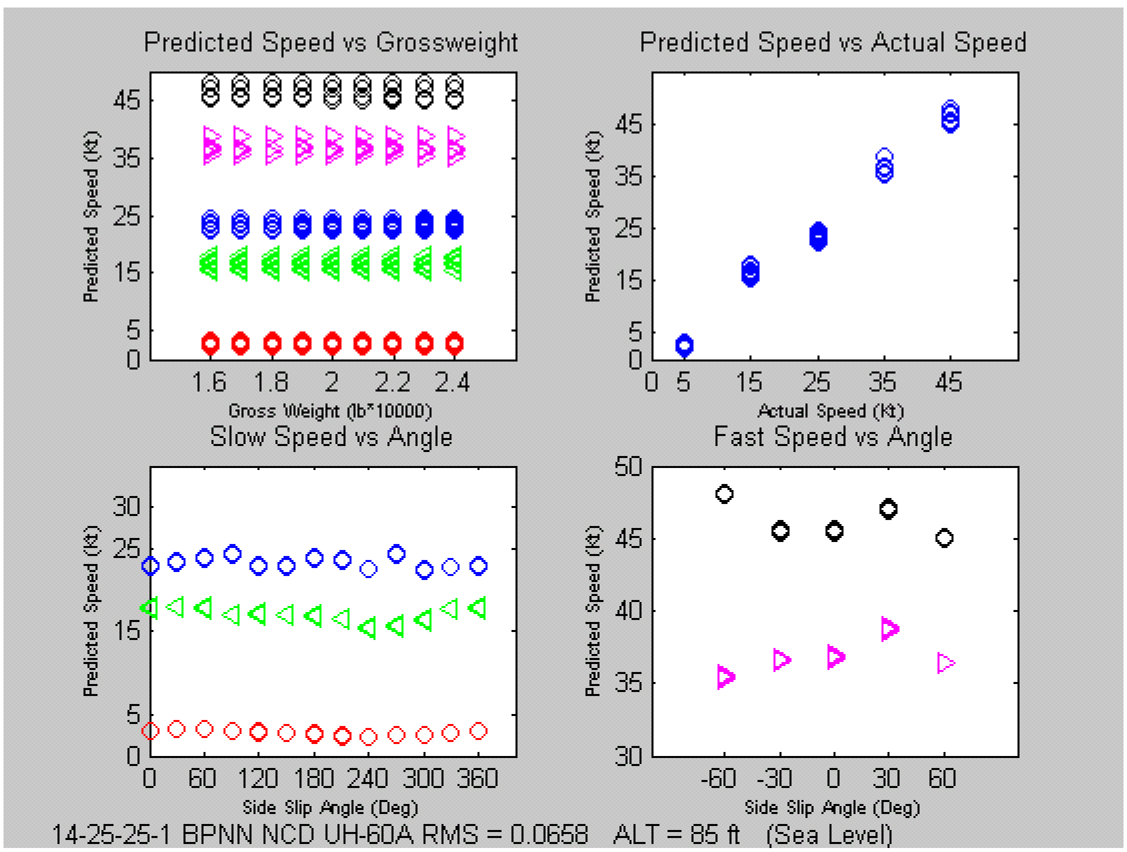


Figure 9. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; NCD learning rule.

(3) 14-25-25-1 DBD. Figure 5 and Table 10 summarize the findings for this configuration. Results show the RMS error for this learning rule is 0.0721. This model predicts the speed very accurately when the actual speed is 45 knots with a mean value of 45.19 knots, corresponding to a speed error prediction rate of 1.42%. However, when the speed is 15 knots and 35 knots the maximum error in the prediction increases significantly to 4.9681 knots, while the error SD goes up to 1.1142 knots. The total airspeed error for the DBD configuration is equal to 0.7544 knots.

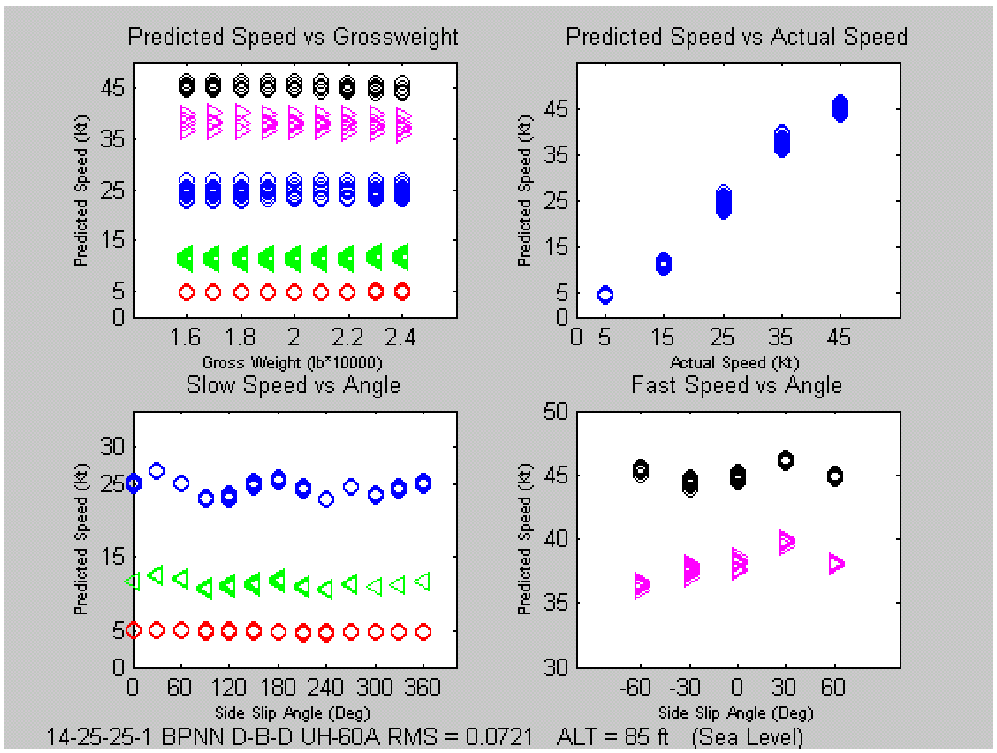


Figure 10. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; DBD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7544			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	4.8389	0.1628	3.2559%	0.4681
15	11.3654	0.5467	3.4647%	4.5417
25	24.5698	1.0986	4.3944%	2.2028
35	38.0388	1.1142	3.1834%	4.9681
45	45.1962	0.6442	1.4227%	1.2996

Table 5. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; DBD learning rule.

(4) 14-25-25-1 QUICKPROP. Figure 11 and Table 6 present results obtained for this configuration. The RMS error for the quickprop learning rule is 0.0711. The maximum predicted speed absolute value error is 4.6189 knots. Note that the results are very close to those obtained for the DBD configuration. Although the mean airspeed value at 15 knots is significantly lower than that of the target speed, the error SD is the largest at 25 knots. The overall network airspeed error SD is 0.7326 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7326			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	4.5007	0.1677	3.3546%	0.8191
15	11.4068	0.6067	4.0444%	4.6189
25	24.8524	1.1168	4.4673%	2.3644
35	37.9663	0.8684	2.4811%	4.4915
45	44.6973	0.5294	1.1764%	1.1524

Table 6. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; Quickprop learning rule.

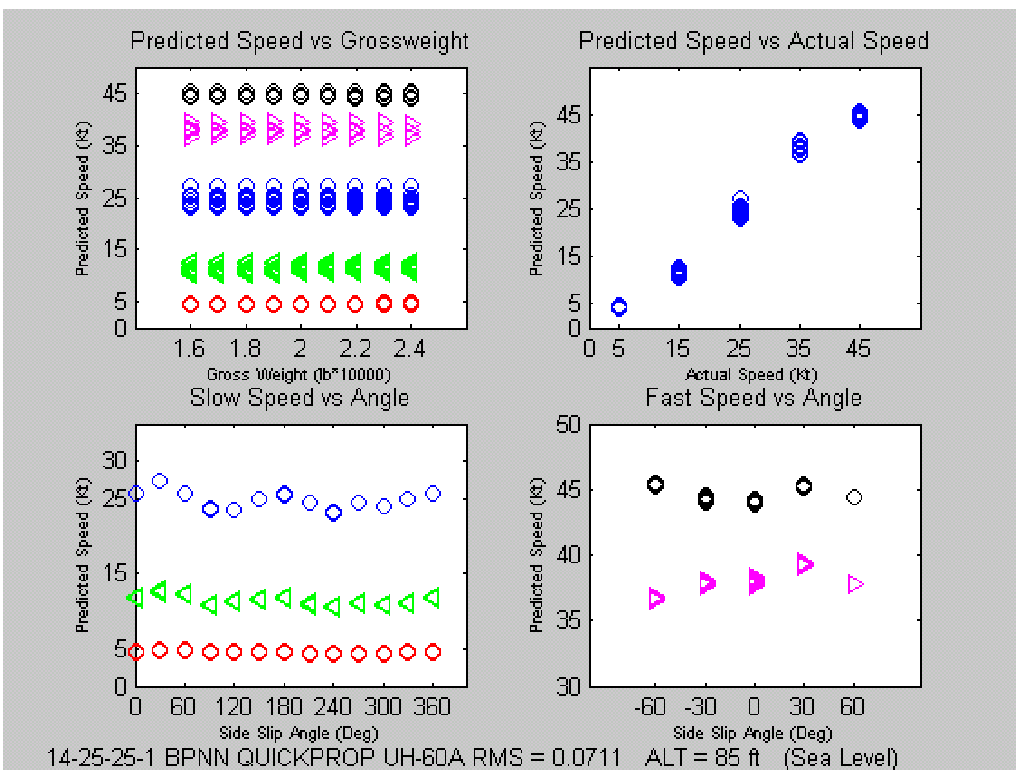


Figure 11. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; Quickprop learning rule.

b. 1-Hidden Layer BPNN

Several one hidden layer BPNNs were implemented; the configurations considered were 14-8-1, 14-10-1, 14-12-1, 14-20-1, 14-22-1, 14-25-1, and 14-50-1. Results for best performing networks are shown below.

(1) 14-20-1 NCD. Table 7 and Figure 12 summarize the findings for this configuration. The 14-20-1 NCD model performs quite well for the velocity of 25 knots with the error SD of 0.5 knots and percent error of 2 %. For all other velocities, the error percentage is 4 %. The RMS error is found to be 0.0407. The maximum error is 3.3 knots at the speed of 5 knots and 35 knots. However, the accuracy of prediction is better than many 2-hidden layer BPNN. The overall error SD is ± 0.8394 knots.

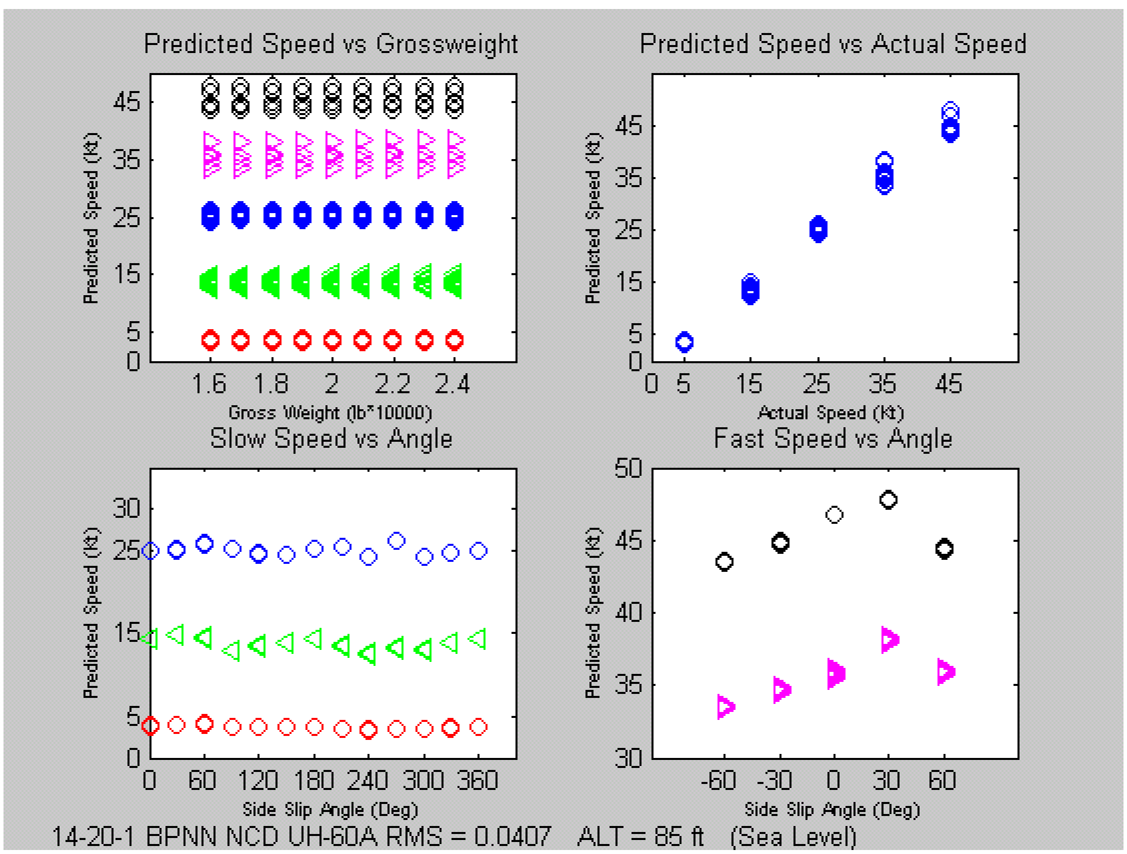


Figure 12. Results for the UH-60A helicopter at 85 ft.; network configuration 14-20-1; NCD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.8394			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.6992	0.2008	4.0170%	3.3712
15	13.7729	0.6762	4.5079%	1.6789
25	25.0478	0.5373	2.1494%	1.2362
35	35.6197	1.5618	4.4623%	3.3712
45	45.4842	1.5915	3.5366%	2.8769

Table 7. Results for the UH-60A helicopter at 85 ft.; network configuration 14-20-1; NCD learning rule.

(2) 14-20-1 Ext. DBD. Table 8 and Figure 13 summarize the findings obtained for this configuration. Results show the RMS error is 0.0657 and that the error SD increases along with the speed. The maximum error is 3.7970 knots for a 35 knots speed. The error SD is also the highest at this speed. However, results show a significant degradation of the airspeed prediction in % at 5 knots. The overall error SD is 0.7002, which is quite better than that obtained with the other networks.

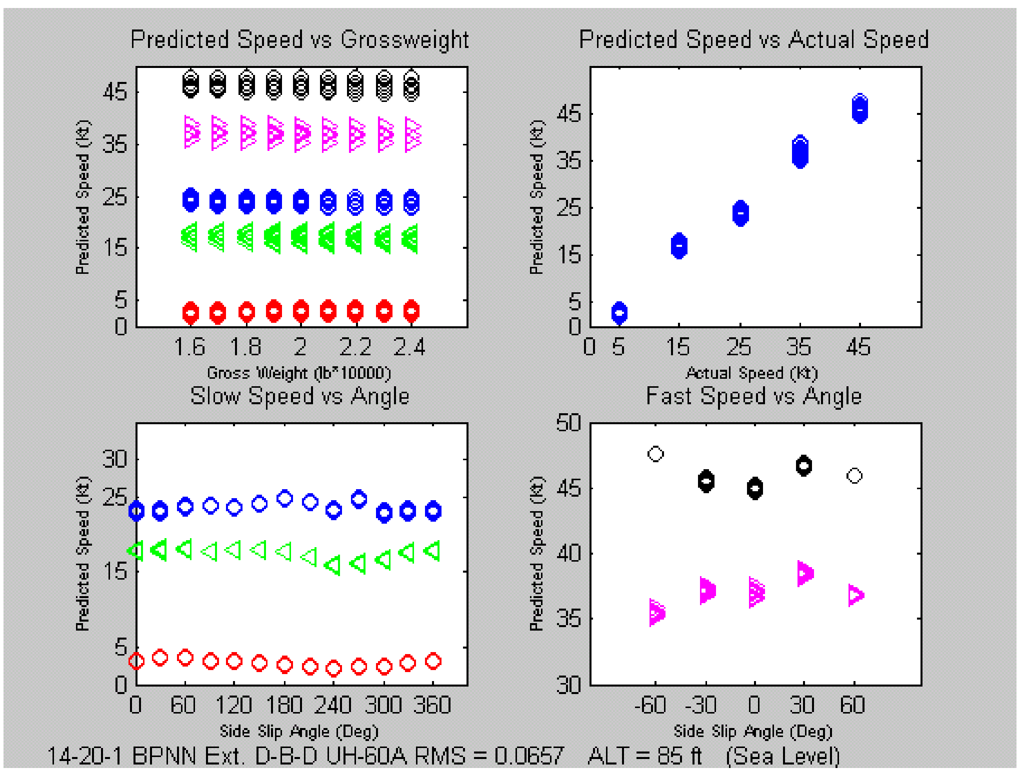


Figure 13. Results for the UH-60A helicopter at 85 ft.; network configuration 14-20-1; Ext. DBD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7002			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.8849	0.4173	8.3461%	2.9556
15	17.4373	0.7170	4.7798%	3.2811
25	23.7263	0.6597	2.6389%	2.3582
35	36.9723	1.0187	2.9105%	3.7970
45	46.1386	0.9508	2.1129%	2.6505

Table 8. Results for the UH-60A helicopter at 85 ft.; network configuration 14-20-1; Ext. DBD learning rule.

2. RBFN Networks

Figure 14 and Table 9 present findings obtained for the 14-200-1 RBFN configuration. Several RBFN network configurations were implemented. The best result was obtained with the 14-200-1 NCD configuration; where the RMS error is 0.0733. However, results also show a larger velocity variance and maximum error than those obtained with BPNN implementations. In addition, the maximum speed error is 9.2719 knots, which is much larger than that observed with BPNN implementations. The error SD is around 1 knot at all speeds, except at 45 knots where it significantly increases to about 4 knots. The overall error SD is 1.634. Therefore, results showed that BPNN implementations are better suited than RBFN configurations for this problem.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.6340			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.1368	1.0915	21.8301%	5.4400
15	14.3318	1.0551	7.0338%	2.7856
25	25.6402	1.1621	4.6485%	2.9813
35	35.9359	1.4807	4.2307%	3.8438
45	44.0553	3.8565	8.5699%	9.2719

Table 9. Results for the UH-60A helicopter at 85 ft.; network configuration 14-200-1; RBFN NCD learning rule.

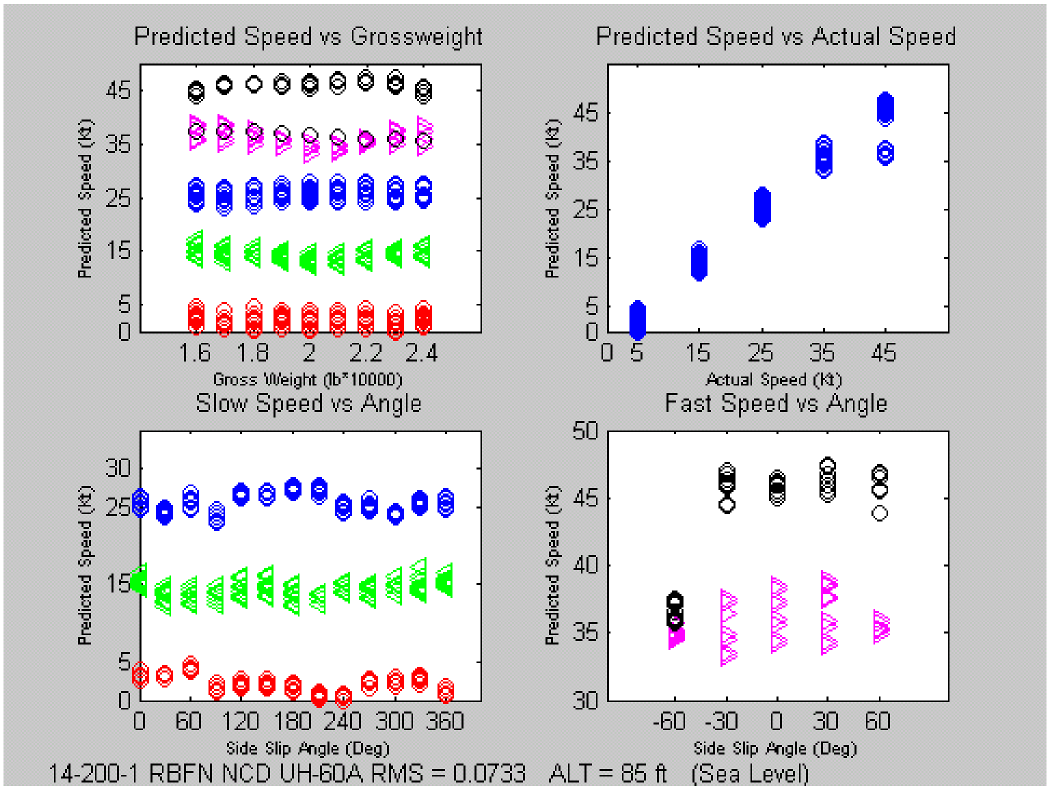


Figure 14. Results for the UH-60A helicopter at 85 ft.; network configuration 14-200-1; RBFN NCD learning rule.

3. Pruned BPNN

Figure 15 and Table 10 present the findings for this configuration. The BPNN was implemented with a pruning capability embedded in the NeuralWare software to increase the prediction efficiency, where pruning is used to remove connections from the network when their contributions are very small [Ref. 6].

In light of the previous analyses, the 14-25-25-1 BPNN configuration with the NCD learning rule was chosen. In this scheme, data were presented to the network as before but the network performance was monitored against the previous ones during the training process after each 1000 iterations. The RMS error was compared at each "checkpoint", and PE contributions checked. At that point the training stopped when the RMS was larger than at the last check, otherwise, it continued for the next 1000 iterations. During this process, PEs with contributions smaller than a given tolerance were disabled, which reduces network complexity and increases generalization efficiency [Ref. 4].

In this analysis, the process went up to 46000 iterations and 7 PEs in the first hidden layer were disabled. Results show that the RMS error is 0.0355 which is best among all architectures investigated. The maximum predicted speed error is ± 3.35 knots, which occurs when the actual speed is 35 knots. Errors for low velocities are below ± 1.8 knots. The error percentage for 5, 15 and 35 knots is about 4 % and about 2.5 % for other speeds. Results show that the predicted speed mean values are very close to the target values when compared with previous networks. Finally, the overall error SD is 0.7167, which is quite good.

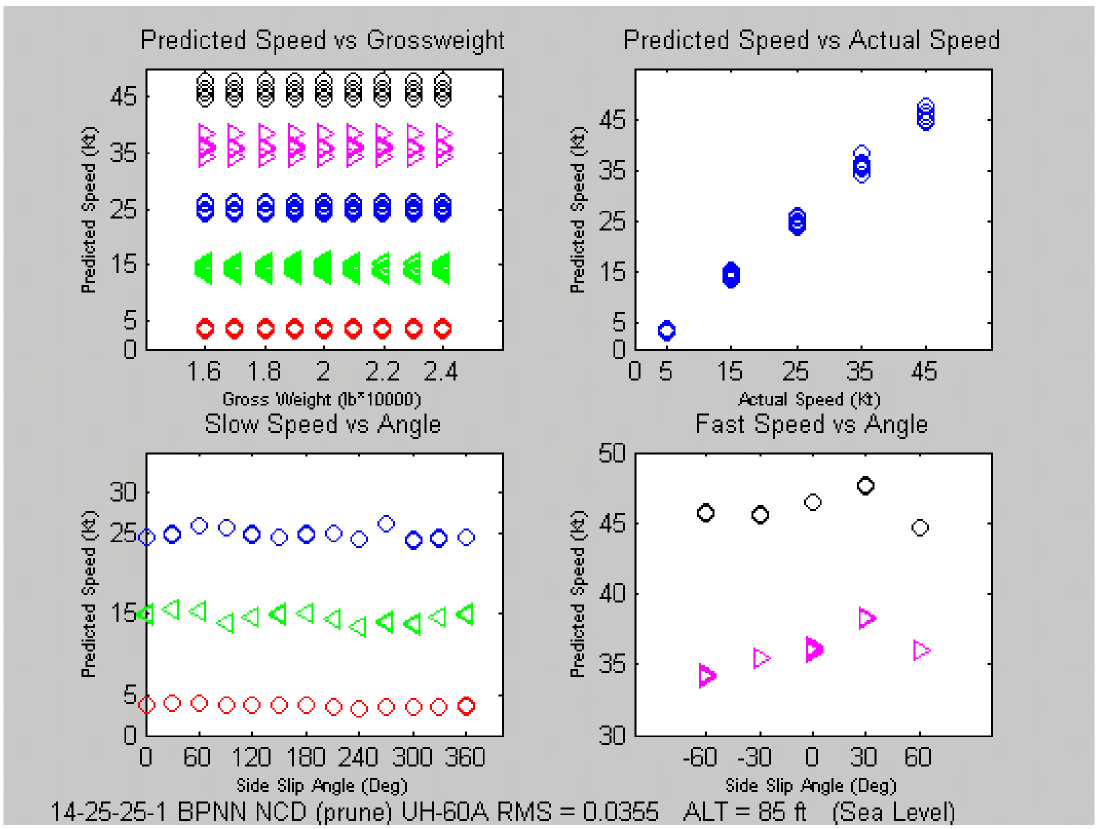


Figure 15. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; NCD (Pruned) learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7167			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.6789	0.2113	4.2253%	1.7089
15	14.5970	0.6322	4.2147%	1.6131
25	24.9281	0.6610	2.6440%	1.2897
35	36.0136	1.3438	3.8395%	3.3496
45	46.0334	1.0039	2.2309%	2.6618

Table 10. Results for the UH-60A helicopter at 85 ft.; network configuration 14-25-25-1; NCD (Pruned) learning rule

The above results showed that the networks with NCD and Ext. DBD learning rules and pruned network have better performance than those obtained with other networks. Therefore, all other analyses, UH-60A in-ground effect, baseline data and OH-6A analyses, were performed using these configurations only.

4. In-Ground Effect Analysis

In-ground effect analysis of the UH-60A helicopter is performed when the pressure altitude is at sea level and the altimeter above ground is at 20 feet. The sideslip angle parameter was added to the NN inputs. Three network architectures were used for the in-ground effect analysis; 15-25-25-1 BPNN with NCD and Ext. DBD learning rules, and 15-18-25-1 BPNN with the NCD learning rule.

a. 15-25-25-1 BPNN NCD

Figure 16 and Table 11 summarize the findings for this configuration. The RMS error is 0.0374 and the maximum error is 3 knots (at 35 knots). The error SD is also the highest at this speed. The network performance at low speeds and at 45 knots is quite good, with the error SD of less than 1.2, but the error SD goes up to 1.8 knots for a 35 knots speed, which corresponds to about 5 %. The overall airspeed error SD is 0.8469 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.8469			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	4.1567	0.2334	4.6686%	1.3234
15	14.0652	0.5405	3.6033%	2.1345
25	25.8750	0.6741	2.6963%	2.5196
35	35.5357	1.8206	5.2017%	3.0036
45	45.2548	1.2123	2.6941%	2.2981

Table 11. Results for the UH-60A helicopter at 20 ft.; network configuration 15-25-25-1; NCD learning rule.

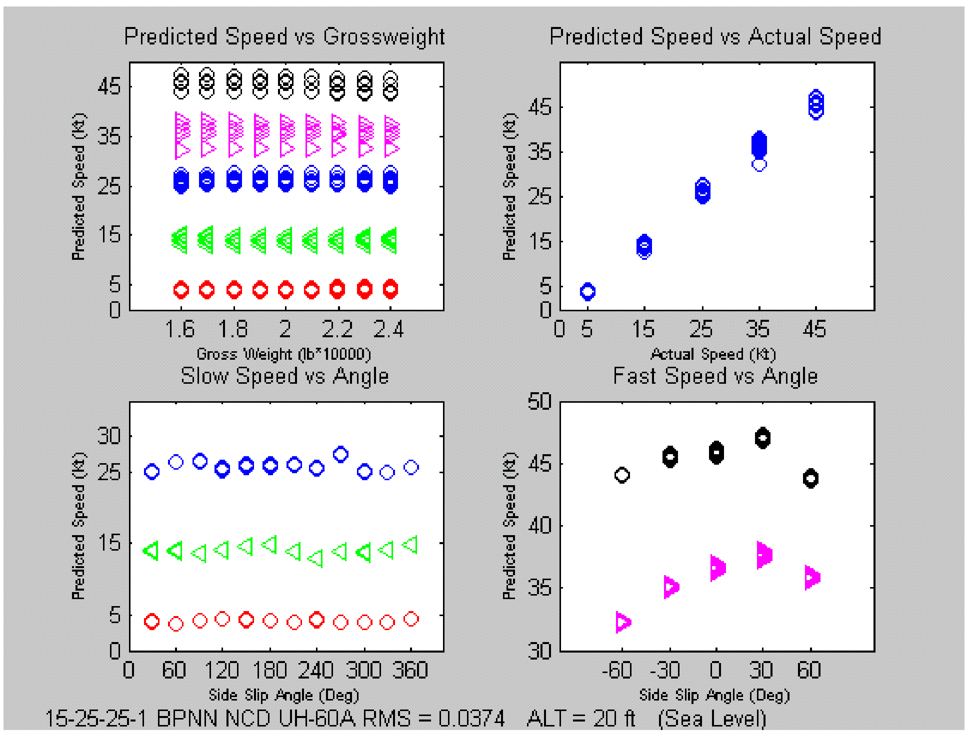


Figure 16. Results for the UH-60A helicopter at 20 ft.; network configuration 15-25-25-1; NCD learning rule.

b. 15-25-25-1 BPNN Ext. DBD

Figure 17 and in Table 12 present the results for this configuration. The RMS error obtained is 0.0637. Note that the maximum error occurs when the helicopter is moving at 45 knots with a -60^0 angle, unlike other networks. Moreover, the maximum error, 4.81 knots, is higher than that obtained in most networks. Apparently, the error SD is also larger at 45 knots. Although the error SD at low speeds is less than 0.88 knots, the error percentage is about 6%. Overall, the error SD is 0.8308 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.8308			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.5383	0.3051	6.1027%	3.1938
15	16.7466	0.8875	5.9166%	3.7167
25	23.4740	0.5188	2.0750%	2.1472
35	35.6272	0.9530	2.7229%	2.1010
45	46.5186	1.6658	3.7018%	4.8112

Table 12. Results for the UH-60A helicopter at 20 ft.; network configuration 15-25-25-1; Ext. DBD learning rule.

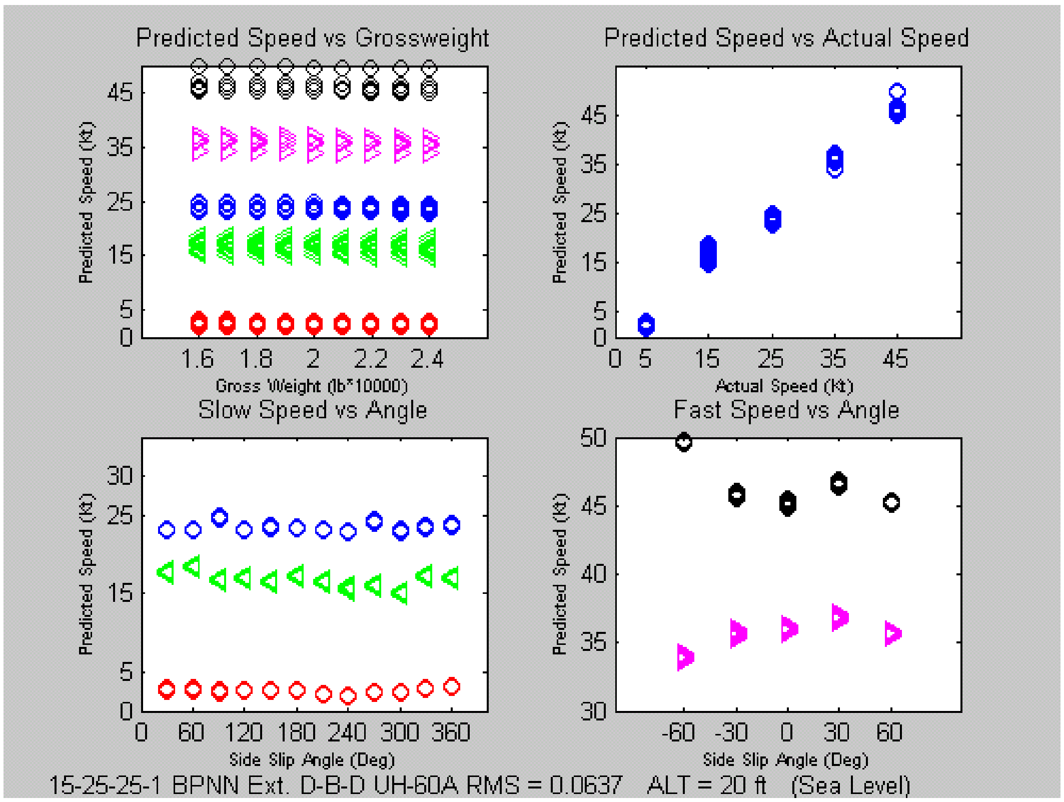


Figure 17. Results for the UH-60A helicopter at 20 ft.; network configuration 15-25-25-1; Ext. DBD learning rule.

c. 15-25-25-1 BPNN NCD (Pruned)

Table 13 and Figure 18 present the results for this configuration. Seven PEs were disabled in the first hidden layer. The RMS error is about 0.0724. The maximum error is 5.2 knots at 35 knots. The error SD is also larger at this speed. The overall error SD is 0.8884 knots which is slightly larger than the NCD 15-25-25-1 network.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.8884			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	4.7761	0.2380	4.7603%	0.7986
15	11.4320	0.4591	3.0609%	4.8662
25	24.8531	1.3065	5.2260%	2.8201
35	37.9958	1.4152	4.0435%	5.2195
45	45.4203	0.7768	1.7263%	1.9679

Table 13. Results for the UH-60A helicopter at 20 ft.; network configuration 15-25-25-1; NCD (Pruned) learning rule.

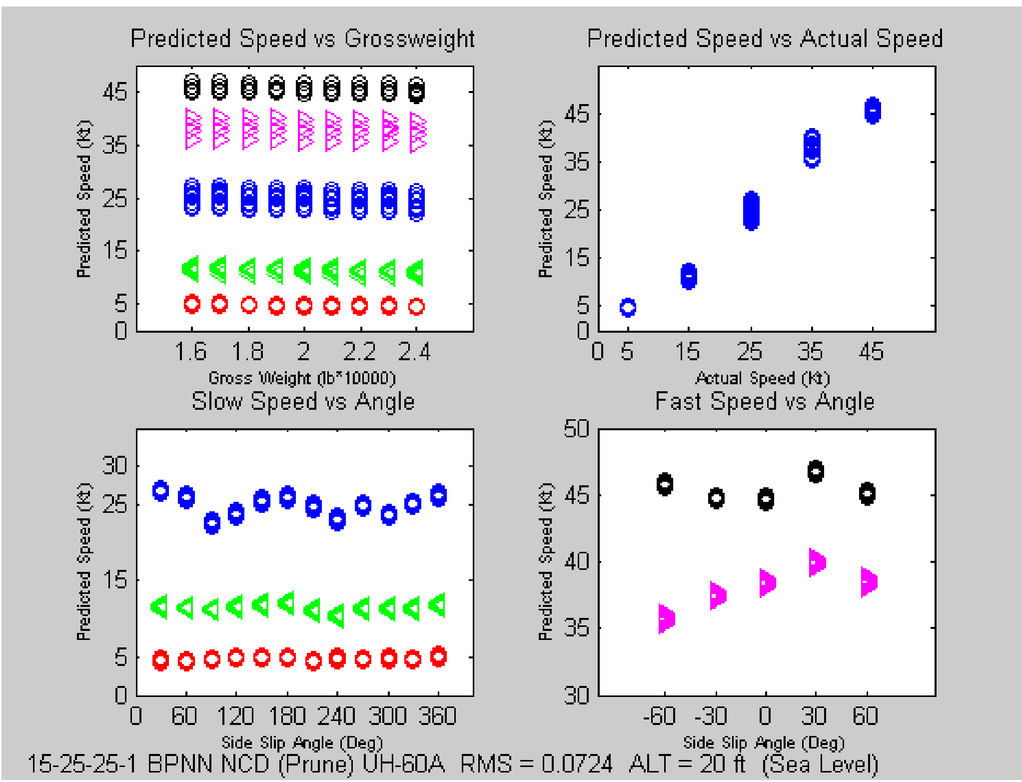


Figure 18. Results for the UH-60A helicopter at 20 ft.; network configuration 15-25-25-1; NCD (Pruned) learning rule.

d. 15-18-25-1 BPNN NCD

Table 14 and Figure 19 present the results for this configuration. This setup is implemented as an alternative to the pruned network to reduce the network training time and complexity. The RMS error is about 0.0384. The maximum error is 2.27 knots at 35 knots. We note the error SD is larger at this speed. The overall error SD is 0.8853 knots, which is slightly higher than the NCD 15-25-25-1 network.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.8853			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.7156	0.2597	5.1943%	1.7481
15	15.0498	0.6162	4.1081%	1.3976
25	24.1842	0.9682	3.8726%	1.7528
35	34.6029	1.5548	4.4424%	2.2788
45	45.5511	1.2541	2.7870%	2.2318

Table 14. Results for the UH-60A helicopter at 20 ft.; network configuration 15-18-25-1; NCD learning rule. Results of UH-60A for 15-18-25-1 NCD.

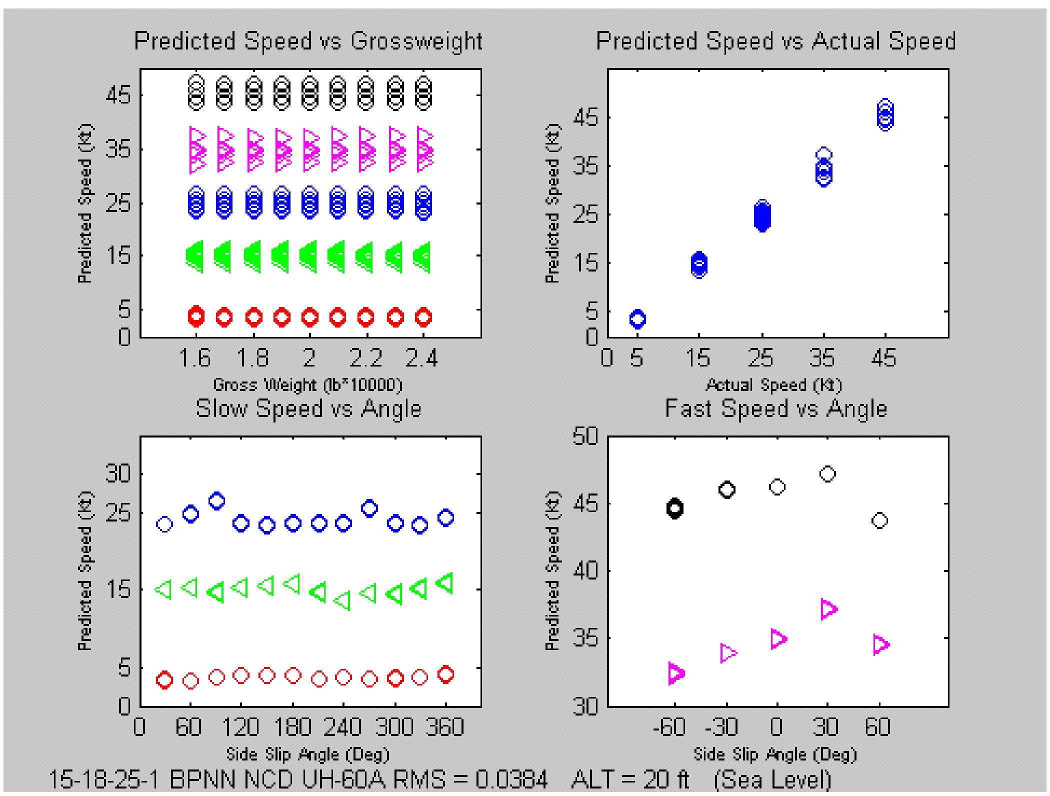


Figure 19. Results for the UH-60A helicopter at 20 ft.; network configuration 15-18-25-1; NCD learning rule.

5. Baseline Data Set Analysis

The baseline data set was formed combining both 20 ft. and 85 ft. data sets. First, the network was trained with this data set, and the network performance was measured using the baseline test set. Then, using single condition test data sets separately, the network performance was evaluated for IGE and OGE conditions. As a result, the network-training time increased, due to the increase in the number of data points in the baseline data set.

a. 15-18-25-1 BPNN NCD

Results for this setup are shown in Table 15 and in Figure 20. They show that the RMS error is 0.0755. The maximum error is about 6.3 knots when the speed is 35 knots. The error SD of 1.4224 knots is higher than that observed with other baseline networks and the single condition data networks. The error percentage for all speeds at 1 σ is about 6 %. The error SD is higher at fast speed, especially when the sideslip angle is –60 and 60 degrees.

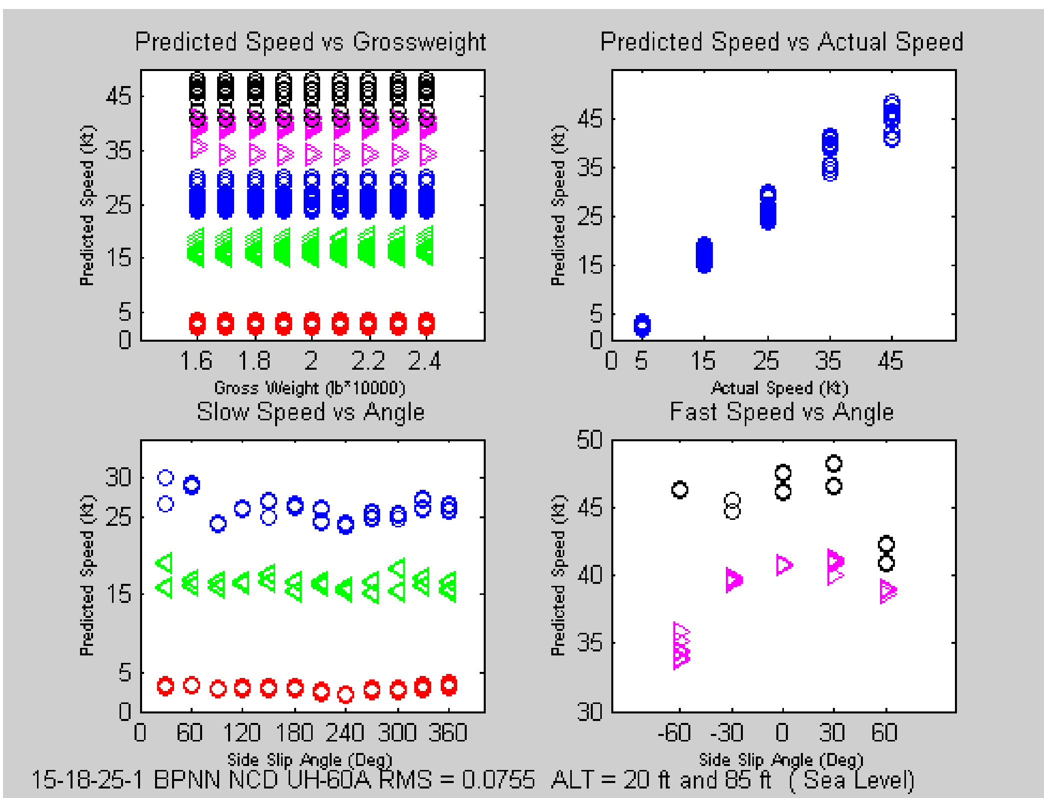


Figure 20. Results for the UH-60A helicopter with baseline data; network configuration 15-18-25-1; NCD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.4224			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.9564	0.3565	7.1292%	2.8415
15	16.4937	0.9016	6.0108%	4.2300
25	26.0262	1.5856	6.3426%	5.1133
35	38.8061	2.3985	6.8529%	6.2953
45	45.4173	2.1685	4.8188%	4.1447

Table 15. Results for the UH-60A helicopter with baseline data; network configuration 15-18-25-1; NCD learning rule.

b. 15-25-25-1 BPNN NCD

Results are depicted in Figure 21 and Table 16. They show that the RMS error is 0.0762, and the maximum error is 6.2424 knots at 45 knots. The airspeed error SD is larger when the speed is 25 knots. The largest error percentage is equal to 9 % (at 5 knots). The overall network error SD is 1.5664 knots.

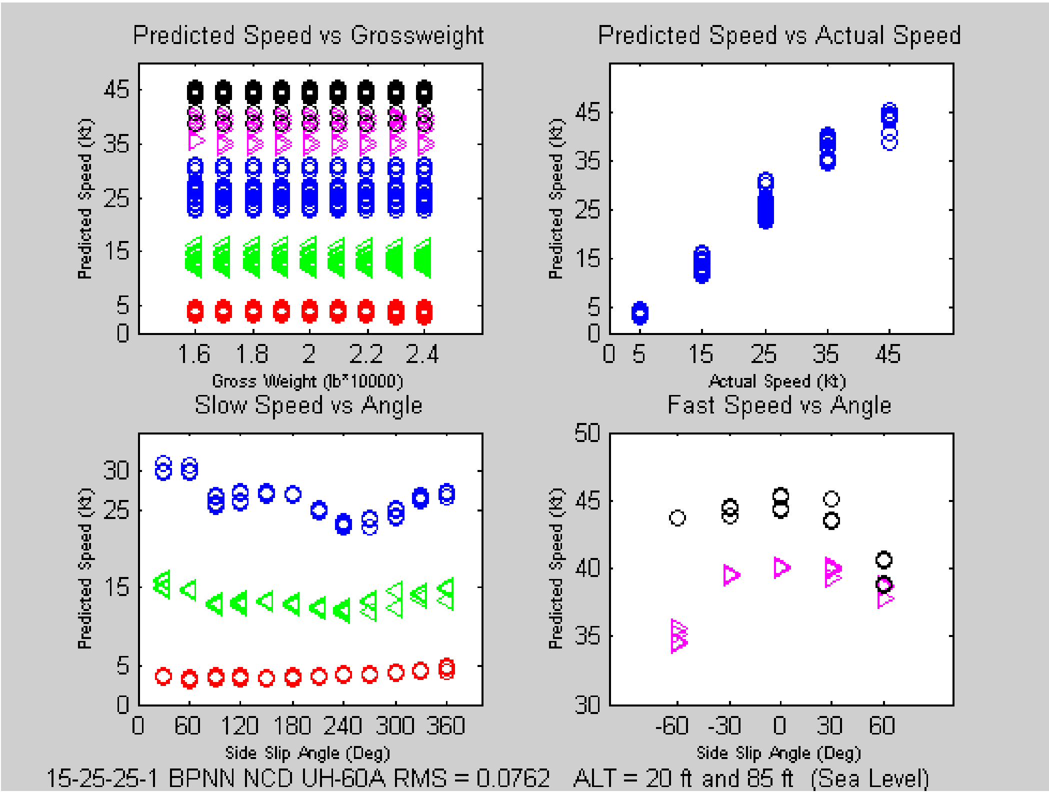


Figure 21. Results for the UH-60A helicopter with baseline data; network configuration 15-25-25-1; NCD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.5664			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.7732	0.4005	8.0108%	1.8981
15	13.4327	1.0494	6.5960%	3.1877
25	26.4841	2.2515	9.0058%	6.1337
35	38.4891	1.9183	5.4810%	5.1834
45	43.3588	1.9560	4.3466%	6.2424

Table 16. Results for the UH-60A helicopter with baseline data; network configuration 15-25-25-1; NCD learning rule.

c. 15-25-25-1 BPNN Ext. DBD

Table 17 and in Figure 22 present the findings for this configuration. Results show that the RMS error is 0.0501, which is better than that of the NCD scheme. The maximum error is about 4.5 knots (at 35 knots). We note the predicted speed mean values are close to the target speeds. The error percentage at the speed of 5 knots is about 10 %. The NN prediction is quite good at the speed of 45 knots. The overall network error SD is 0.7320, which is the best obtained with the baseline data set.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7320			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.2079	0.5382	10.7647%	2.8105
15	16.2084	0.6293	4.1953%	2.5304
25	25.0000	0.7696	3.0782%	2.3555
35	37.3300	1.0832	3.0949%	4.5313
45	45.9893	0.8478	1.8839%	1.9112

Table 17. Results for the UH-60A helicopter with baseline data; network configuration 15-25-25-1; Ext. DBD learning rule.

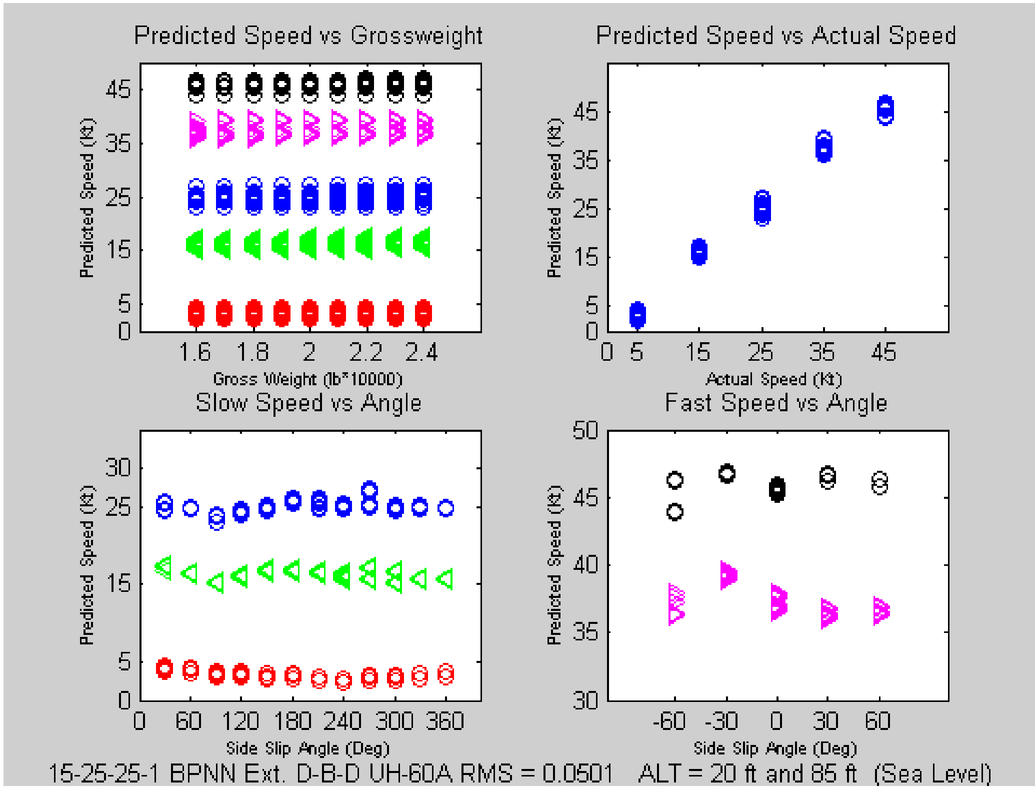


Figure 22. Results for the UH-60A helicopter with baseline data; network configuration 15-25-25-1; Ext. DBD learning rule.

d. 15-25-25-1 BPNN NCD (Pruned)

Table 18 and Figure 23 present the findings for this configuration. Results show that one PE was pruned in the first hidden layer. The resulting RMS error is 0.1213, and the maximum error at 45 knots is 8.6414 knots. The largest error SD and error percentage (11.5 %) are obtained for a 25 knots speed. The overall network error SD is 2.0830 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 2.0830			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.9069	0.3420	6.8404%	2.8877
15	9.8217	1.1116	7.4108%	7.2736
25	23.5765	2.8864	11.5454%	6.5795
35	38.3384	2.8346	6.6262%	7.1468
45	43.8399	2.9818	1.8839%	8.6414

Table 18. Results for the UH-60A helicopter with baseline data; network configuration 15-25-25-1; NCD (Pruned) learning rule.

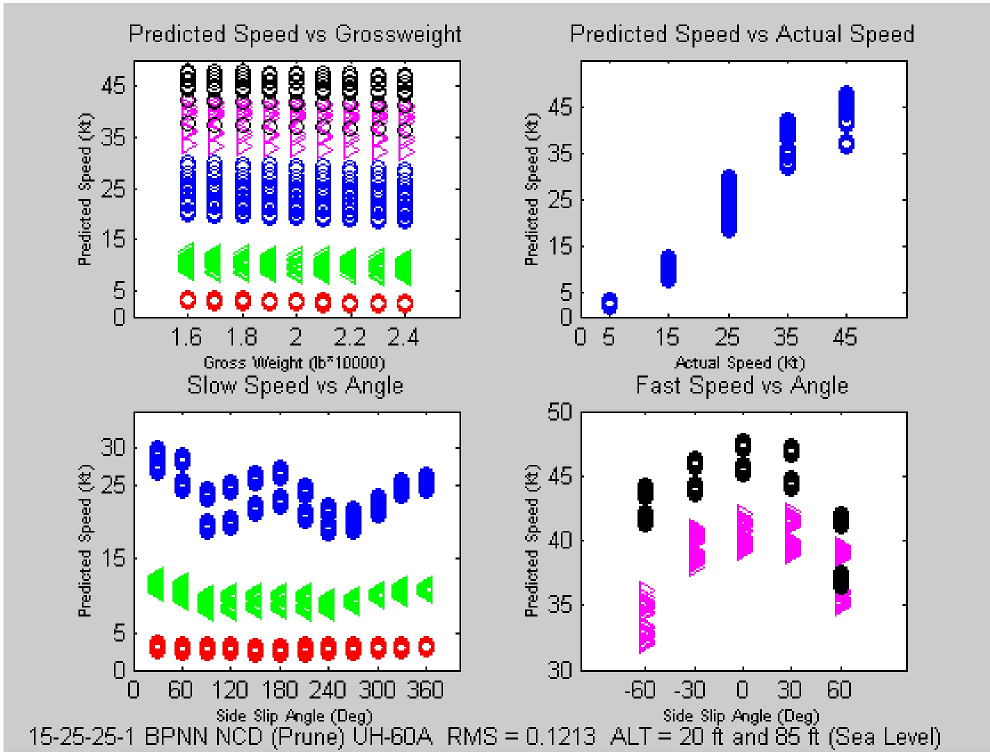


Figure 23. Results for the UH-60A helicopter with baseline data; network configuration 15-25-25-1; NCD (Pruned) learning rule.

e. Baseline Data Set IGE Analysis

(1) 15-25-25-1 BPNN NCD. Results are shown in Table 19 and Figure 24. They show the RMS error is 0.0759. The maximum error is 6.2 knots, which is almost the same for speeds 25 knots and higher. The network performance at high speeds is better than that of at low speeds since the percent error at 35 and 45 knots is about 5 %, whereas at lower speeds it is twice that number. The overall error SD is 1.5516 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.5516			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.7699	0.5063	10.1268%	1.8981
15	13.7003	1.1242	7.4947%	2.7778
25	26.5671	2.1650	8.6599%	6.1337
35	38.3892	1.7970	5.1342%	5.1715
45	42.8593	2.0669	4.5931%	6.2424

Table 19. Results for the UH-60A helicopter at 20 ft with baseline data; network configuration 15-25-25-1; NCD learning rule.

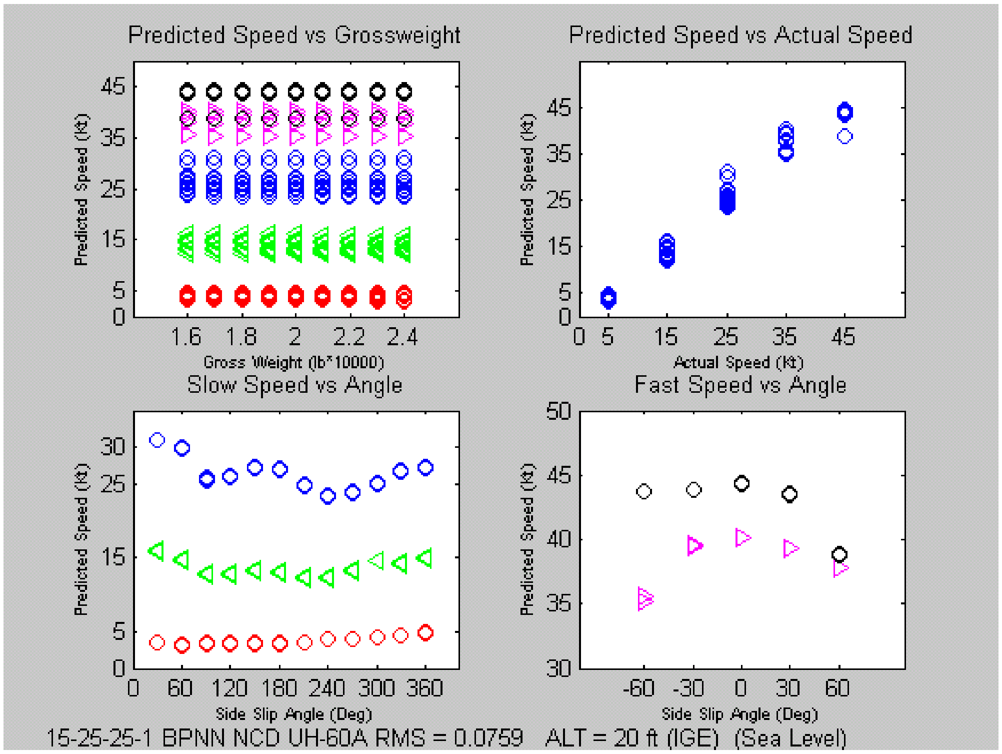


Figure 24. Results for the UH-60A helicopter at 20 ft with baseline data; network configuration 15-25-25-1; NCD learning rule.

(2) 15-25-25-1 BPNN Ext. DBD. Results are shown in Table 20 and Figure 25. They show the RMS error is 0.0499. The maximum error of 4.5 knots is obtained at a 35 knots speed. Note that the network performance is quite good when compared with the NCD scheme. The overall error SD is 0.6505 knots, which is significantly less than it is in the NCD scheme.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.6505			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.5127	0.4383	8.7656%	2.2475
15	16.4063	0.6193	4.1287%	2.5304
25	25.1852	0.7545	3.0179%	2.3505
35	37.6802	0.9720	2.7772%	4.5313
45	46.0822	0.4932	1.0961%	1.8867

Table 20. Results for the UH-60A helicopter at 20 ft with baseline data; network configuration 15-25-25-1; Ext. DBD learning rule.

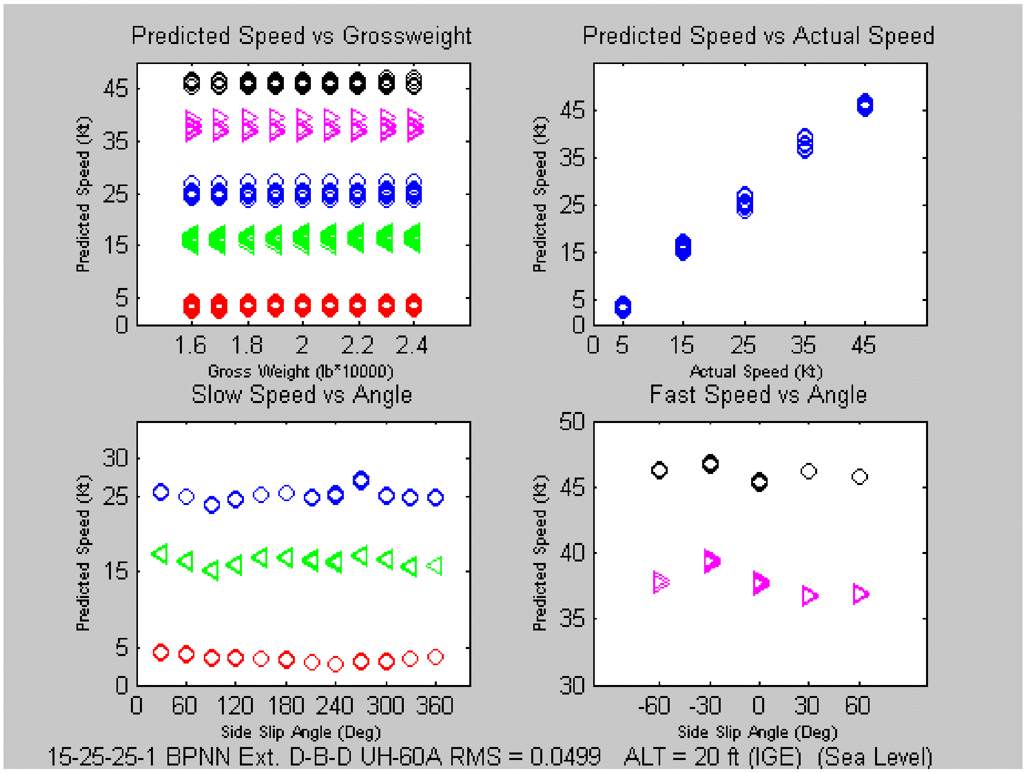


Figure 25. Results for the UH-60A helicopter at 20 ft with baseline data; network configuration 15-25-25-1; Ext. DBD learning rule.

(3) 15-25-25-1 BPNN NCD (Pruned). Results are shown in Table 21 and Figure 26. They show that the RMS error is 0.0901. The maximum speed error of 7.9 knots is obtained for a 45 knots speed. We note that the network performance is not good when the pruning facility is enabled. The overall error SD is 1.7173 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.7173			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	4.3654	0.2933	5.8653%	1.1256
15	11.5338	1.1787	7.8578%	4.9166
25	23.9066	2.2817	9.1268%	4.0707
35	37.3058	2.1850	6.2429%	4.4207
45	42.1901	2.5806	5.7346%	7.9128

Table 21. Results for the UH-60A helicopter at 20 ft with baseline data; network configuration 15-25-25-1; NCD (Pruned) learning rule.

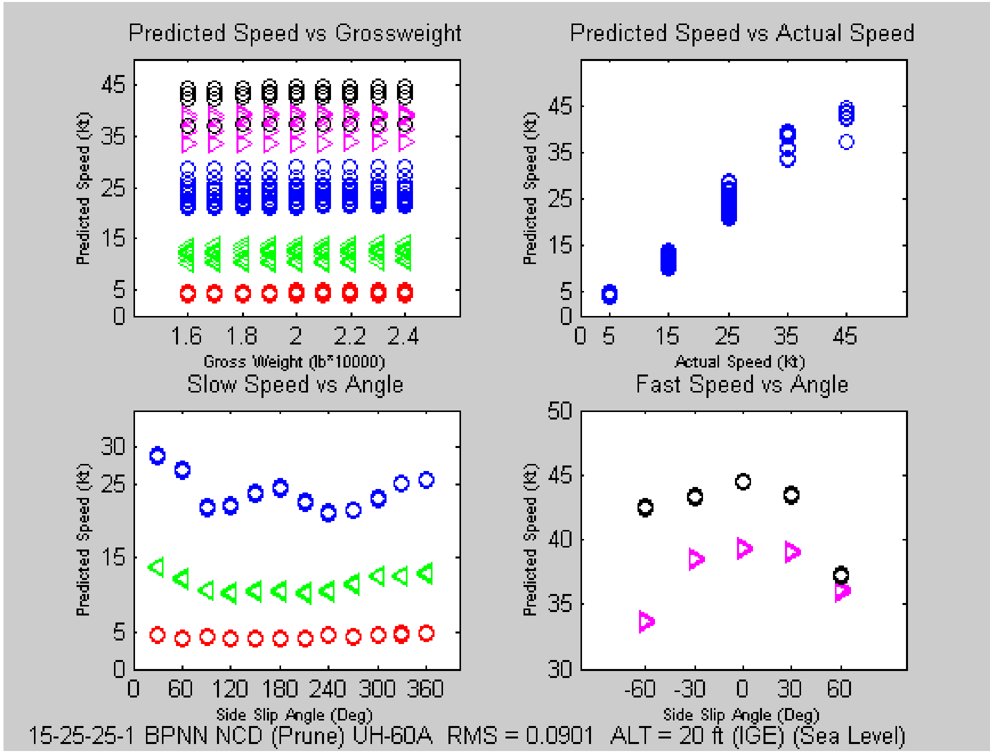


Figure 26. Results for the UH-60A helicopter at 20 ft with baseline data; network configuration 15-25-25-1; NCD (Pruned) learning rule.

f. Baseline Data Set OGE Analysis

(1) 15-25-25-1 BPNN NCD. Findings for this configuration are presented in Table 22 and Figure 27. Results show that the RMS error is 0.0766. The maximum error is 5.86 knots. The network performance at a speed of 25 knots is not good, as the error SD is 2.3 knots at that speed. The error percentage is about 5 % for speeds other than 25 knots but it increases up to 9 % for 25 knots. The overall error SD is 1.5516 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.5516			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.7779	0.2569	5.1390%	1.6144
15	13.1631	0.8966	5.9771%	3.1890
25	26.4013	2.3416	9.3664%	5.8619
35	38.5880	2.0476	5.8502%	5.1844
45	43.8585	1.7192	3.8204%	4.3961

Table 22. Results for the UH-60A helicopter at 85 ft with baseline data; network configuration 15-25-25-1; NCD learning rule.

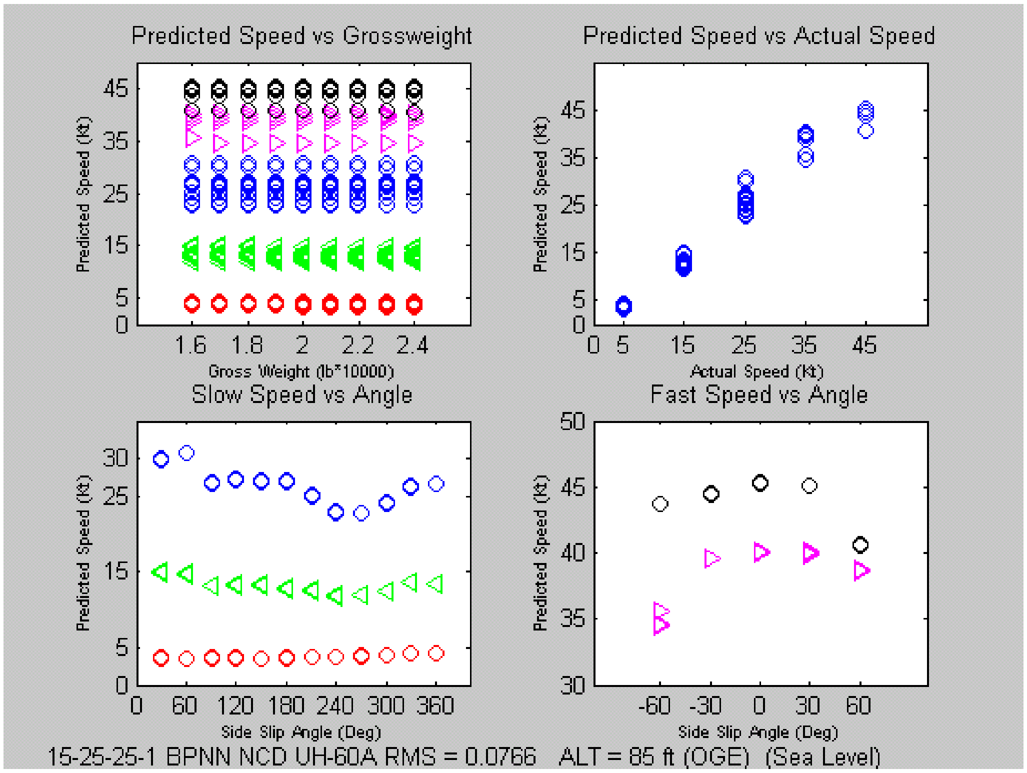


Figure 27. Results for the UH-60A helicopter at 85 ft with baseline data; network configuration 15-25-25-1; NCD learning rule.

(2) 15-25-25-1 BPNN Ext. DBD. Table 23 and in Figure 28 present the results for this configuration. Results show that the RMS error is 0.0482. The maximum error observed at 35 knots is equal to 3.8 knots. Note that this error is smaller than that observed in the NCD scheme, and the network performance is quite good when compared with the NCD scheme for the OGE baseline setup. At low speeds, the error SD is about 0.5 knots. The network performance is quite good for all speeds except for 5 knots where the error percentage increases to 9 %. The overall error SD is 0.6850 knots.

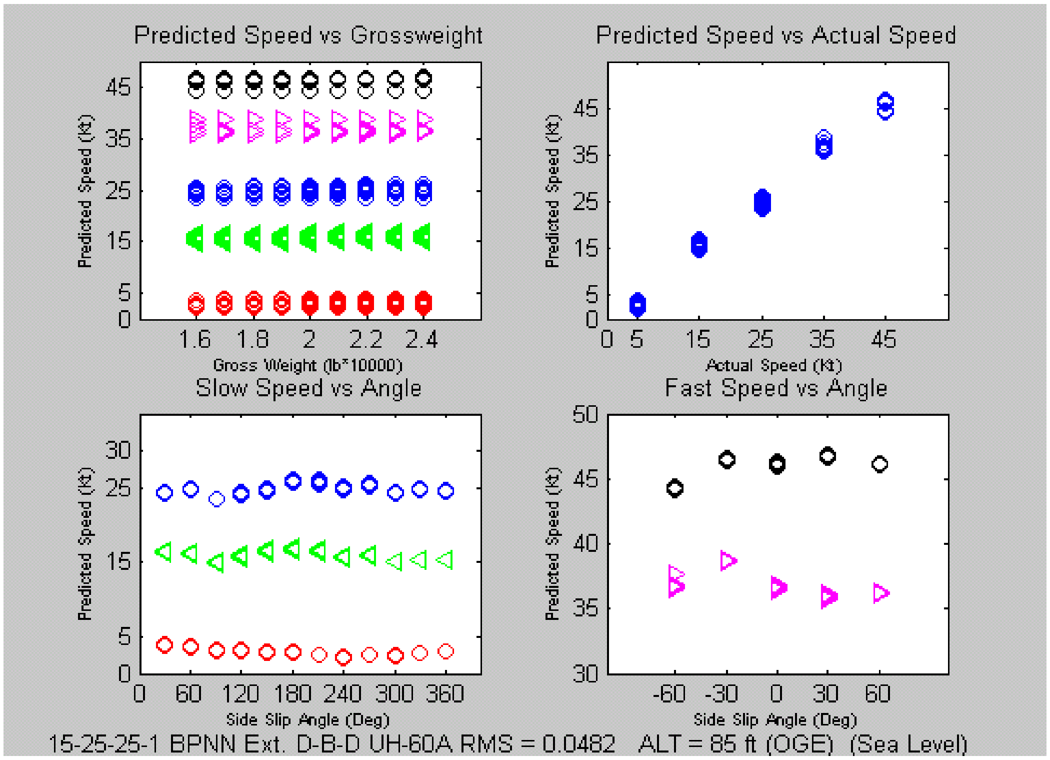


Figure 28. Results for the UH-60A helicopter at 85 ft with baseline data; network configuration 15-25-25-1; Ext. DBD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.6850			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.9227	0.4746	9.5519%	2.8377
15	15.8606	0.5962	3.9746%	1.9213
25	24.8085	0.6801	2.7205%	1.5601
35	36.8469	1.0210	2.9171%	3.8006
45	45.9612	0.9043	2.0096%	1.9007

Table 23. Results for the UH-60A helicopter at 85 ft with baseline data; network configuration 15-25-25-1; NCD learning rule.

(3) 15-25-25-1 BPNN NCD (Pruned). Table 24 and in Figure 29 present the results for this configuration. Results show that the RMS error is 0.1087. The maximum error observed for a 35 knots speed is equal to 7.14 knots. Note that the network performance degrades when the pruning facility is enabled. The overall error SD is 1.8642 knots.

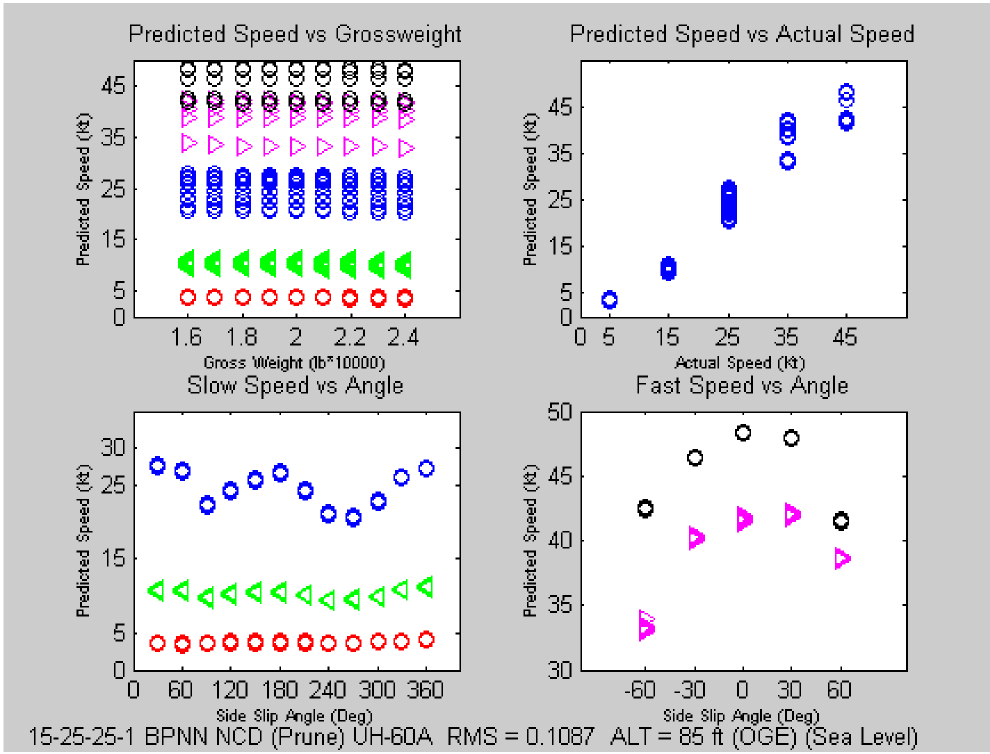


Figure 29. Results for the UH-60A helicopter at 85 ft with baseline data; network configuration 15-25-25-1; NCD (Pruned) learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.8642			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.7243	0.1567	3.1338%	1.6318
15	10.3127	0.5639	3.7592%	5.7803
25	24.6236	2.3566	9.4263%	4.6493
35	39.1423	3.2081	9.1660%	7.1416
45	45.3547	2.8266	6.2814%	3.5760

Table 24. Results for the UH-60A helicopter at 85 ft with baseline data; network configuration 15-25-25-1; NCD (Pruned) learning rule.

6. Simplifying The Data Set Using Eigenvalues and Eigenvectors

Preprocessing the data to obtain a simplified NN is often useful. Goff developed a NN helicopter airspeed prediction study similar to the Haas and McCool study [Ref. 20]. Goff analyzed the each input contribution to the network performance and determined that 11 of the 16 inputs played an important role. Results obtained by preprocessing the data and simplifying the network inputs were quite successful. His study on the Lynx MK9 showed that airspeed in the low speed environment could be predicted with an error of +/-3.1 knots at 1 σ [Ref. 20].

Another way to decrease the NN input dimension and simplify the network structure by projecting the high-dimensional data onto a lower dimensional input space using principal component analysis (PCA). PCA is applied by considering the input data set as a matrix, A, and translating this matrix to a diagonal or upper triangular form to compute its eigenvalues and eigenvectors. Eigenvectors are the normal modes of the system and they act independently. The beauty of eigenvalues and eigenvectors is that they capture the characteristics and behavior of the whole system. The key equation for eigenvalues and eigenvectors is:

$$Ax = \lambda x , \tag{4.1}$$

where λ is an eigenvalue of the matrix A and the vector x is the associated eigenvector. Most vectors x will not satisfy this equation. A typical x changes direction when multiplied by A, so that Ax is not a multiple of x. Thus, only certain special numbers are eigenvalues and only certain special vectors are eigenvectors [Ref. 18].

After obtaining the eigenvalues and eigenvectors of the single condition data set, dominating eigenvalues were selected. It was observed that six of the eigenvalues were the dominating eigenvalues, leading to a input data sub-matrix with a dimension of [14x6]. This sub-matrix was multiplied by the whole data set again. The purpose of this procedure was to reduce the number of inputs to 6 while still capturing the properties from all inputs. Several network architectures were implemented using this simplified data set. Network configurations with the best performances are described in the following sections. Note that none of these performed as well as the best network trained with the full data sets.

a. 6-18-18-1 BPNN Ext. DBD

Results of this setup are shown in Figure 30 and in Table 25. They show that the RMS error is 0.0565. Although the maximum error of 5.4 knots is at 45 knots, the 11% error percentage is higher for a 5 knots speed. The overall error SD is 1.1692 knots.

The network performance at low speeds is not as satisfactory as it is at fast speeds since the percent error is quite large at low speeds.

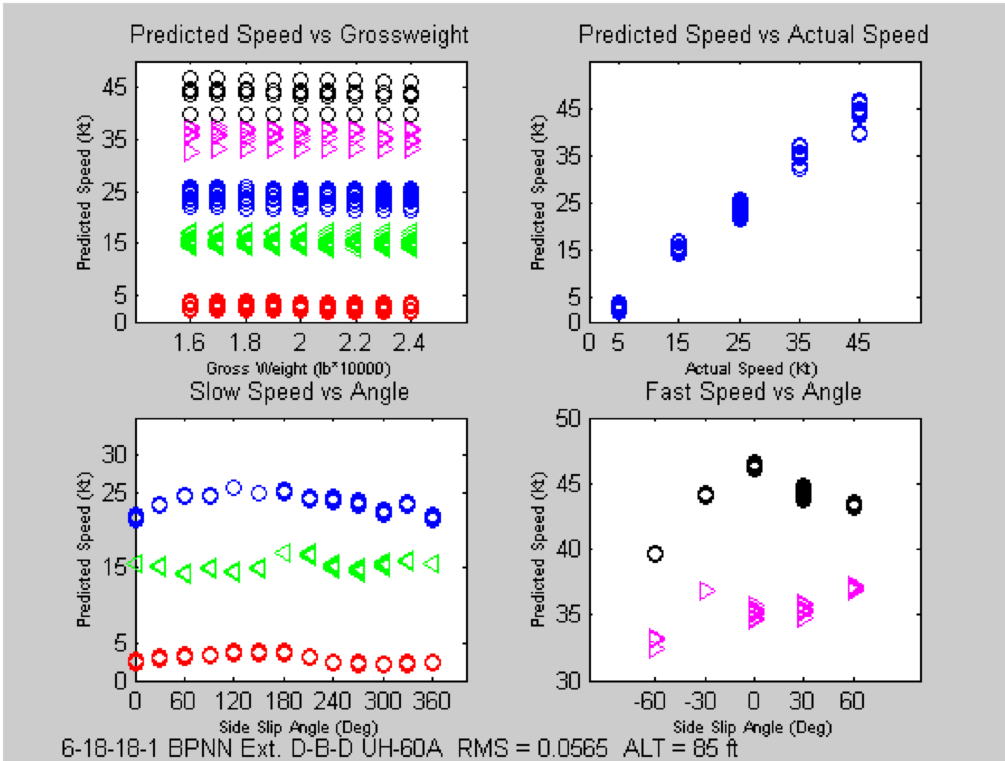


Figure 30. Results for the UH-60A helicopter at 85 ft with simplified data; network configuration 6-18-18-1; Ext. DBD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.1692			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.9375	0.5879	11.7571%	3.0269
15	15.3870	0.7861	5.2444%	2.2.1139
25	23.8447	1.2453	4.9813%	3.5517
35	35.4789	1.4587	4.1677%	2.5422
45	43.5555	2.2160	4.9244%	5.3797

Table 25. Results for the UH-60A helicopter at 85 ft with simplified data; network configuration 6-18-18-1; Ext. DBD learning rule.

b. 6-18-18-1 BPNN NCD (Pruned)

Results are depicted in Figure 31 and Table 26. One PE in the first layer was disabled by the NN. Results show that the RMS error is 0.0648. The maximum error is over 5.0 knots at speeds of 25 and 35 knots. The error SD is also larger at those speeds. We note that, the maximum error percentage is 7.95 % at a speed of 5 knots, and the overall error SD is 1.1942 knots. In conclusion, results showed that the network performance is not as good as that obtained when using the NCD learning rule without the pruning facility.

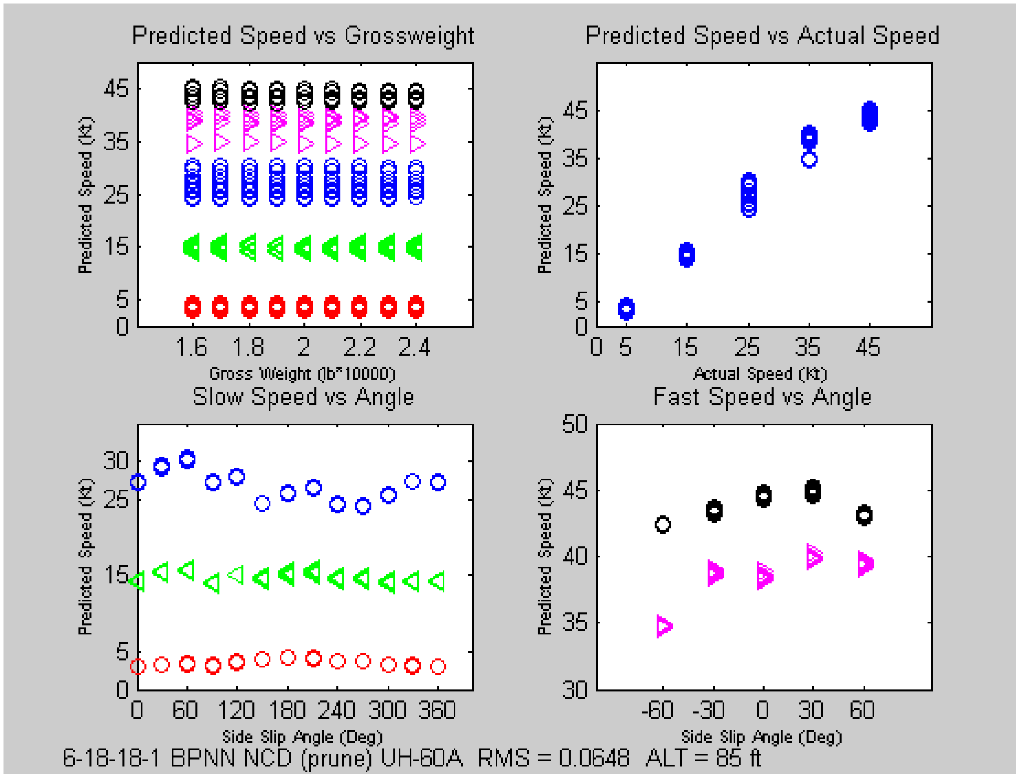


Figure 31. Results for the UH-60A helicopter at 85 ft with simplified data; network configuration 6-18-18-1; NCD (Pruned) learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.1942			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.5084	0.3977	7.9568%	1.9809
15	14.7297	0.5445	3.6297%	1.1050
25	26.7339	1.8057	7.2229%	5.4495
35	38.2674	1.8897	5.3992%	5.3390
45	43.6582	0.9459	2.1020%	2.6491

Table 26. Results for the UH-60A helicopter at 85 ft with simplified data; network configuration 6-18-18-1; NCD (Pruned) learning rule.

c. 6-18-18-1 BPNN NCD

Figure 32 and Table 27 present the results for this configuration. Results show that the RMS error equal to 0.06561 is the best among all the configurations investigated with the simplified data. The maximum error is about 4 knots when the helicopter is moving at a speed of 35 knots with a 30^0 sideslip angle to the left. The error SD is above 1.5 knots at that speed. The error percentage for each speed is about 5 %, except for a 45 knots speed, and the overall error SD is equal to 1.0032 knots.

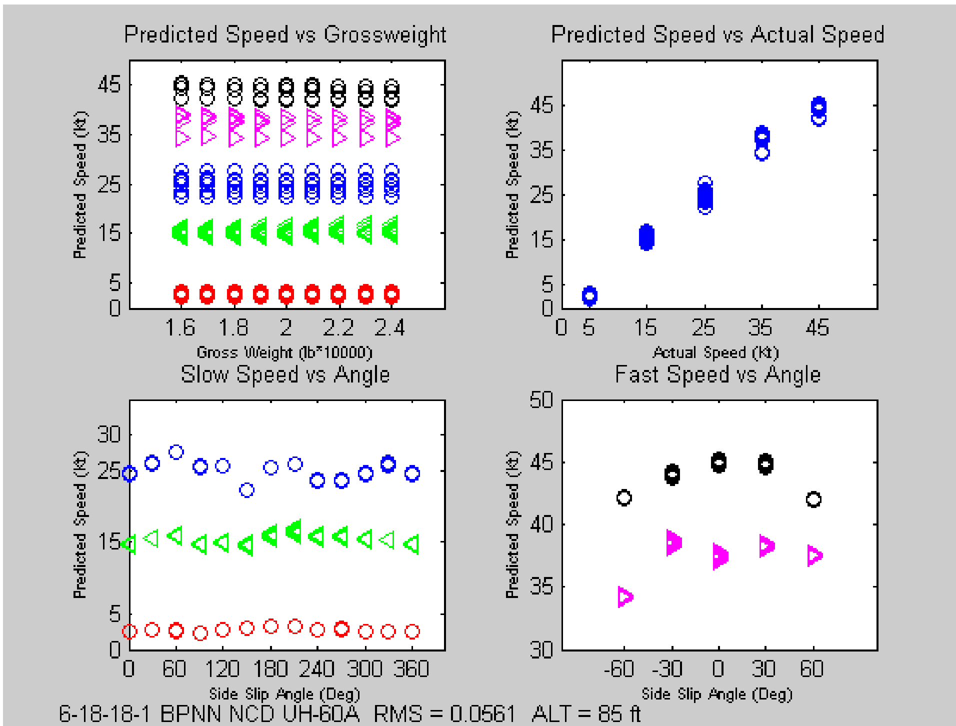


Figure 32. Results for the UH-60A helicopter at 85 ft with simplified data; network configuration 6-18-18-1; NCD learning rule

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.0032			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.7558	0.2888	5.5765%	2.7471
15	15.3714	0.6022	4.0149%	1.9253
25	25.0220	1.3322	5.3287%	2.7581
35	37.1657	1.5913	4.5467%	3.8912
45	43.5339	1.2950	2.8777%	3.1240

Table 27. Results for the UH-60A helicopter at 85 ft with simplified data; network configuration 6-18-18-1; NCD learning rule.

Results showed that the network with simplified data performs worse than the network with the pruning facility enabled. Therefore, the simplified data was not used for OH-6A analyses.

B. ANALYSIS OF THE OH-6A MODEL

1. Out of Ground Effect Analysis at Sea Level

This section presents results of the OGE analysis for the OH-6A helicopter with the simulator altimeter set at 100 feet AGL. The network was trained using the single condition data set obtained for this altitude similarly to the UH-60A analysis. Results for the OGE analysis using the combined data set are shown in the OH-6A Baseline Analysis Section. Note that the engine torque input parameter for the NN model was removed due to the limitation of the engine model parameters for the OH-6A helicopter in the FLIGHTLAB simulator. Therefore, a 14-25-25-1 setup was used for all OH-6A model analyses. Finally, in light of the results obtained from the UH-60A helicopter analysis, only the NCD and Ext. DBD learning rules were investigated.

a. 14-25-25-1 BPNN NCD

Results for this setup are given in Figure 33 and Table 28. They show that the RMS error is 0.0609. The 5 knots maximum speed error is observed at 35 knots, while the absolute maximum error is less than 3 knots at other speeds. The airspeed error SD is less than 1 knot except for 45 knots. The overall error SD is 0.7921 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7921			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	4.2398	0.4797	9.5947%	1.5363
15	12.7387	0.7130	4.7531%	3.3062
25	25.6227	0.8655	3.4618%	2.0925
35	38.6028	0.8703	2.4865%	5.2048
45	44.8812	1.2253	2.7229%	1.9604

Table 28. Results for the OH-6A helicopter at 100 ft (SL); network configuration 14-25-25-1; NCD learning rule.

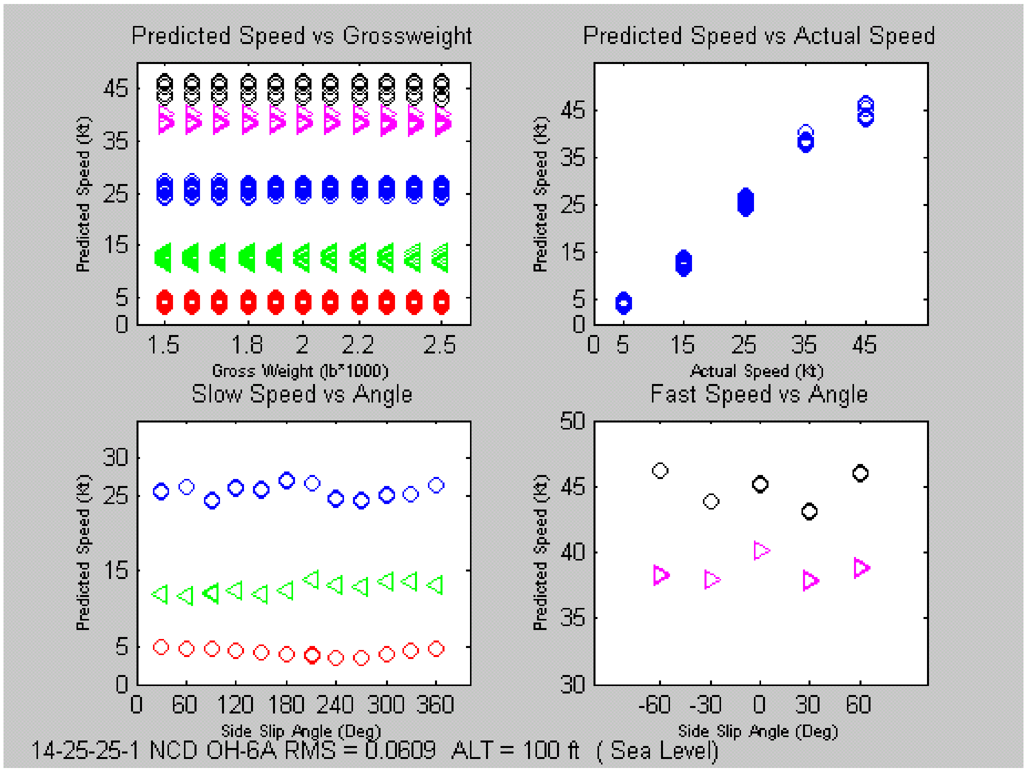


Figure 33. Results for the OH-6A helicopter at 100 ft (SL); network configuration 14-25-25-1; NCD learning rule.

b. 14-25-25-1 BPNN Ext. DBD

Results are shown in Figure 34 and Table 29. They show that the RMS error is 0.0735. Note that the absolute maximum error is about 5 knots at the speed of 35 knots, but the overall error SD of 1.3207 knots is larger than that observed for the NCD setup. For all speeds, except for 45 knots, the error percentage is 5 % and over.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.3207			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.6946	0.4724	9.4478%	2.3008
15	13.3800	1.2846	8.5642%	3.8498
25	25.3109	1.2145	4.8578%	3.2181
35	36.3450	2.6079	7.7543%	5.0472
45	43.8989	1.1780	2.6179%	3.2799

Table 29. Results for the OH-6A helicopter at 100 ft (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

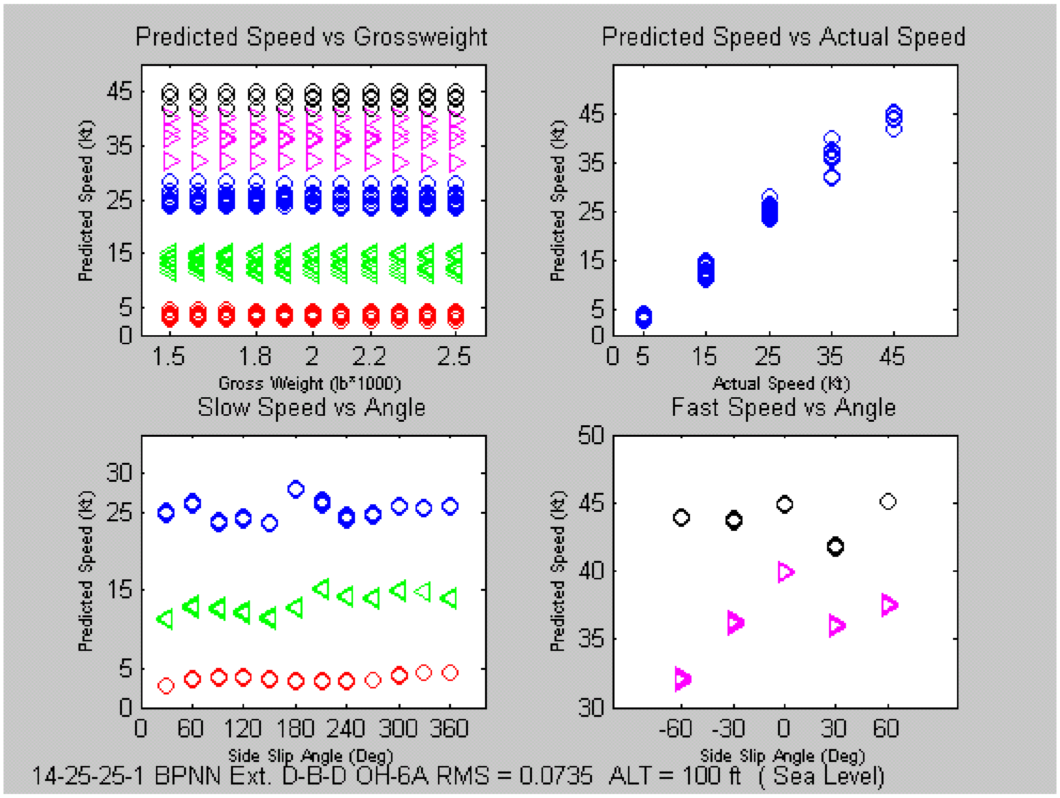


Figure 34. Results for the OH-6A helicopter at 100 ft (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

c. *14-25-25-1 BPNN NCD (Pruned)*

Figure 35 and Table 30 present the findings for this configuration. Results show that the RMS error is 0.0669. The maximum speed error is about 4.6 knots at a speed of 35 knots. The error SD is less than 0.88 knots except for a 45 knots speed. The network performance is quite good except for 5 knots as the error percentage is equal to 9 %, which is very large at that speed. The overall error SD is 0.759 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7590			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.9667	0.4976	9.9521%	1.9224
15	12.0952	0.6239	4.1595%	4.0568
25	25.1773	0.8877	3.5509%	1.7791
35	38.4750	0.6479	1.8510%	4.6989
45	44.9067	1.2291	2.7914%	1.9686

Table 30. Results for the OH-6A helicopter at 100 ft (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

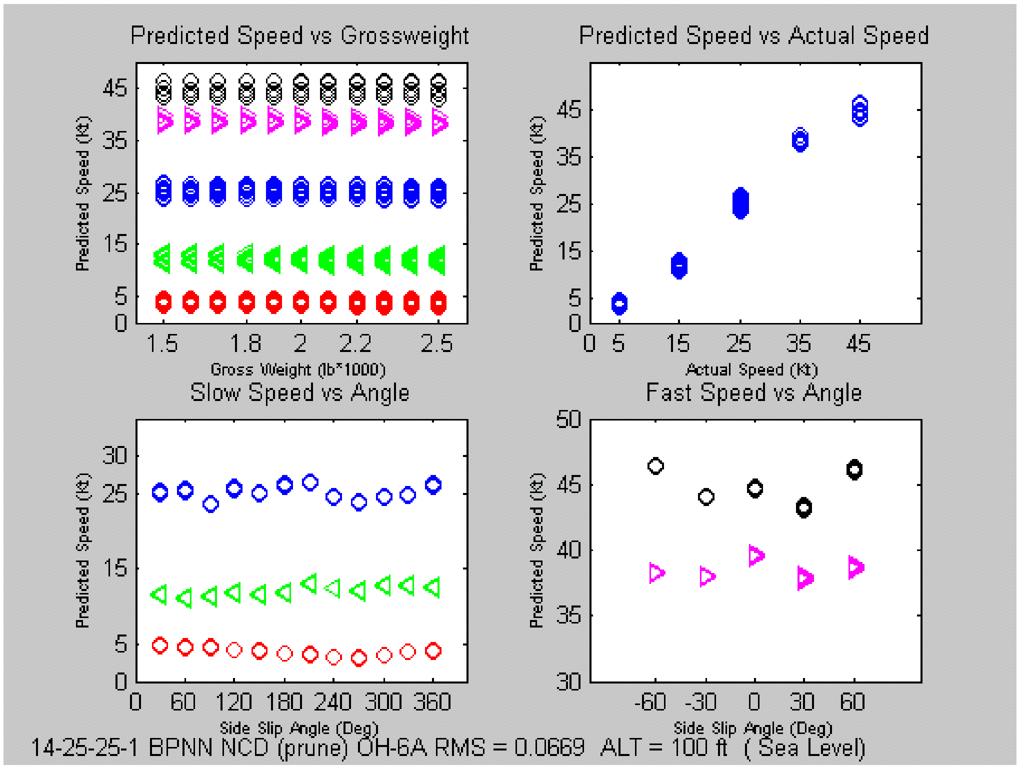


Figure 35. Results for the OH-6A helicopter at 100 ft (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

2. In-Ground Effect Analysis at Sea Level

The OH-6A helicopter in-ground effect analysis was performed at 12 ft AGL. Same network architectures as those considered with out-of-ground analyses were considered. Results are presented below.

a. 14-25-25-1 BPNN NCD

Results are given in Table 31 and Figure 36. They show that the RMS error for this architecture is 0.0592. The maximum error is 5 knots for a speed equal to 35 knots. The percentage of error at 1 σ is about 3% for all speeds except for 5 knots. The speed error SD gets larger at speeds above 25 knots. The overall error SD is 0.8465 knots.

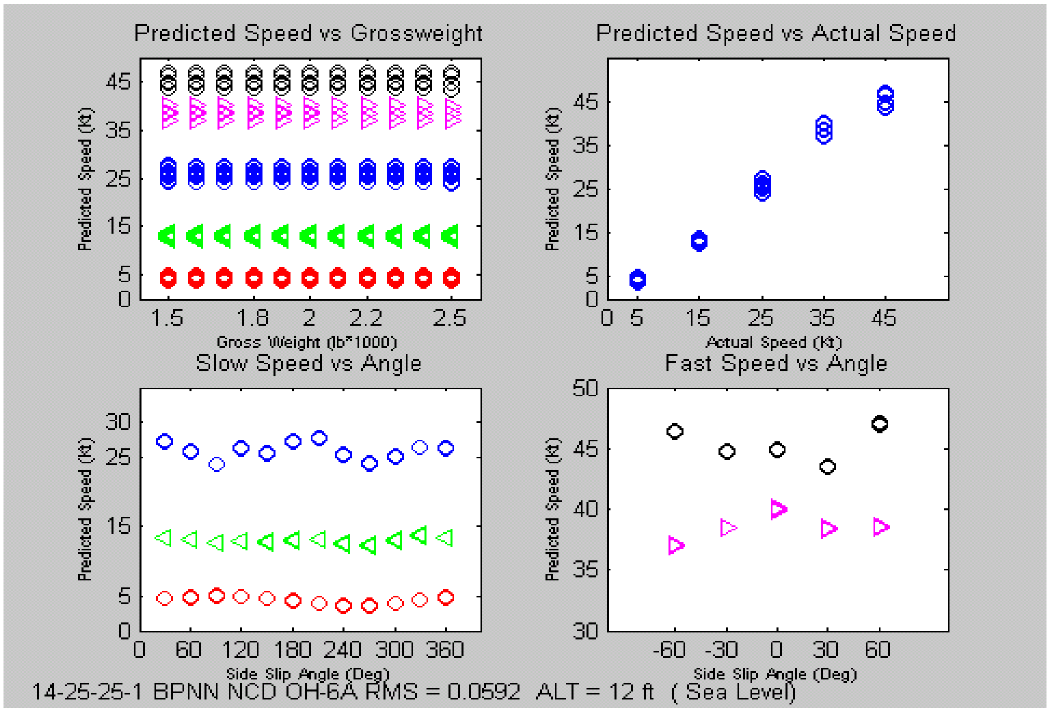


Figure 36. Results for the OH-6A helicopter at 12 ft (SL); network configuration 14-25-25-1; NCD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.8465			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	4.4013	0.4882	9.7633%	1.4300
15	13.0112	0.3802	2.5348%	2.7769
25	25.9480	1.1424	4.5695%	2.7853
35	38.5135	0.9581	2.7373%	5.1048
45	45.3026	1.2993	2.8872%	2.1848

Table 31. Results for the OH-6A helicopter at 12 ft (SL); network configuration 14-25-25-1; NCD learning.

b. 14-25-25-1 BPNN Ext. DBD

Table 32 and Figure 37 present the findings for this configuration. Results show that the RMS error is 0.062. The maximum error is equal to 6 knots (at 35 knots). The maximum error percentage is 12 % and occurs at 5 knots. The network performance is quite good at 45 knots with an error SD of 0.8 knots. We note that the overall error SD of 0.9678 is slightly higher than that of obtained with the NCD scheme.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.9678			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	4.9451	0.6122	12.2449%	1.1009
15	15.4143	0.6925	4.6168%	1.6649
25	27.1087	1.1644	4.6577%	3.7000
35	38.7739	1.6571	4.7345%	5.9939
45	45.7621	0.8066	1.7952%	2.0848

Table 32. Results for the OH-6A helicopter at 12 ft (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

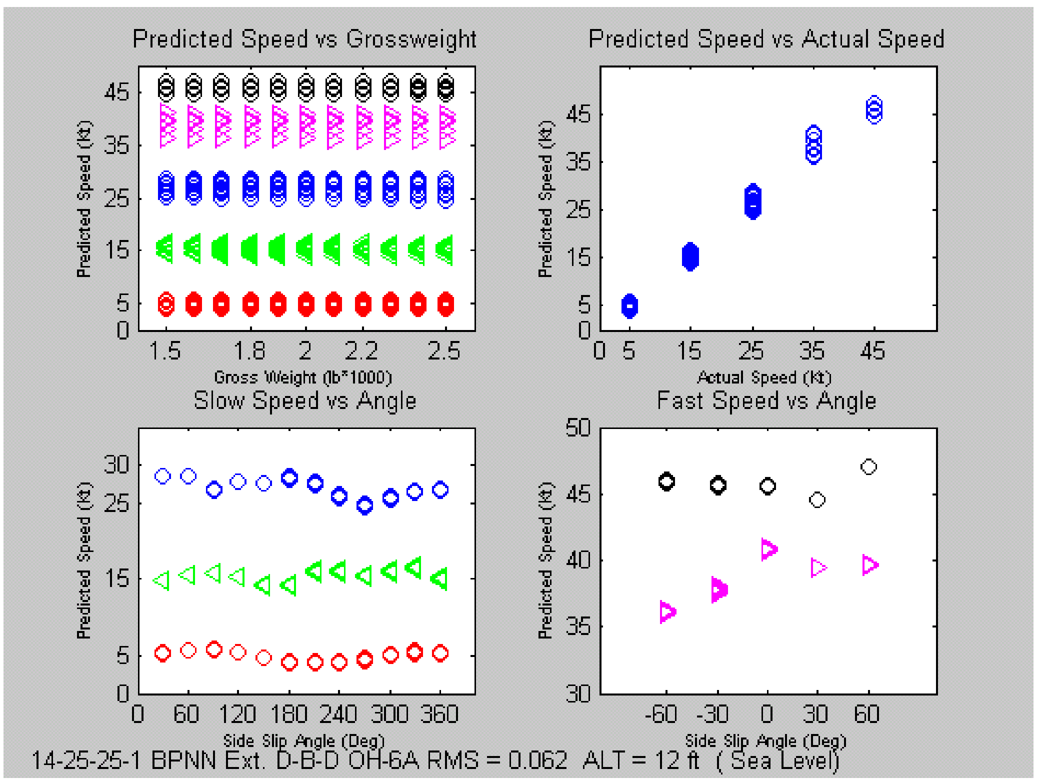


Figure 37. Results for the OH-6A helicopter at 12 ft (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

c. 14-25-25-1 BPNN NCD (Pruned)

Figure 38 and Table 33 present the results for this setup. Results show that the RMS error is 0.0674 and the maximum error of 4.4 knots occurs at 35 knots. Note that the maximum error is about 2 knots for all other speeds. The error SD is the largest at 25 and 45 knots. The overall network error SD is 0.8803 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.8803			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	4.1006	0.6088	12.1766%	1.9135
15	12.2608	0.3937	2.6246%	3.4638
25	25.5791	1.1830	4.7320%	2.5694
35	38.5929	0.5935	1.6956%	4.4157
45	45.1430	1.4917	3.3148%	2.0239

Table 33. Results for the OH-6A helicopter at 12 ft (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

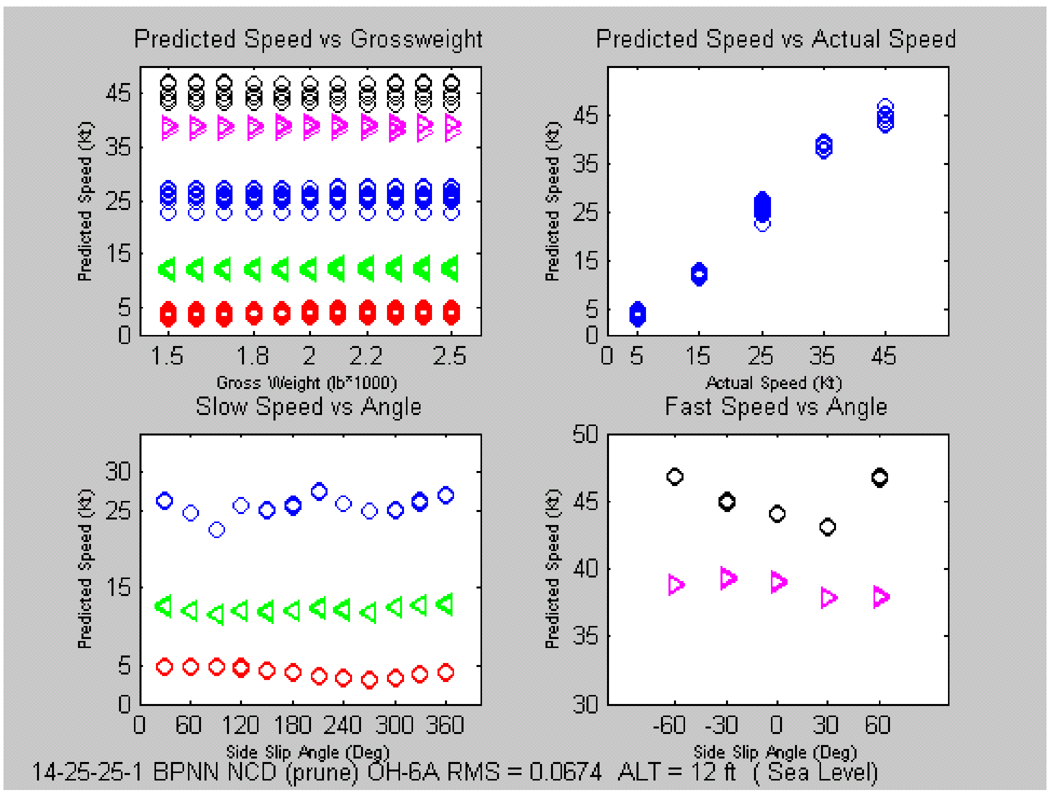


Figure 38. Results for the OH-6A helicopter at 12 ft (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

3. OH-6A Baseline Data Analysis at Sea Level

The baseline data set was formed by combining both 12 ft. and 100 ft. data sets. First the network was trained with this combined data set, and the network performance was measured with the baseline test set. Then, using single condition test data sets separately, the network performance was evaluated for IGE and OGE conditions.

a. 14-25-25-1 BPNN NCD

Results for this configuration are shown in Figure 39 and Table 34. They show that the RMS error is 0.0622. The absolute maximum speed error occurs at 35 knots and is equal to 4.82 knots. The airspeed error SD at 1 σ is more than 1 knot at the speed of 25 knots and over. The overall error SD is 1.2 knots.

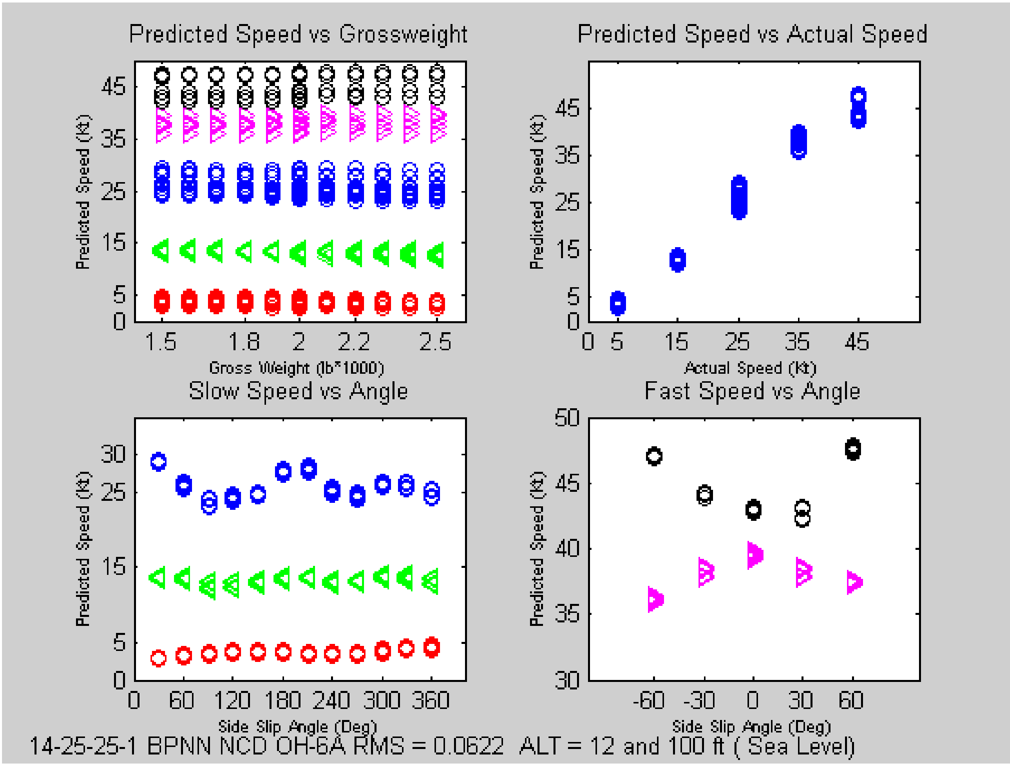


Figure 39. Results for the OH-6A helicopter with baseline data (SL); network configuration 14-25-25-1; NCD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.1926			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.6199	0.4348	8.6964%	2.3155
15	13.2606	0.4590	3.0599%	2.9471
25	25.8467	1.6476	6.5904%	4.3737
35	37.8501	1.1617	3.3190%	4.8224
45	44.8255	2.0805	4.6234%	2.8179

Table 34. Results for the OH-6A helicopter with baseline data (SL); network configuration 14-25-25-1; NCD learning rule.

b. 14-25-25-1 BPNN Ext. DBD

Table 35 and Figure 40 present the results for this setup. Results show that the RMS error is 0.0687. The maximum speed error observed is 7.5 knots for 35 knots. At most speeds, the error SD at 1 σ is more than 1.2 knots. Hence, the error percentages are also larger when compared with other network performances. The overall error SD is 1.4071 knots.

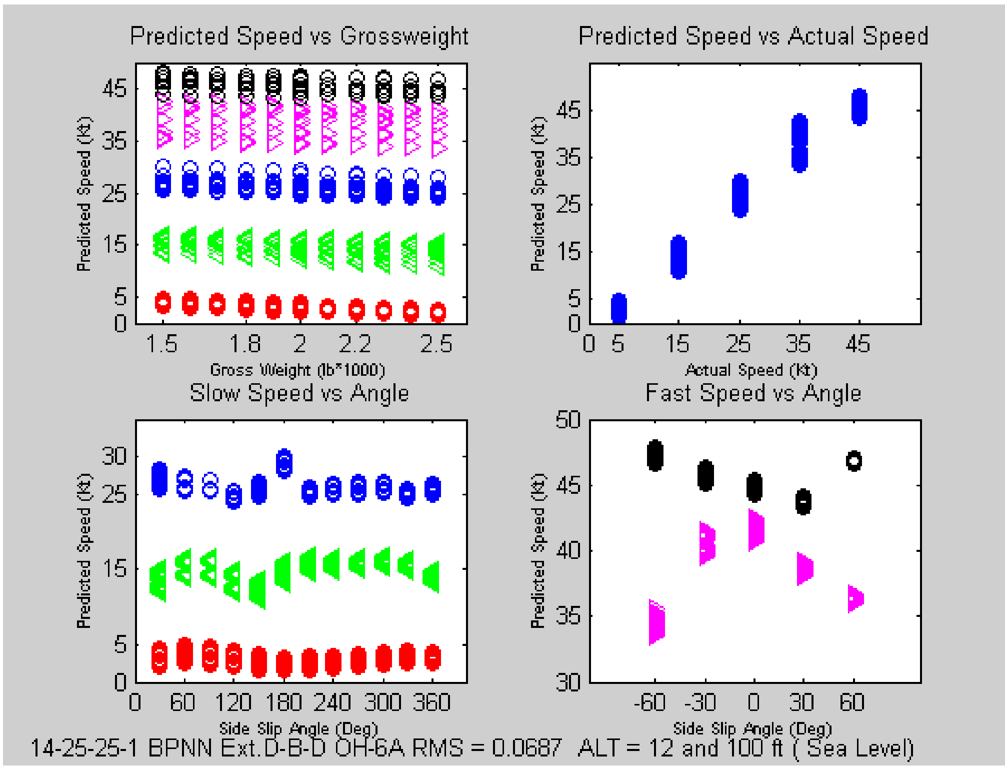


Figure 40. Results for the OH-6A helicopter with baseline data (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.4071			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.9860	0.7204	14.4071%	3.5396
15	14.7369	1.3806	9.2038%	4.0839
25	26.0708	1.2023	4.8092%	4.9530
35	38.3260	2.6810	7.6601%	7.4491
45	45.6813	1.3554	3.0120%	2.9237

Table 35. Results for the OH-6A helicopter with baseline data (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

c. *14-25-25-1 BPNN NCD (Pruned)*

Findings for this configuration are presented in Table 36 and Figure 41. The pruning disabled 3 PEs in the first hidden layer. The RMS error is 0.0611 and the maximum speed error is 4.98 knots at 25 knots. The error SD is greater than 1.2 knots for the speeds 25 knots and higher. The overall error SD is 1.1689 knots.

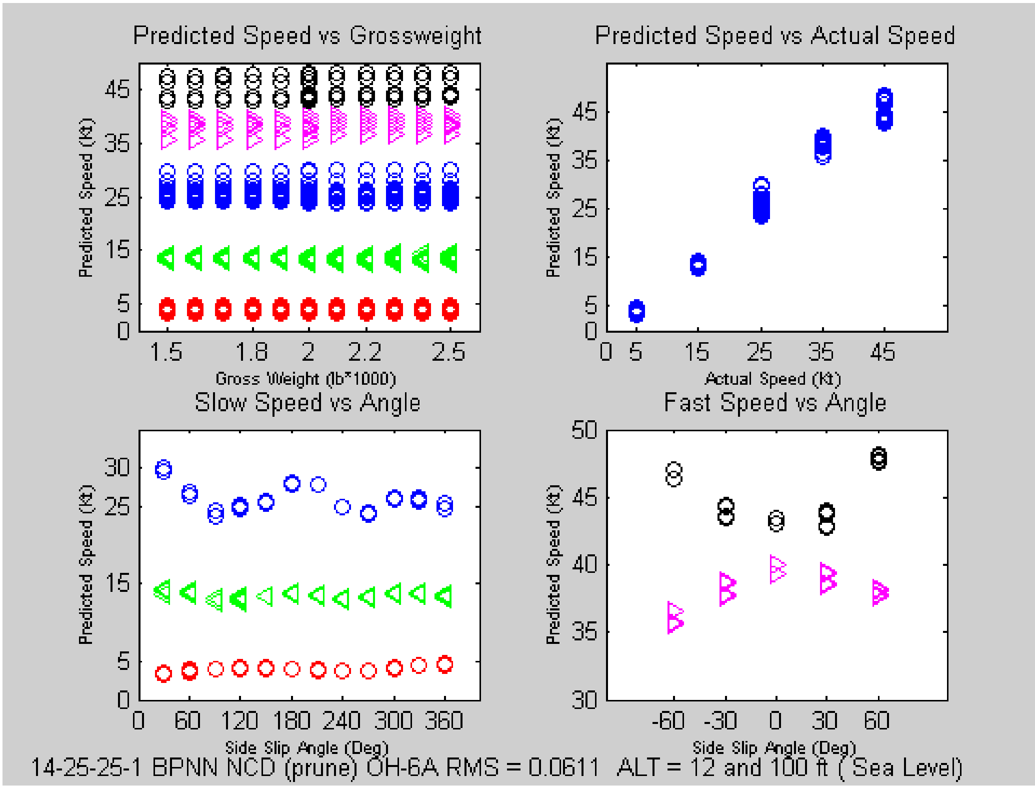


Figure 41. Results for the OH-6A helicopter with baseline data (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.1689			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.9731	0.3442	6.8035%	1.7741
15	13.4709	0.3862	2.5744%	2.3807
25	26.1026	1.6586	6.6343%	5.0413
35	38.1568	1.2776	3.6504%	4.9842
45	44.9858	1.9399	4.3109%	3.1513

Table 36. Results for the OH-6A helicopter with baseline data (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

d. OH-6A SL Baseline Data IGE Analysis

(1) 14-25-25-1 NCD IGE. Table 37 and Figure 42 present detailed results for this configuration. Results show that the RMS error is 0.0593. The maximum speed error is 4.5 knots observed for a 35 knots speed. The largest error SD is 2.14 knots at 45 knots. The overall error SD is 1.1501 knots.

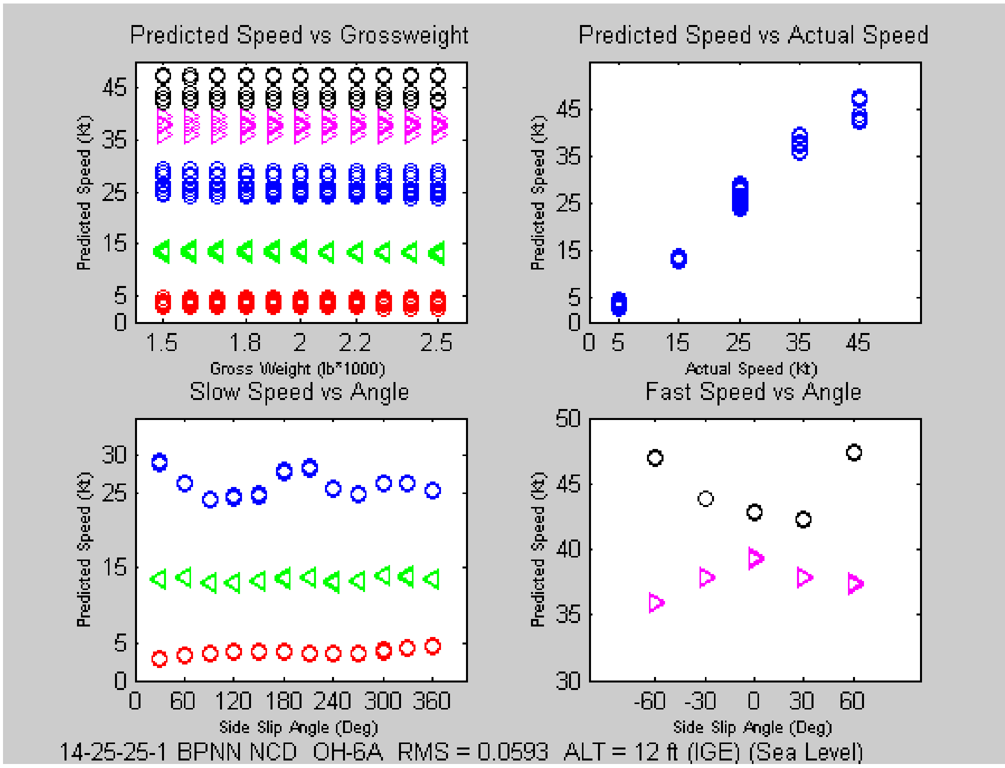


Figure 42. Results for the OH-6A helicopter at 12 ft with baseline data (SL); network configuration 14-25-25-1; NCD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.1501			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.7499	0.4647	9.2946%	2.2505
15	13.4738	0.3264	2.1762%	2.1856
25	26.0936	1.5397	6.1587%	4.2902
35	37.6622	1.1159	3.1883%	4.4943
45	44.6424	2.1425	4.7612%	2.8106

Table 37. Results for the OH-6A helicopter at 12 ft with baseline data (SL); network configuration 14-25-25-1; NCD learning rule.

(2) 14-25-25-1 NCD (Pruned) IGE. Results are given in Table 38 and Figure 43. They show that the RMS error is 0.0592. The maximum speed error is 4.5 knots observed for a 35 knots speed. The overall error SD is 1.155 knots. We note that no significant improvement was obtained when compared with the un-pruned NCD scheme.

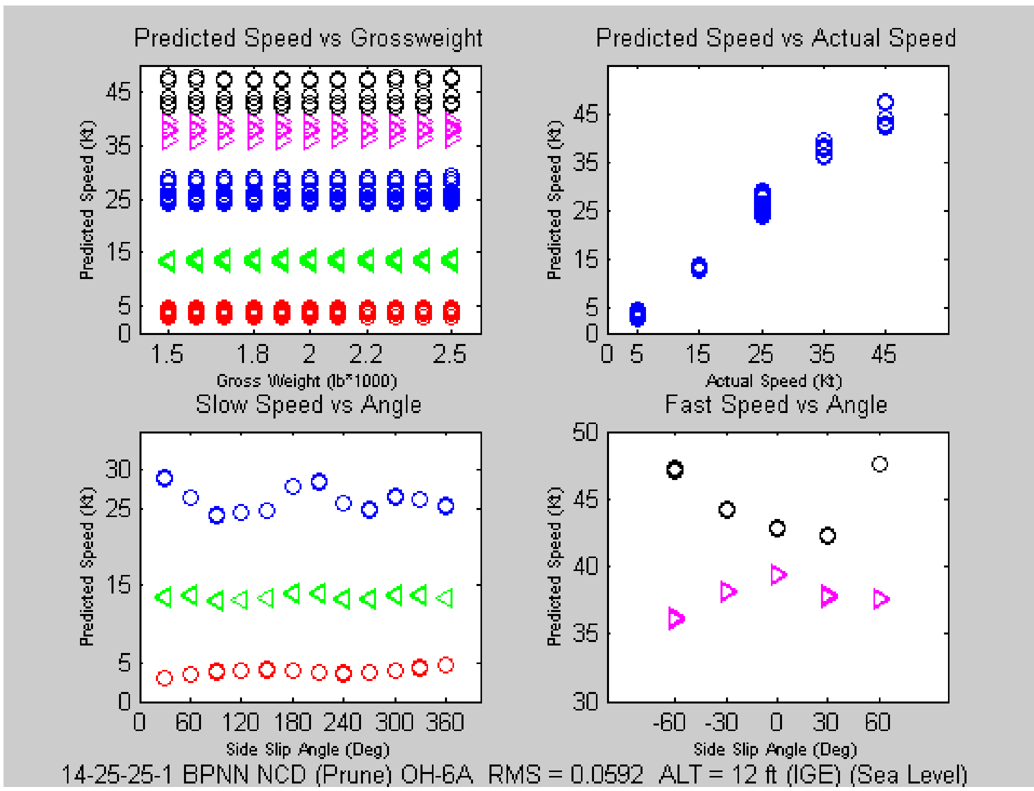


Figure 43. Results for the OH-6A helicopter at 12 ft with baseline data (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.1549			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error At 1 σ	Abs. Maximum Error (kts)
5	3.8734	0.4085	8.1709%	2.0251
15	13.5302	0.3172	2.1143%	2.0447
25	26.1493	1.5478	6.1914%	4.1953
35	37.7847	1.0744	3.0696%	4.5171
45	44.7929	2.2042	4.8981%	2.8103

Table 38. Results for the OH-6A helicopter at 12 ft with baseline data (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

(3) 14-25-25-1 Ext. DBD IGE. Results are illustrated in Figure 44 and Table 39. They show that the RMS error is 0.0686. The maximum speed error is 7.4491 knots observed for a 35 knots speed. We note that neither the prediction of low speed nor the prediction of fast speed is as good as that obtained with the NCD setup. The overall error SD is 1.3293 knots.

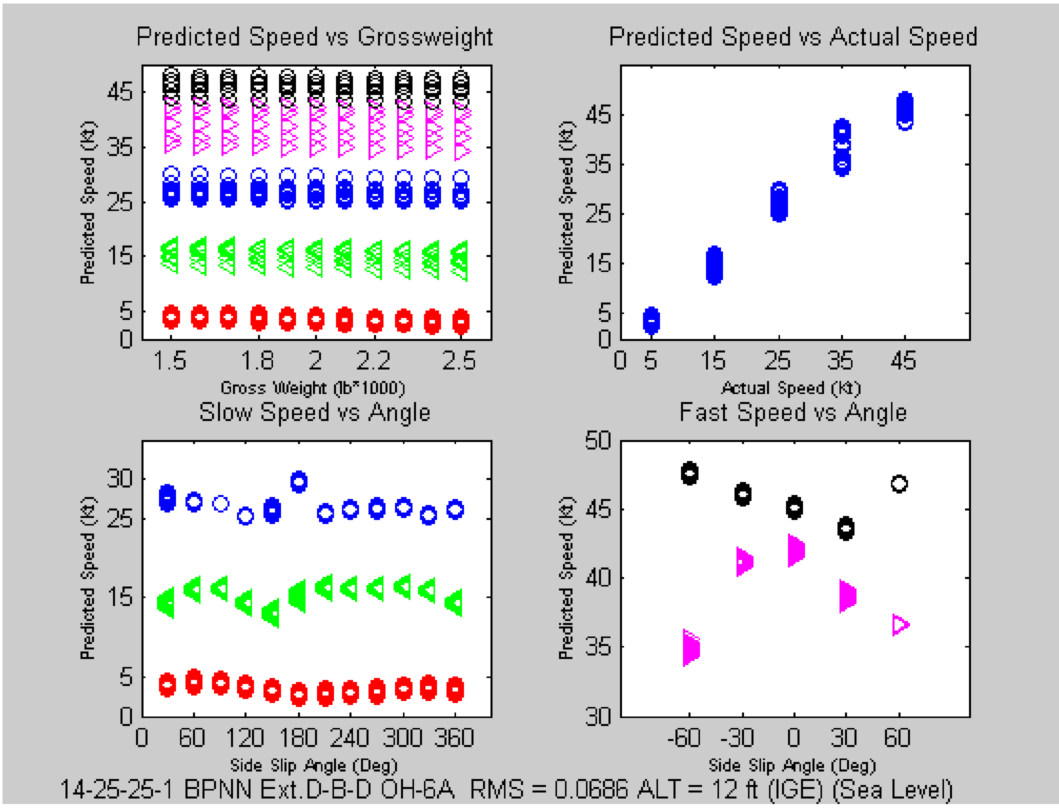


Figure 44. Results for the OH-6A helicopter at 12 ft with baseline data (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.3293			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.4696	0.5957	11.9134%	2.7024
15	15.3847	1.1166	7.4493%	2.5874
25	26.5757	1.1507	4.6029%	4.9330
35	38.6624	2.7151	7.7574%	7.4491
45	45.8415	1.4335	3.1855%	2.9237

Table 39. Results for the OH-6A helicopter at 12 ft with baseline data (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

e. OH-6A SL Baseline Data OGE Analysis

(1) 14-25-25-1 Ext. DBD OGE. Results for this network are depicted in Figure 45 and Table 40. They show that the RMS error is 0.0687. The maximum speed error is 6.6 knots observed for a 35 knots speed. The error SD is over 1 knot almost at all speeds. The overall error SD is 1.3015 knots.

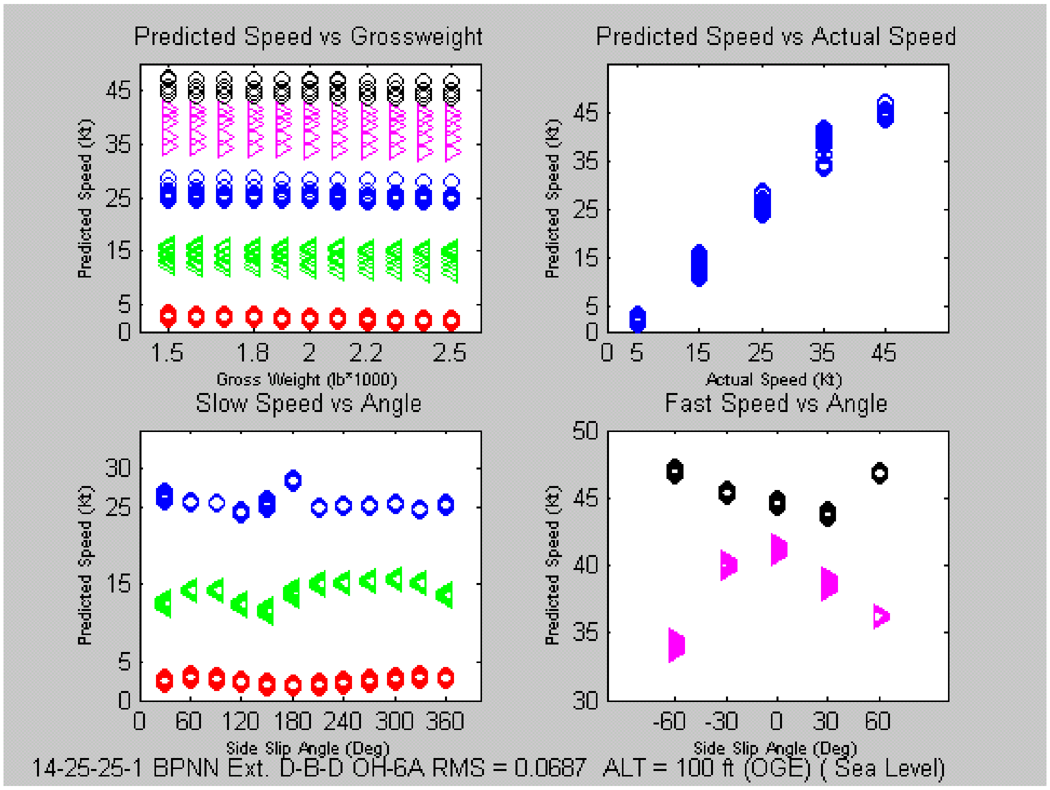


Figure 45. Results for the OH-6A helicopter at 100 ft with baseline data (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.3015			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.5071	0.4741	9.4826%	3.5396
15	14.0892	1.3168	8.7787%	4.0839
25	25.5659	1.0316	4.1264%	3.7463
35	37.9896	2.6280	7.5087%	6.6395
45	45.5210	1.2654	2.8119%	2.2839

Table 40. Results for the OH-6A helicopter at 100 ft with baseline data (SL); network configuration 14-25-25-1; Ext. DBD learning rule.

(2) 14-25-25-1 BPNN NCD OGE. Figure 46 and Table 41 present the results for this setup. Results show that the RMS error is 0.0671. The maximum speed error is 4.8 knots observed for a 35 knots speed. The error SD is above 1 knot for speeds 25 knots and higher. The overall error SD is 1.2022 knots.

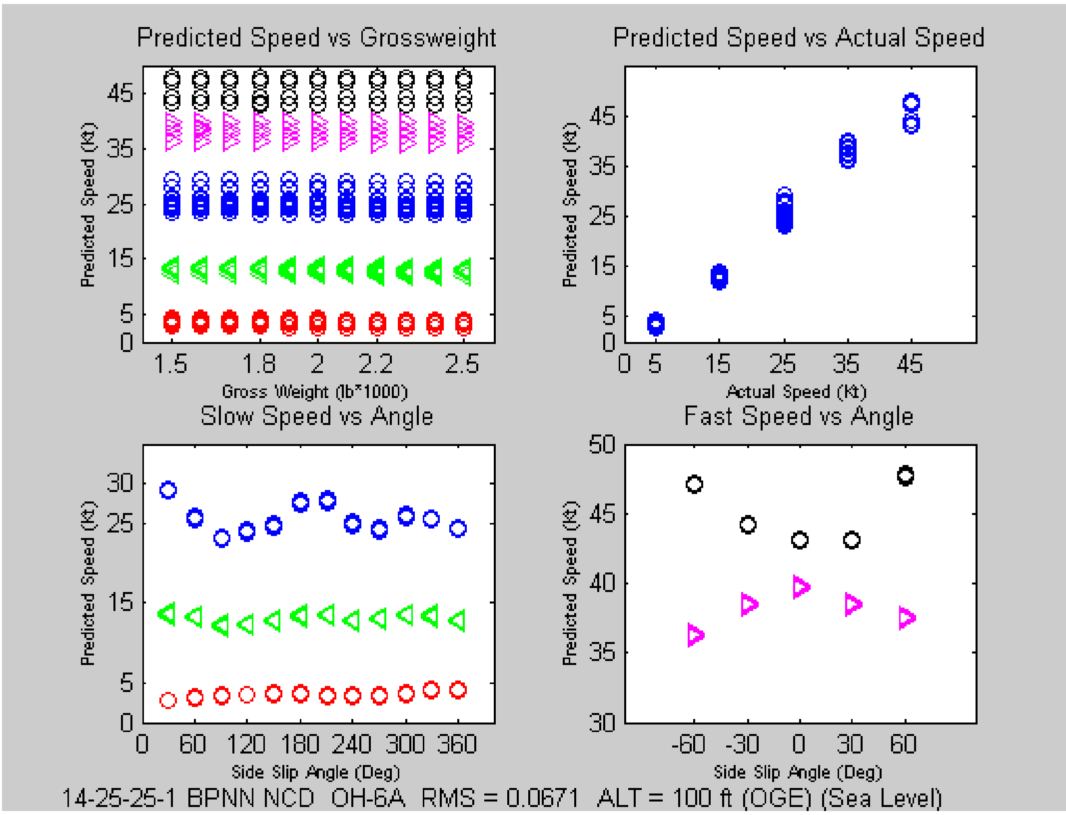


Figure 46. Results for the OH-6A helicopter at 100 ft with baseline data (SL); network configuration 14-25-25-1; NCD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.2022			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.4898	0.3599	7.1982%	2.3155
15	13.0474	0.4741	3.1618%	2.9471
25	25.5998	1.7194	6.8786%	4.3737
35	38.038	1.1861	3.3887%	4.8224
45	45.0086	2.0195	4.4874%	2.8176

Table 41. Results for the OH-6A helicopter at 100 ft with baseline data (SL); network configuration 14-25-25-1; NCD learning rule.

(3) 14-25-25-1 BPNN NCD (Pruned) OGE. Results are depicted in Figure 47 and in Table 42. Results show that the RMS error is 0.0650. The maximum speed error is about 5 knots at 35 knots. The maximum error is about 2 knots for all other speeds. One PE in the first hidden layer was disabled. The overall error SD is 1.2 knots. Note that performance of this setup is better than that of the Ext. DBD configuration.

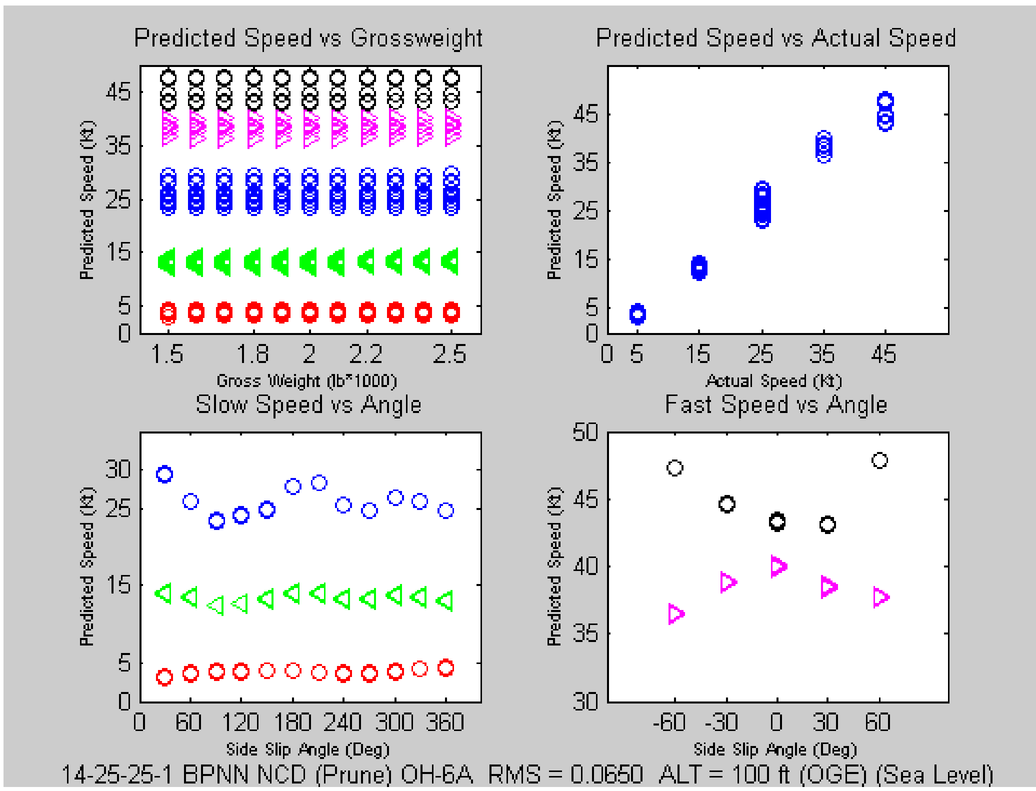


Figure 47. Results for the OH-6A helicopter at 100 ft with baseline data (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.2096			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.8439	0.3031	6.0621%	1.8782
15	13.3969	0.5177	3.4513%	2.5735
25	25.9133	1.7329	6.9317%	4.5386
35	38.2975	1.184	3.3828%	5.1005
45	45.2226	2.0303	4.5118%	2.8686

Table 42. Results for the OH-6A helicopter at 100 ft with baseline data (SL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

4. OH-6A Out of Ground Effect Analysis at High Altitude

The simulator was run at a pressure altitude of 6000 feet for high altitude analysis. At that pressure altitude, the altitude AGL was set to 100 ft. for the OGE condition, which is the same AGL altitude, that the sea level OGE analysis was performed at.

a. 14-25-25-1 BPNN NCD

Results are presented in Figure 48 and Table 43. They show that the RMS error for this setup is 0.0485. The maximum speed error, which is 2.47 knots, and the maximum error percentage equal to 9.4 % are both observed for a 5 knots speed. The maximum airspeed error SD of 1.1356 knots occurs at 35 knots. The overall error SD is 0.6637 knots.

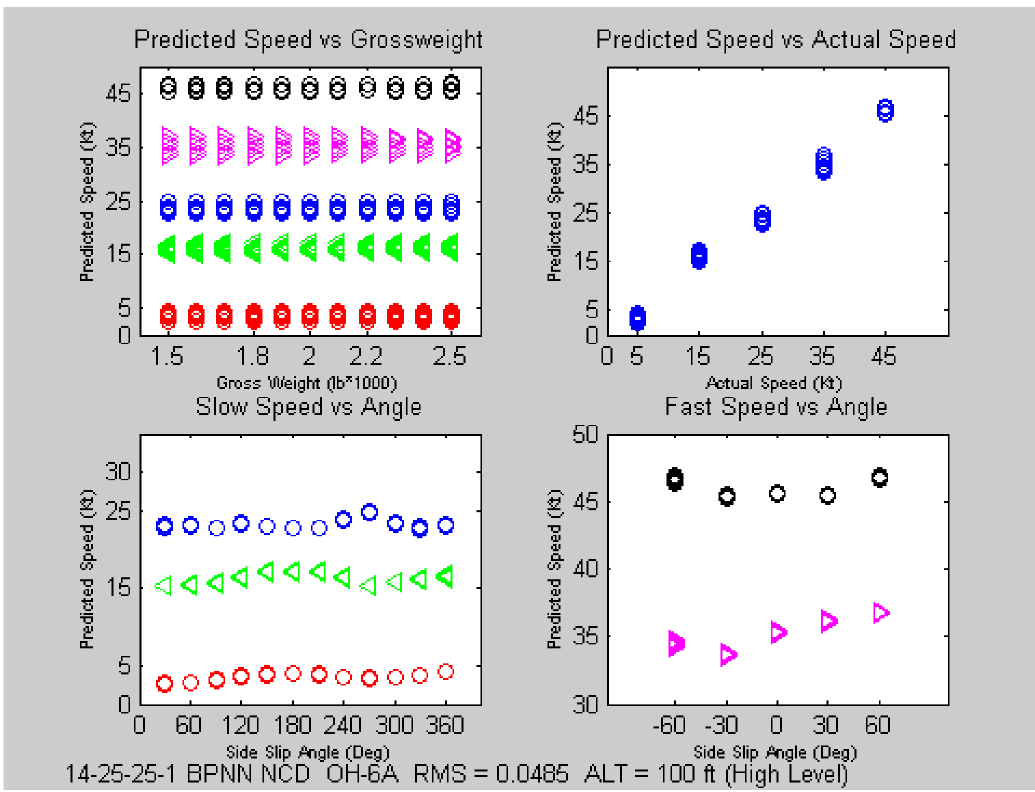


Figure 48. Results for the OH-6A helicopter at 100 ft (HL); network configuration 14-25-25-1; NCD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.6637			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.5101	0.4738	9.4794%	2.4762
15	16.209	0.6861	4.5744%	2.2914
25	23.272	0.5601	2.2405%	2.3443
35	35.288	1.1356	3.2444%	1.8547
45	45.985	0.6179	1.3732%	1.8831

Table 43. Results for the OH-6A helicopter at 100 ft (HL); network configuration 14-25-25-1; NCD learning rule.

b. 14-25-25-1 BPNN Ext. DBD

Figure 49 and Table 44 present the findings for this configuration. Results show that the RMS error is 0.0958. This is one of the largest errors observed for the OH-6A analysis. The maximum error is 6.36 knots, which is more than twice the error of the NCD setup. The error percentage at 5 knots is better than that of the NCD configuration. The overall error SD is 1.022 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.022			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.1676	0.2838	5.6769%	2.3937
15	19.875	0.7859	5.2395%	6.3677
25	23.339	1.2915	5.1659%	5.3069
35	33.995	1.9872	5.6777%	3.4085
45	45.699	0.3127	0.6917%	1.1757

Table 44. Results for the OH-6A helicopter at 100 ft (HL); network configuration 14-25-25-1; Ext. DBD learning rule.

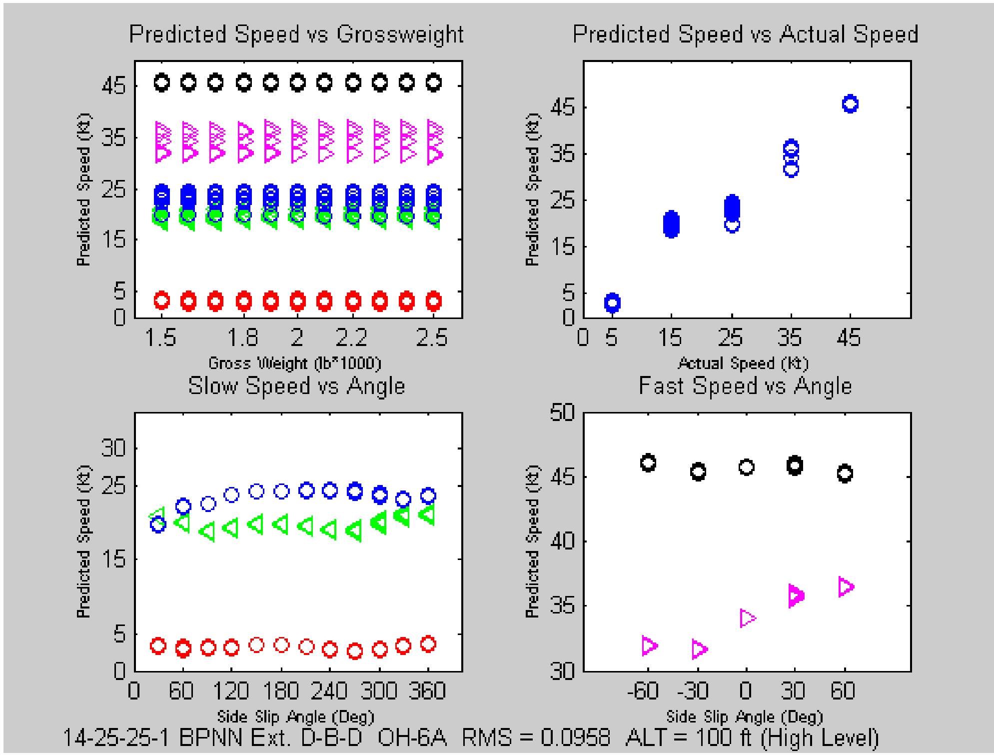


Figure 49. Results for the OH-6A helicopter at 100 ft (HL); network configuration 14-25-25-1; Ext. DBD learning rule.

c. 14-25-25-1 BPNN NCD (Pruned)

Figure 50 and Table 45 present the findings for this configuration. The pruning disabled 1 PE of the first hidden layer. Results show that the RMS error is 0.0475. The maximum speed error is 2.85 knots observed for 5 knots speed. The results for fast speed predictions are better than the those at low speeds. The overall error SD is 0.6401 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.6401			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.8147	0.3546	7.0932%	2.8415
15	14.812	0.7660	5.1070%	1.4709
25	24.0	0.7102	2.8410%	2.2145
35	36.208	0.7016	2.0047%	2.0621
45	46.040	0.6219	1.3821%	2.0527

Table 45. Results for the OH-6A helicopter at 100 ft (HL); network configuration 14-25-25-1; NDC (Pruned) learning rule.

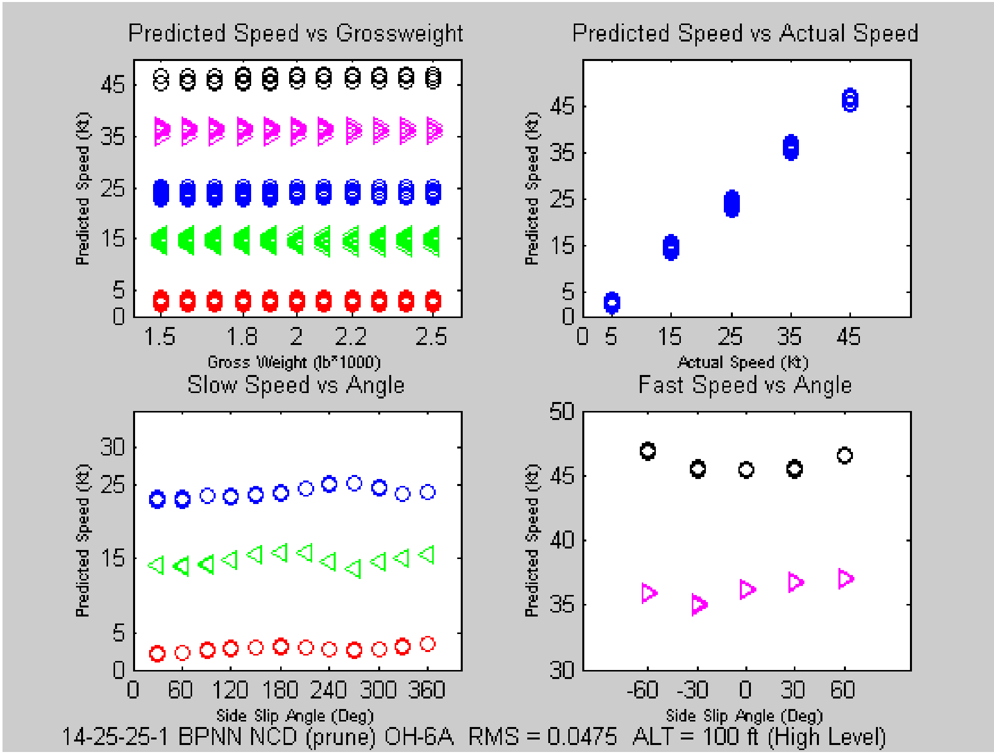


Figure 50. Results for the OH-6A helicopter at 100 ft (HL); network configuration 14-25-25-1; NDC (Pruned) learning rule.

5. OH-6A In-Ground Effect Analysis at High Altitude

In-ground effect analyses of the OH-6A at high altitude were implemented by setting the simulator pressure altitude to 6000 feet. At that pressure altitude, the altitude AGL was set to 12 ft. for the IGE condition, which is the same AGL altitude at which sea level IGE analyses were performed.

a. 14-25-25-1 BPNN NCD

Figure 51 and Table 46 present the findings for this configuration. Results show that the RMS error is 0.0516, and the maximum speed error is 2.596 knots for a 5 knots speed. The maximum error percentage equal to 9.12 % is also observed at that speed. The maximum airspeed error SD of 1.20 knots occurs at 35 knots. The overall error SD is 0.6855 knots.

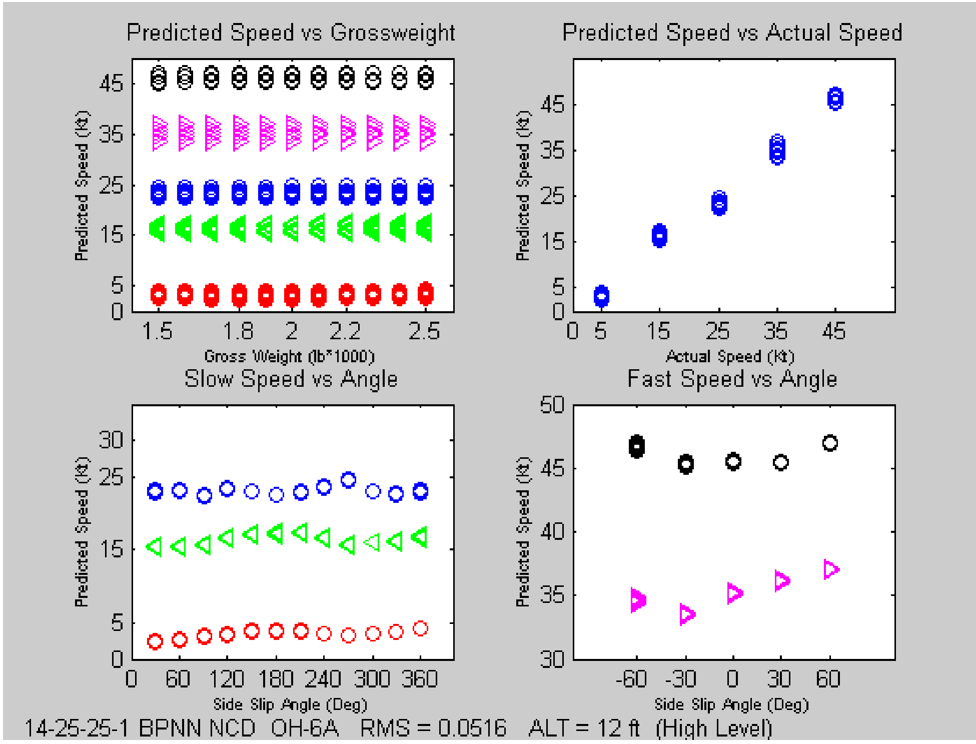


Figure 51. Results for the OH-6A helicopter at 12 ft (HL); network configuration 14-25-25-1; NDC learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.6855			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.4367	0.4811	9.1210%	2.5962
15	16.3790	0.6755	4.503%	2.4700
25	23.1736	0.5637	2.2547%	2.5948
35	35.3076	1.2062	3.4462%	2.0407
45	46.0330	0.7075	1.5722%	2.0734

Table 46. Results for the OH-6A helicopter at 12 ft (HL); network configuration 14-25-25-1; NDC learning rule.

b. 14-25-25-1 BPNN Ext. DBD

Results for this configuration are presented in Figure 52 and Table 47. They show that the RMS error is 0.0865, which is larger than that of the NCD setup. Although the maximum error is 5.84 knots for a 15 knots speed, the maximum error percentage equal to 5.53 % is significantly smaller than that of the NCD scheme. The maximum airspeed error SD of 1.71 knots occurs at 35 knots. Note that the airspeed estimation is poor at low speeds and it is best at high speeds when the gross weight of the helicopter is less than 2100 lb. The overall error SD is 0.9556 knots.

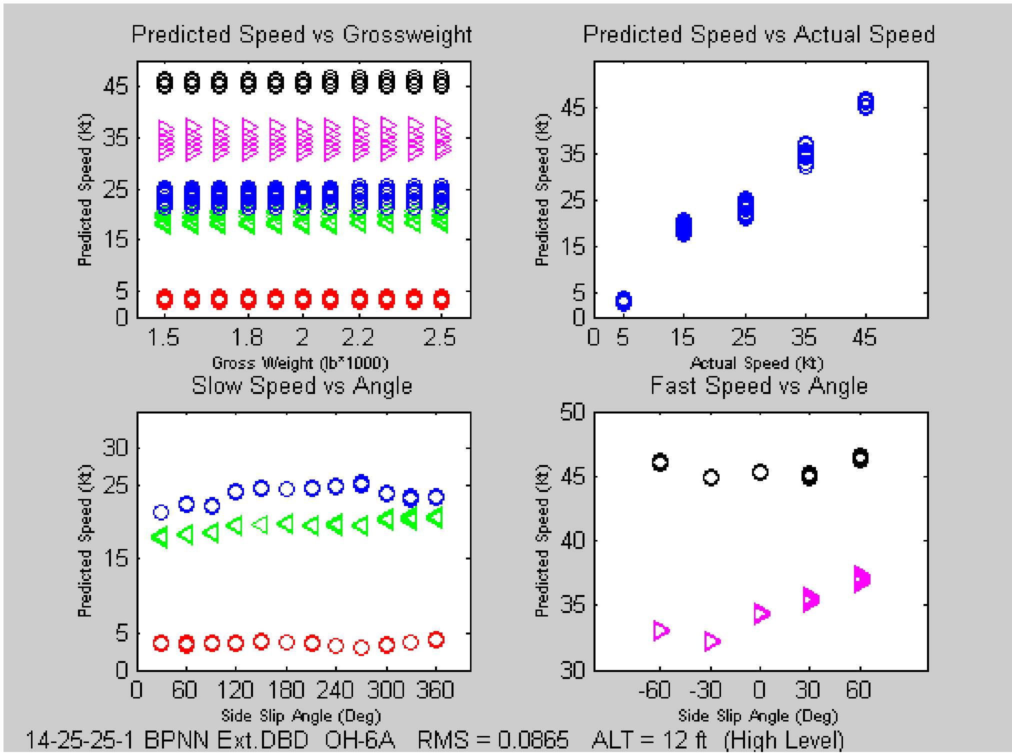


Figure 52. Results for the OH-6A helicopter at 12 ft (HL); network configuration 14-25-25-1; Ext. DBD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.9556			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.5658	0.2767	5.5336%	1.9728
15	19.5361	0.8258	5.5054%	5.8437
25	23.7345	1.1813	4.7250%	3.7400
35	34.4205	1.7182	4.9093%	2.8189
45	45.5856	0.6026	1.3391%	1.6418

Table 47. Results for the OH-6A helicopter at 12 ft (HL); network configuration 14-25-25-1; Ext. DBD learning rule.

c. *14-25-25-1 BPNN NCD (Pruned)*

Results are presented in Figure 53 and Table 48. One PE in the first hidden layer was disabled. Results show that the RMS error is 0.0443, which is the best result obtained for the analysis of this altitude. The maximum speed error is 2.58 knots (at 5 knots) and the maximum error percentage is 5.59 %. The maximum airspeed error SD equal to 0.5883 knots occurs at 15 knots speed. The overall error SD is 0.6401 knots.

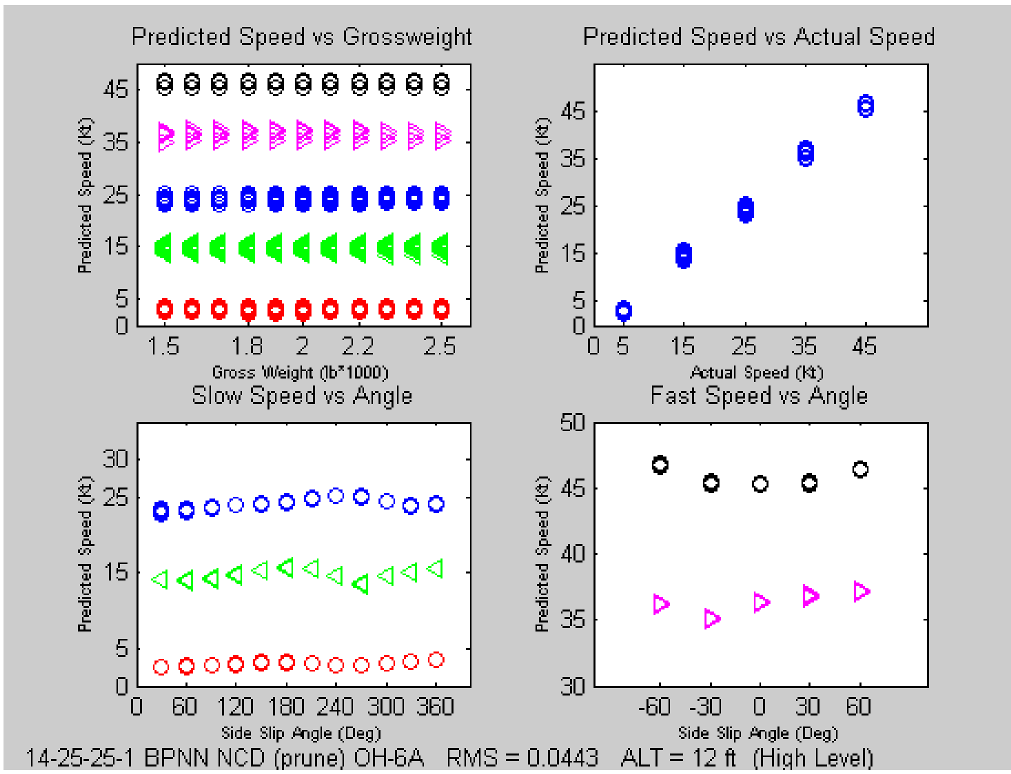


Figure 53. Results for the OH-6A helicopter at 12 ft (HL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.443			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.9434	0.2797	5.5945%	2.5801
15	14.7437	0.6863	4.5756%	1.4916
25	24.2145	0.6529	2.6118%	2.0764
35	36.3517	0.6982	1.9949%	2.1910
45	46.8740	0.6187	1.3749%	1.8320

Table 48. Results for the OH-6A helicopter at 12 ft (HL); network configuration 14-25-25-1; NCD (Pruned) learning rule.

6. OH-6A Baseline Data Analysis at High Level

A baseline training data set was formed by combining both 12 ft. and 100 ft. set of high level altitude data. First, the network was trained with this combined data set, and then the network performance was measured using the baseline testing set. Finally, the network performance was evaluated for IGE and OGE conditions, using single condition test data sets separately. Results are given below.

a. 14-25-25-1 BPNN NCD

Results are shown in Figure 54 and Table 49. They show that the RMS error is 0.0507, and the absolute maximum speed error is 3.15 knots (at 25 knots). The airspeed error SD at 1 σ is less than 1 knot at all speeds. The overall error SD is 0.7139 knots.

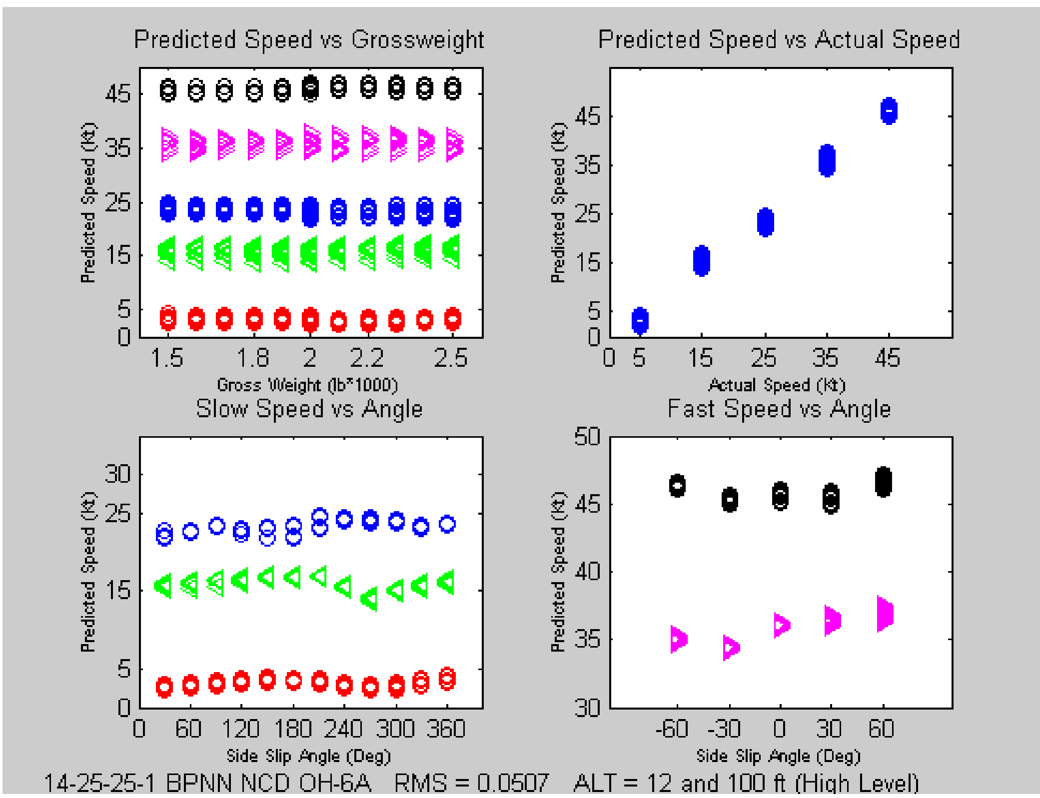


Figure 54. Results for the OH-6A helicopter with baseline data (HL); network configuration 14-25-25-1; NCD learning rule.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7139			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.1503	0.4366	8.7318%	2.7569
15	15.9246	0.8179	5.4525%	2.1210
25	23.3005	0.7533	3.0132%	3.1582
35	35.7437	0.9455	2.7014%	2.5005
45	45.8256	0.6244	1.3876%	2.1869

Table 49. Results for the OH-6A helicopter with baseline data (HL); network configuration 14-25-25-1; NCD learning rule.

b. 14-25-25-1 BPNN Ext. DBD

Results are presented in Figure 55 and Table 50. They show that the RMS error is 0.0772, and the absolute maximum speed error is 5.39 knots, which occurs at 15 knots speed. The maximum airspeed error SD at 1 σ is 1.61 knots (at 25 knots). The prediction accuracy of low speeds is not as good as that of fast speeds. The overall error SD is 1.0209 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.0209			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.7873	0.4398	8.7951%	2.0195
15	19.0707	0.7415	4.9436%	5.3927
25	24.8423	1.6157	6.4628%	3.7711
35	35.460	0.9531	2.7230%	2.0827
45	45.6283	0.8166	1.8147%	1.8147

Table 50. Results for the OH-6A helicopter with baseline data (HL); network configuration 14-25-25-1; Ext. DBD learning rule.

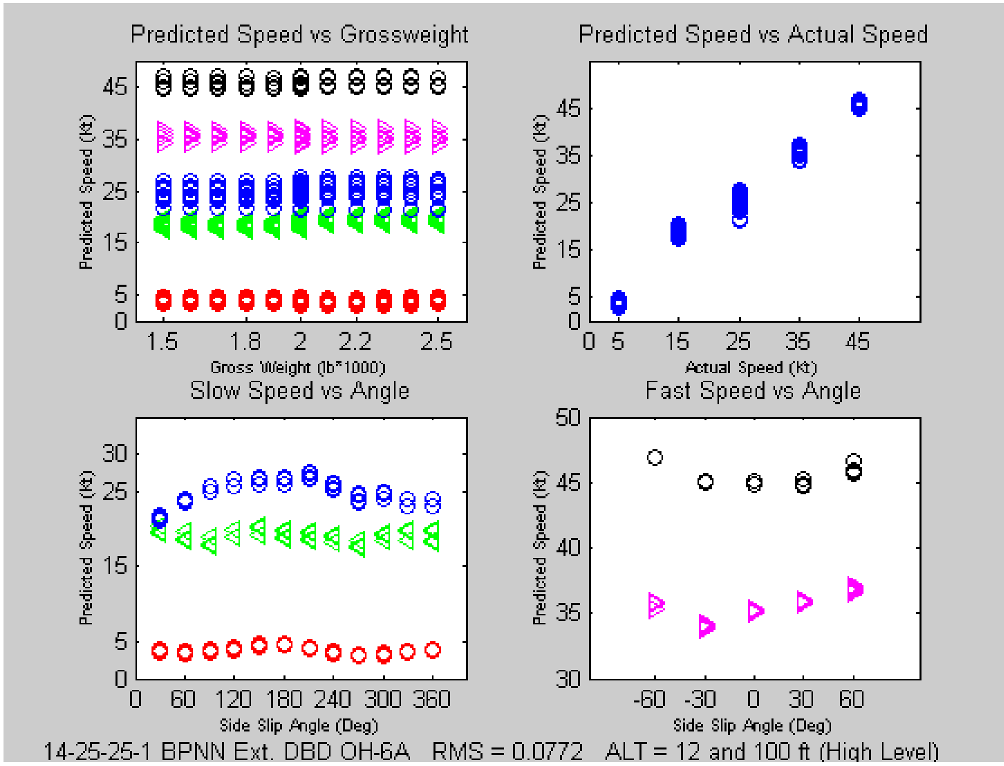


Figure 55. Results for the OH-6A helicopter with baseline data (HL); network configuration 14-25-25-1; Ext. DBD learning rule.

c. *14-25-25-1 BPNN Ext. DBD (Pruned)*

The Ext. DBD pruned network results are presented because they were better than those of the NCD pruned network. The RMS error for this configuration is found to be 0.0549. The absolute maximum speed error is 3.43 knots (at 5 knots). The maximum airspeed error SD at 1 σ is less than 1 knot at all speeds. The overall error SD is 0.6505 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.6505			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.3968	0.3774	7.5484%	3.4378
15	15.6340	0.5457	3.6383%	1.4552
25	23.7417	0.8713	3.4852%	3.3192
35	36.3556	0.6385	1.8243%	2.4002
45	45.9307	0.7895	1.7544%	2.1794

Table 51. Results for the OH-6A helicopter with baseline data (HL); network configuration 14-25-25-1; Ext. DBD (Pruned) learning rule.

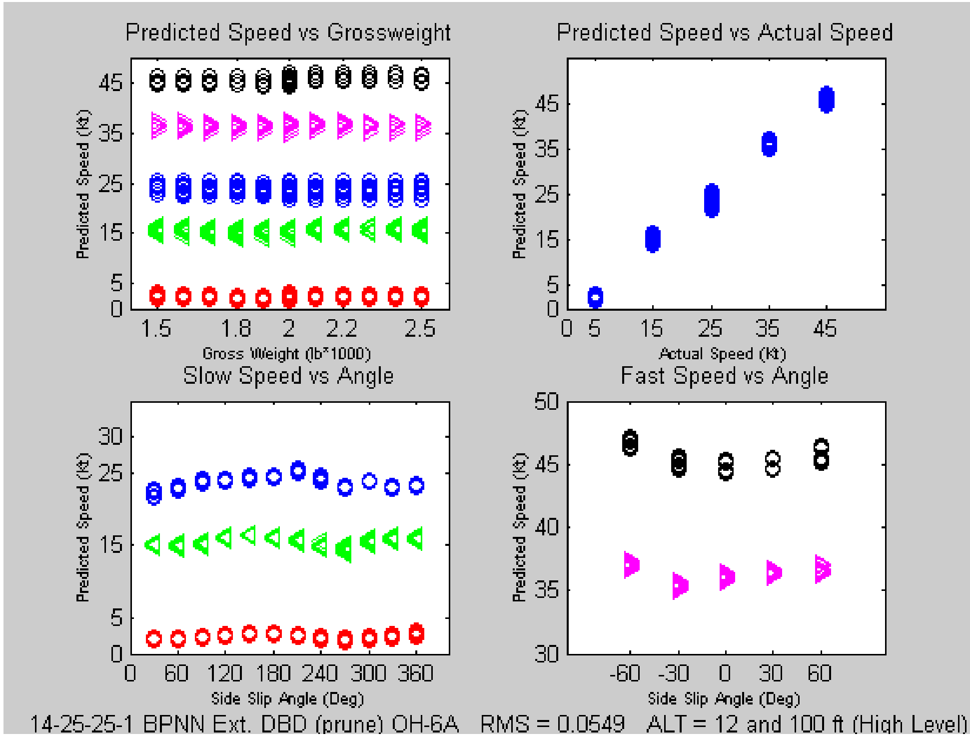


Figure 56. Results for the OH-6A helicopter with baseline data (HL); network configuration 14-25-25-1; Ext. DBD (Pruned) learning rule.

7. OH-6A HL Baseline Data IGE Analysis

a. 14-25-25-1 NCD IGE

The findings for this configuration are presented in Table 57 and Figure 57. Results show that the RMS error is 0.0446 and the absolute maximum speed error is2.49 knots (at5 knots). The airspeed error SD at 1 σ is less than 1 knot at all speeds. The overall error SD is 0.6296 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.6296			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.2718	0.4172	8.8436%	2.4936
15	15.7052	0.7943	5.2950%	1.9305
25	23.5417	0.5634	2.2538%	2.2911
35	35.6807	0.7965	2.2758%	1.8238
45	45.5805	0.5918	1.3150%	1.4307

Table 52. Results for the OH-6A helicopter at 12 ft with baseline data (HL); network configuration 14-25-25-1; NCD learning rule.

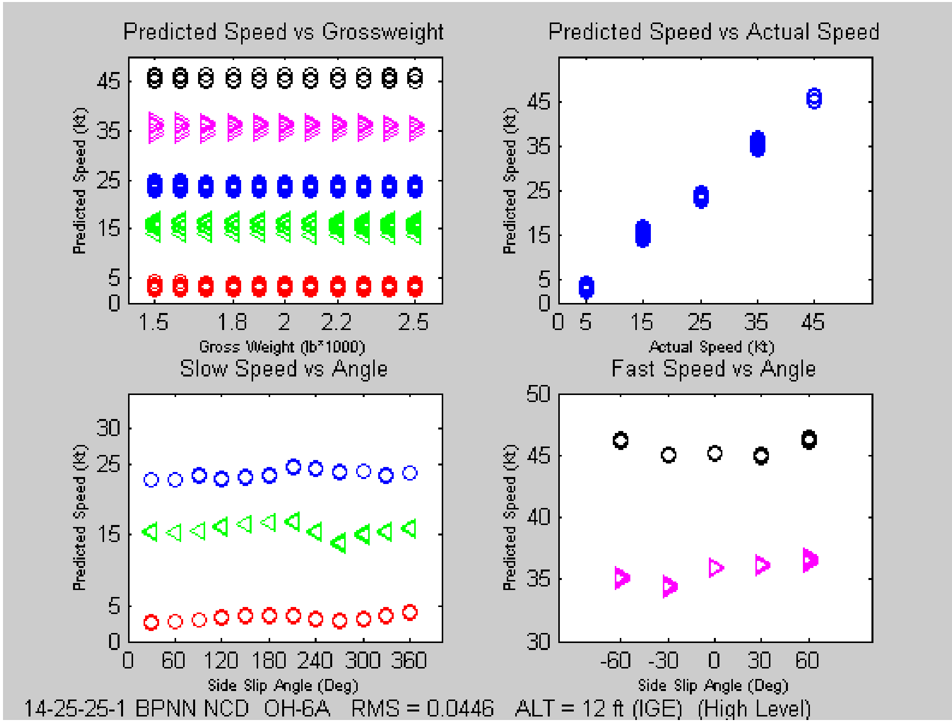


Figure 57. Results for the OH-6A helicopter at 12 ft with baseline data (HL); network configuration 14-25-25-1; NCD learning rule.

b. 14-25-25-1 Ext. DBD IGE

Table 53 and Figure 58 present the findings for this configuration. Results show that the RMS error is 0.0682, and the absolute maximum speed error is 4.55 knots, which occurs at 5 knots speed. The maximum airspeed error SD at 1 σ is 1.47 knots (at 25 knots). The overall error SD is 0.9027 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.9027			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.8334	0.4224	8.4479%	1.8070
15	18.5289	0.4925	3.2834%	4.5586
25	24.4931	1.7556	5.9025%	3.3707
35	35.5342	0.8230	2.3513%	1.7742
45	45.4751	0.7995	1.7767%	1.8916

Table 53. Results for the OH-6A helicopter at 12 ft with baseline data (HL); network configuration 14-25-25-1; Ext. DBD learning rule.

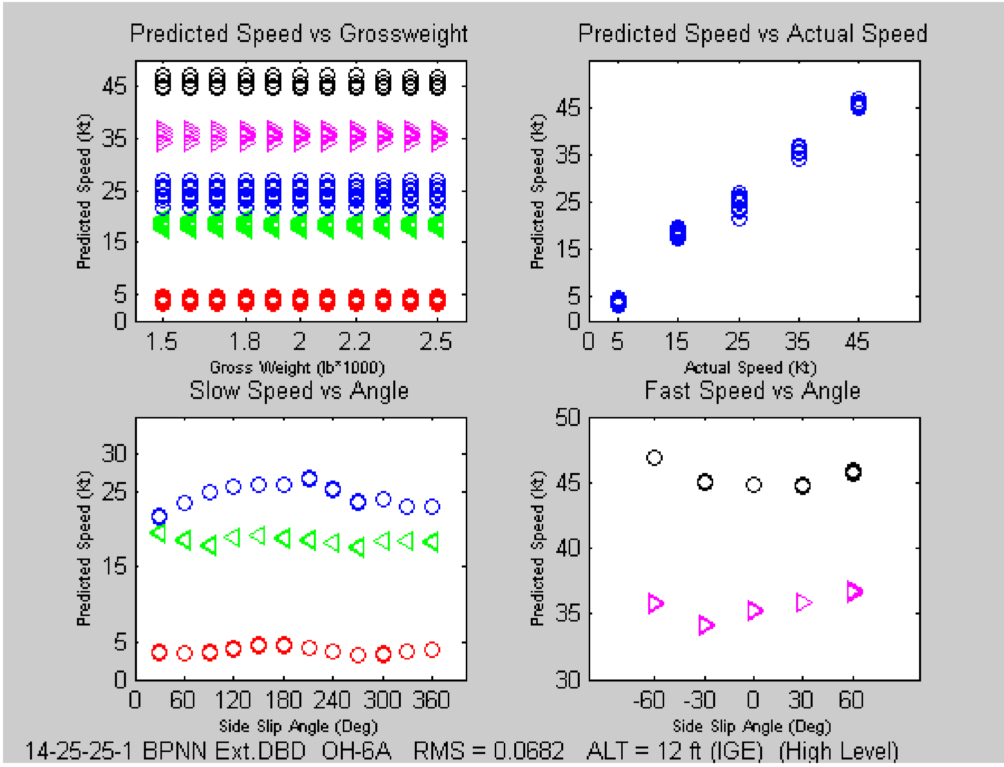


Figure 58. Results for the OH-6A helicopter at 12 ft with baseline data (HL); network configuration 14-25-25-1; Ext. DBD learning rule

c. *14-25-25-1 NCD (Pruned) IGE*

Figure 59 and Table 54 present the findings for this configuration. Results show that the RMS error is 0.0732. The absolute maximum speed error equal to 2.09 knots occurs at 25 knots speed. The maximum airspeed error SD at 1 σ is less than 1 knot at all speeds. The overall error SD is 0.6655 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.6655			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.6158	0.3581	7.1625%	2.0768
15	15.4433	0.8653	5.7688%	1.9782
25	24.1375	0.6593	2.6372%	2.0969
35	35.1451	0.6882	1.9693%	1.3938
45	44.5488	0.6975	1.5501%	1.4001

Table 54. Results for the OH-6A helicopter at 12 ft with baseline data (HL); network configuration 14-25-25-1; NCD learning rule.

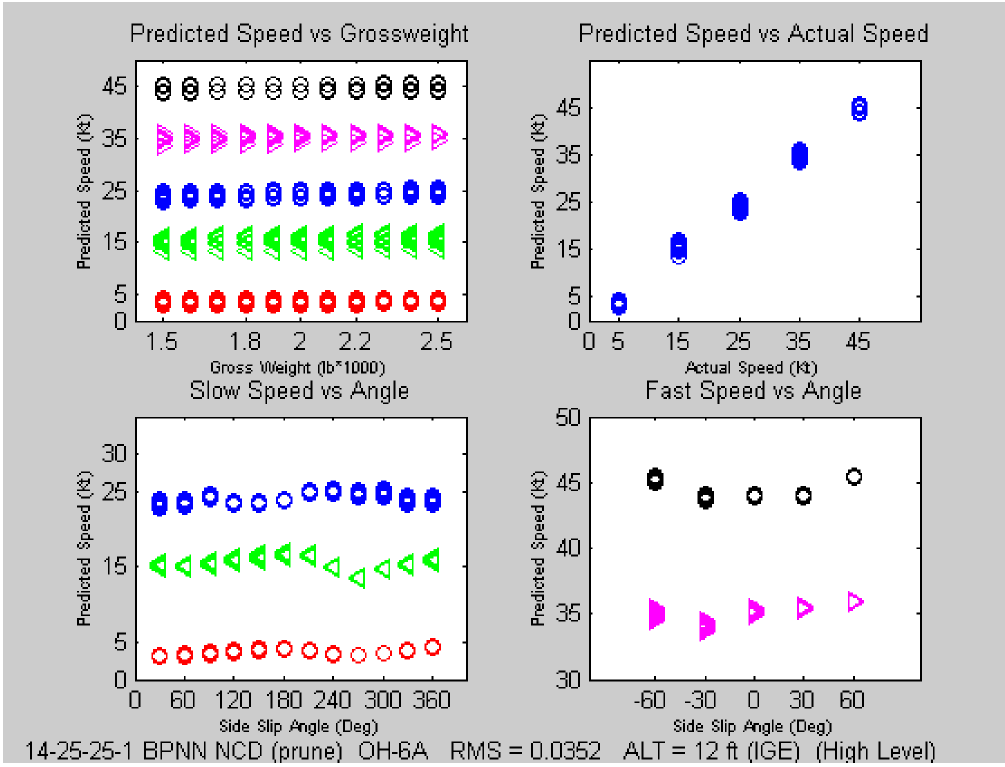


Figure 59. Results for the OH-6A helicopter at 12 ft with baseline data (HL); network configuration 14-25-25-1; NCD (Pruned) learning rule

8. OH-6A HL Baseline Data OGE Analysis

a. 14-25-25-1 NCD OGE

Results for this setup are shown in Figure 60 and Table 55. They show that the RMS error is 0.0561. The absolute maximum error equal to 3.15 knots occurs at 25 knots. The maximum airspeed error SD at 1 σ is 1.07 knots, which is observed for 35 knots speed. The overall error SD is 0.7393 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.7393			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.0288	0.4230	8.4591%	2.7569
15	16.1439	0.7843	5.2287%	2.1210
25	23.0594	0.8394	3.3576%	3.1582
35	35.8068	1.0779	3.0798%	2.5005
45	46.0707	0.5608	1.2461%	2.1869

Table 55. Results for the OH-6A helicopter at 100 ft with baseline data (HL); network configuration 14-25-25-1; NCD learning rule.

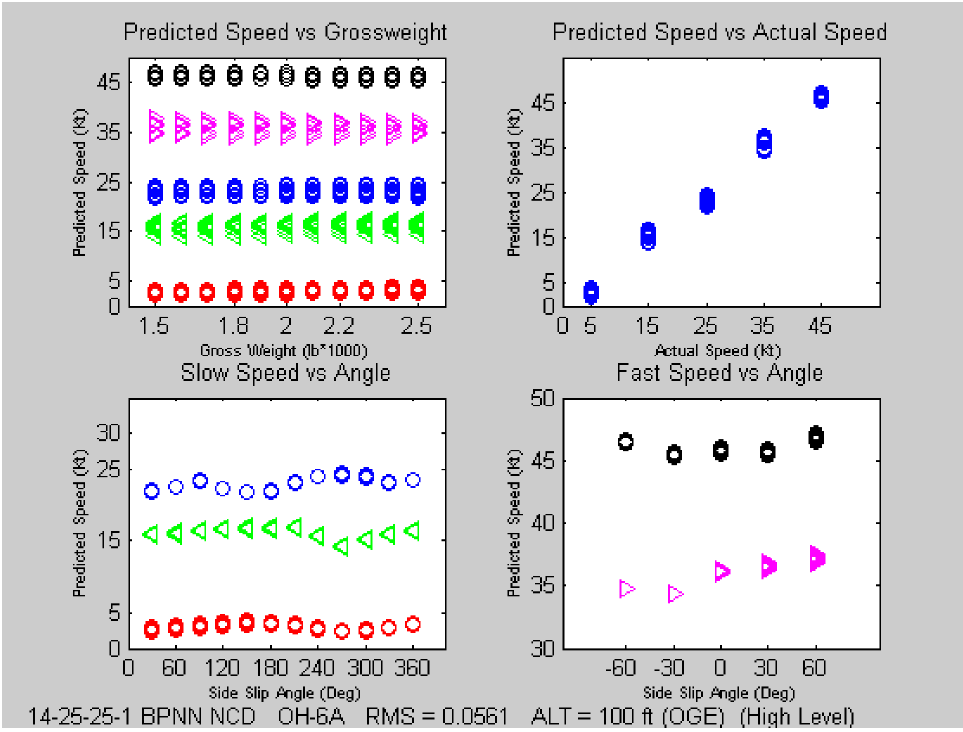


Figure 60. Results for the OH-6A helicopter at 100 ft with baseline data (HL); network configuration 14-25-25-1; NCD learning rule.

b. 14-25-25-1 Ext. DBD OGE

Results are shown in Figure 61 and Table 56. They show that the RMS error is 0.0853 and the absolute maximum speed error is 5.39 knots (at 15 knots). The maximum airspeed error SD at 1 σ equal to 1.67 knots is observed for 25 knots speed. The overall error SD is 1.0233 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 1.0233			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	3.7411	0.4534	9.0672%	2.0195
15	19.6126	0.5194	3.4625%	5.3927
25	25.1916	1.6786	6.7143%	3.7711
35	35.3858	1.070	3.0573%	2.0827
45	45.7815	0.8118	1.8040%	1.9277

Table 56. Results for the OH-6A helicopter at 100 ft with baseline data (HL); network configuration 14-25-25-1; Ext. DBD learning rule.

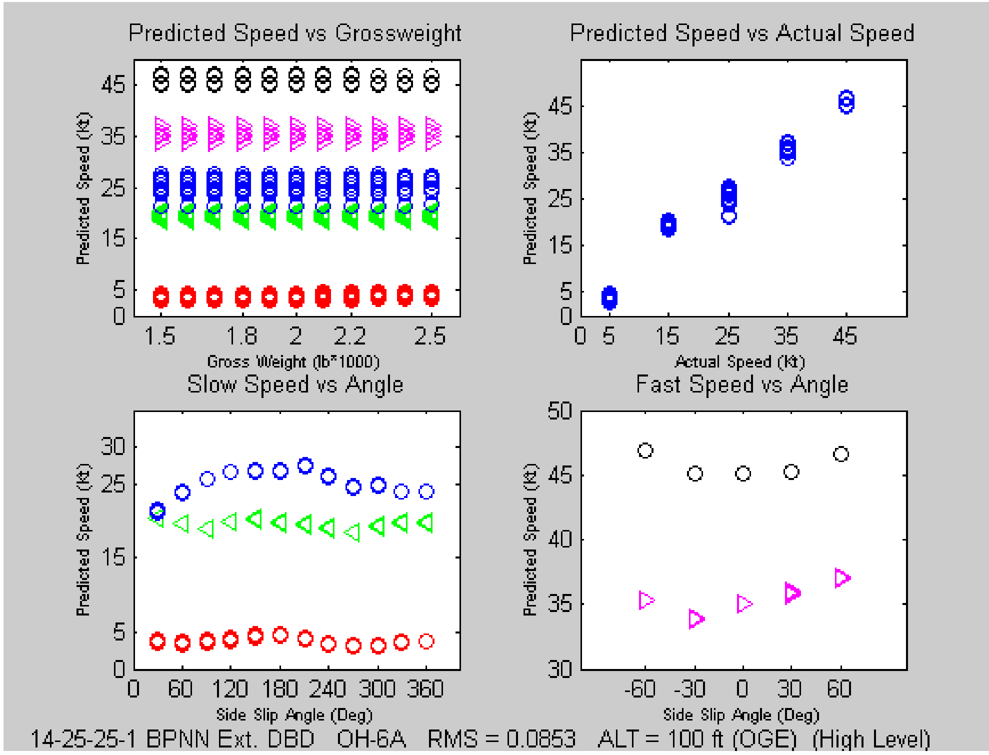


Figure 61. Results for the OH-6A helicopter at 100 ft with baseline data (HL); network configuration 14-25-25-1; Ext. DBD learning rule.

c. 14-25-25-1 Ext. DBD (Pruned) OGE

Figure 62 and Table 57 present the findings for this configuration. Results show that the RMS error is 0.0565, which is slightly greater than that of the NCD setup. The absolute maximum speed error equal to 4.71 knots occurs at 35 knots speed. The maximum airspeed error SD at 1 σ is 1.25 knots (at 35 knots). The overall error SD is 0.8344 knots.

Actual Airspeed (kts)	NEURAL NETWORK RESULTS Total SD = 0.8344			
	Mean of Airspeed (kts)	Airspeed Error at 1 σ (kts)	Percent Error at 1 σ	Abs. Maximum Error (kts)
5	2.9652	0.3962	7.9235%	2.7769
15	16.1362	0.6021	4.0141%	2.1404
25	24.9846	1.1310	4.5241%	2.8046
35	37.5287	1.2565	3.5785%	4.7113
45	46.6852	0.7609	1.6906%	3.0221

Table 57. Results for the OH-6A helicopter at 100 ft with baseline data (HL); network configuration 14-25-25-1; Ext. DBD (Pruned) learning rule.

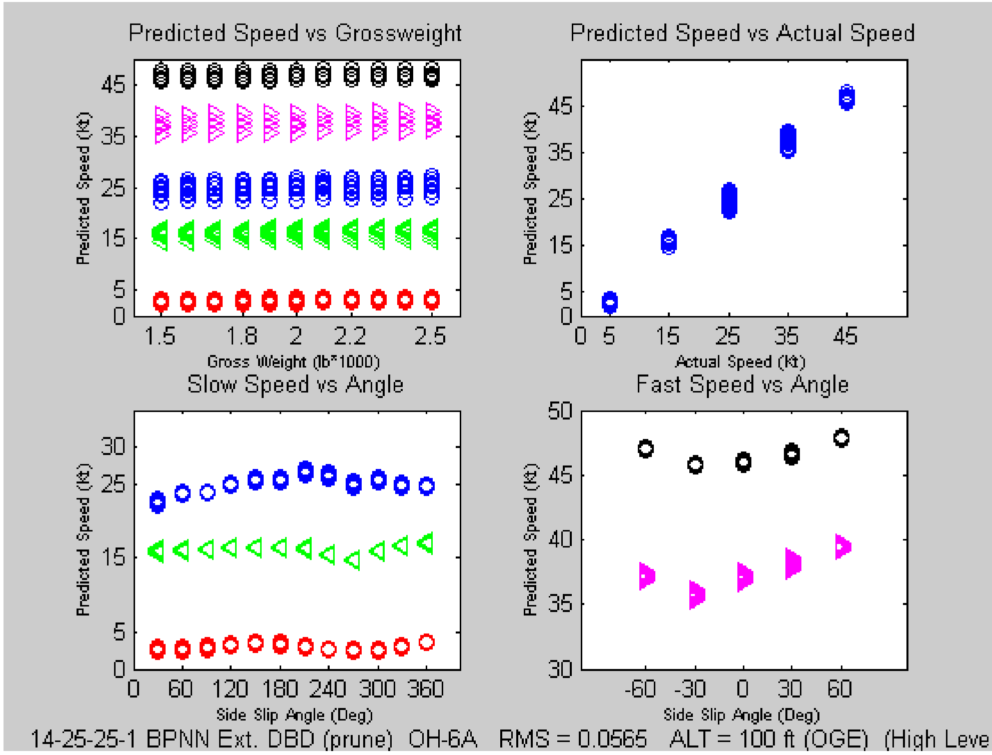


Figure 62. Results for the OH-6A helicopter at 100 ft with baseline data (HL); network configuration 14-25-25-1; Ext. DBD (Pruned) learning rule.

C. NETWORK PERFORMANCES SUMMARY

Results for all experiments are summarized in Table 58 to Table 60. Analysis results for the UH-60A helicopter and OH-6A helicopter at sea level and high level analyses results are shown separately in Tables 58, 59 and 60 in terms of airspeed error SD, absolute maximum error, RMS error and maximum percent error of each NN architecture.

Table 58 displays results obtained for the UH-60A helicopter model. For the UH-60A helicopter at 85 ft (out of ground effect condition) a 2-layer BPNN network with the NCD learning rule with pruning yielded the best results. This configuration shows a predicted airspeed with 0.7 knots error SD, while the maximum error is 3.34 knots, the maximum error percentage error is 4.22 % and the RMS error is 0.0355. When the helicopter is at 20 ft. (in ground effect condition), the network with NCD rule produced an estimate with 0.84 knots error SD, a RMS error equal to 0.0374, a 3 knots maximum error and a maximum error percentage within 5.2 %. These results are significantly better than those obtained by Haas and McCool, with real flight data. Their study showed that using real flight data as input to the NN, UH-60A airspeed can be predicted with an accuracy of $\pm$ 5 knots when the aircraft is in ground effect. However they performed that analysis with a reference airspeed uncertainty of $\pm$ 2 knots. Note that prediction accuracy improved considerably with the simulator data, which has no wind effect nor any other uncertainties.

UH-60A Helicopter BPNN Models	**RMS Error**	**Error of SD at 1 σ**	**Abs. Max. Error (knots)**	**Max. Percent Error (%)**
OGE NCD	0.0658	0.7506	3.85 (at 35 knots)	6.4 (at 5 knots)
OGE Ext. DBD	0.0593	0.7017	3.86 (at 45 knots)	8.49 (at 5 knots)
OGE NCD Prune	0.0355	0.7167	3.34 (at 35 knots)	4.22 (at 5 knots)
Simplified Data OGE NCD	0.0656	1.0032	3.89 (at 35 knots)	5.5 (at 5 knots)
Simplified Data OGE Ext. DBD	0.0565	1.1692	5.4 (at 45 knots)	11.75 (at 5 knots)
Simplified Data NCD Prune	0.0648	1.1942	5.4 (at 25 knots)	7.95 (at 5 knots
One Layer NCD	0.0407	0.8394	3.37 (at 35 knots)	4.5 (at 15 knots)
One Layer Ext. DBD	0.0657	0.7002	3.79 (at 35 knots)	8.34 (at 5 knots)
IGE NCD	0.0374	0.8469	3.0 (at 35 knots)	5.2 (at 35 knots)
IGE Ext. DBD	0.0637	0.8308	4.8 (at 45 knots)	6.0 (at 5 knots)
IGE NCD Prune	0.0724	0.8864	4.86 (15 knots)	5.22 (at 25 knots)
Baseline Data NCD	0.0762	1.5664	6.24 (at 45 knots)	8.0 (at 5 knots)
Baseline Data Ext. DBD	0.0501	0.7320	4.5 (at 35 knots)	10.7 (at 5 knots)
Baseline Data NCD Prune	0.1213	2.0830	8.64 (at 45 knots)	11.5 (at 25 knots)
Baseline Data IGE NCD	0.0759	1.5516	6.24 (at 35 knots)	10.12 (at 5 knots)
Baseline Data IGE Ext. DBD	0.0499	0.6505	4.5 (at 35 knots)	8.7 (at 5 knots)
Baseline Data IGE NCD Prune	0.0901	1.7173	7.91 (at 45 knots)	9.12 (at 25 knots)
Baseline Data OGE NCD	0.0766	1.5516	5.86 (at 25 knots)	9.3 (at 25 knots)
Baseline Data OGE Ext. DBD	0.0482	0.6850	3.8 (at 35 knots)	9.5 (at 5 knots)
Baseline Data OGE NCD Prune	0.1087	1.8642	7.14 (at 35 knots)	9.42 (at 25 knots)

Table 58. Overall Results for the UH-60A Helicopter.

Results showed that the network with the Ext. DBD rule produced the best results for the baseline data analysis. This network predicted airspeed with a 0.73 knots error SD and a 0.05 RMS error. The maximum error was estimated at 4.5 knots at 35 knots speed, while the maximum error percentage was 10.7 % observed for a 5 knots speed. Using the baseline data for the IGE condition airspeed prediction was good with the Ext. DBD network since it produced an error SD of 0.65 knots, a RMS error of 0.05, maximum error of 4.5 knots at the speed of 35 knots and a maximum percent error of 8.7%. The baseline data OGE condition results were very close to those of the baseline data IGE condition results. Using this data, the network with Ext. DBD rule predicted the airspeed with a 0.048 RMS error, a 0.68 knots error SD, a 3.8 knots maximum error at the speed of 35 knots and a 9.5% maximum percent error. The single condition data prediction was slightly better than that observed with the baseline data.

We note that the maximum error and maximum error SD occurred mostly at 35 knots speed for all networks. A potential explanation might be that the hover to forward flight translational lift was set to 30 knots for all the FLIGHTLAB simulator models. The network was trained with the test data set and tested with the training data set in order to explore this idea. Results obtained with the switched data showed that the maximum error and maximum error SD occurs when the helicopter is moving at 30 knots. While not conclusive, this set indicates the difficulty is associated with the simulated helicopter performance near these speeds.

OH-6A analyses were conducted at sea level and at high altitude (at 6000 feet pressure altitude) using a similar methodology for the UH-60A model. Table 59 shows the results of the OH-6A analyses at sea level, and Table 60 shows the results for high altitude analyses.

At sea level, the OGE condition using the single condition data NCD network with the pruning function predicted an airspeed with a 0.76 knots error SD and a 0.067 RMS error. The maximum error was about 4.69 knots for a 35 knots speed while the maximum error was observed at 9.9 %. At sea level, the IGE condition NCD network produced a 0.84 knots error SD. The RMS error was about 0.059, the maximum error was 5.1 knots, and the maximum percent error was 9.7%. Results showed that using baseline data reduced the network performance, as the error SD was 1.169 knots, the RMS error was 0.0611, the maximum error was 5 knots and the maximum percent error was 6.8 %. The NCD network with the pruning function yielded the best results for this condition.

For baseline data, the IGE condition NCD pruned network predicted the airspeed with a 1.15 knots error SD, while the RMS error was about 0.059 and the maximum error was 4.5 for a 35 knots speed. The maximum error percentage was found to be 8.19% for this condition. For the baseline data OGE condition, airspeed was estimated with a 1.20 knots error SD and a 0.065 RMS error with the pruned NCD network. A 5.1 knots maximum error occurred for a 35 knots speed and the maximum error percentage was 6.9%.

OH-6A Helicopter BPNN models (at sea level)	**RMS Error**	**Error of SD at 1σ**	**Abs. Max. Error (knots)**	**Max. Percent Error (%)**
OGE NCD	0.06	0.79	5.2(at 35 knots)	9.5 (at 5knots)
OGE Ext. DBD	0.0735	1.32	5 (at 35 knots)	9.4 (at 5knots)
OGE NCD Prune	0.067	0.759	4.69(at 35 knots)	9.9 (at 5knots)
IGE NCD	0.059	0.8465	5.1(at 35 knots)	9.7 (at 5knots)
IGE Ext. DBD	0.062	0.9678	5.9(at 35 knots)	12.2 (at 5knots)
IGE NCD Prune	0.0674	0.88	4.41(at 35 knots)	12.1(at 5knots)
Baseline Data NCD	0.062	1.192	4.82(at 35 knots)	8.69(at 5knots)
Baseline Data Ext. DBD	0.0687	1.407	7.44(at 35 knots)	14(at 5knots)
Baseline Data NCD prune	0.0611	1.16	5(at 35 knots)	6.8(at 5knots)
Baseline Data IGE NCD	0.0593	1.15	4.49(at 35 knots)	9.29(at 5knots)
Baseline Data IGE Ext. DBD	0.0686	1.32	7.44(at 35 knots)	11.9(at 5knots)
Baseline Data IGE NCD Prune	0.0592	1.15	4.51(at 35 knots)	8.19(at 5knots)
Baseline Data OGE NCD	0.0671	1.20	4.82(at 35 knots)	7.2(at 5knots)
Baseline Data OGE Ext. DBD	0.0687	1.3	6.6(at 35 knots)	9.4(at 5knots)
Baseline Data OGE NCD Prune	0.0650	1.2	5.1(at 35 knots)	6.9(at 5knots)

Table 59. Overall Results for OH-6A Helicopter at Sea Level.

The pruned NCD network predicted the airspeed with a 0.64 knots error SD for the OH-6A high level OGE condition. The RMS error was found to be equal to 0.0475, while the maximum error was 2.8 knots at 5 knots speed and the maximum error was about 7%. Finally, the airspeed was predicted with a 0.64 knots error SD, a 0.0433 RMS error, a 2.58 knots maximum error and a 5.6% maximum error percentage for the IGE condition using the single condition data with the pruned NCD network.

The following results were obtained for the baseline data. The Ext. DBD network performed best with a 0.65 knots error SD, a 0.055 RMS error, a 3.4 knots maximum error for a 5 knots speed and a 7.5 % maximum error percentage. The NCD pruned network yielded the best results for the baseline data IGE condition. For this setup, the RMS error was 0.0352 while the error SD was 0.66 and the maximum error was 2 knots at the speed of 25 knots. The maximum percent error was 7.1% for this condition. The baseline data OGE condition results were close to the IGE condition results. The NCD network estimated the airspeed with an error SD of 0.74 knots, a RMS error of 0.056, a maximum error of 3.15 knots and a maximum percent error of 8.4%.

OH-6A Helicopter BPNN models (at high altitude)	**RMS Error**	**Error of SD at 1σ**	**Abs. Max. Error (knots)**	**Max. Percent Error (%)**
OGE NCD	0.0485	0.6637	2.47 (at 5knots)	9.47(at 5knots)
OGE Ext. DBD	0.0958	1.022	6.36(at 15knots)	5.67(at 5knots)
OGE NCD Prune	0.0475	0.64	2.85(at 5knots)	7.1(at 5knots)
IGE NCD	0.0516	1.20	2.59(at 5knots)	9.12(at 5knots)
IGE Ext. DBD	0.0865	0.95	5.8(at 15knots)	5.53(at 5knots)
IGE NCD Prune	0.0443	0.64	2.58(at 5knots)	5.59(at 5knots)
Baseline Data NCD	0.05	0.71	3.15(at 25knots)	8.73(at 5knots)
Baseline Data Ext. DBD	0.077	1.02	5.4(at 15knots)	8.8(at 5knots)
Baseline Data Ext. DBD prune	0.055	0.65	3.43(at 5knots)	7.5(at 5knots)
Baseline Data IGE NCD	0.0446	0.63	2.49(at 5knots)	8.84(at 5knots)
Baseline Data IGE Ext. DBD	0.0687	0.9	4.55(at 5knots)	8.44(at 5knots)
Baseline Data IGE NCD Prune	0.0352	0.66	2.09(at 25knots)	7.1(at 5knots)
Baseline Data OGE NCD	0.056	0.74	3.15(at 25knots)	8.45(at 5knots)
Baseline Data OGE Ext. DBD	0.0853	1.02	5.39(at 15knots)	9(at 5knots)
Baseline Data OGE Ext. DBD Prune	0.0565	0.83	4.71(at 35knots)	7.9(at 5knots)

Table 60. Overall Results of OH-6A Helicopter at High Altitude.

In summary, these results show that the NN approach to predict airspeed using simulation data is quite promising. The BPNN with two hidden layers and 25 PEs in each layer performs the best among all studied architectures. We note that different learning rules yielded different results and that enabling the pruning facility improved the network performance in most cases.

V. CONCLUSIONS AND RECOMMENDATIONS

A. SUMMARY

Today, military helicopters perform a wide variety of tasks in conditions ranging from hot and dry to cold and wet, windy and low visibility weather. Accurate low speed velocity sensing devices are essential because aircraft velocity and position information are what pilots need to perform safely in these regimes. However, conventional speed measuring systems do not work accurately when the aircraft speed is below 40 knots.

NN-based airspeed prediction studies developed by McCool, Haas and others showed that NN is a good approach to solve this problem. The objective of this thesis is to build a NN model in order to predict OH-6A helicopter airspeed in the low speed environment using a flight simulator to obtain the parameters required for the NN. In our study the NN model is developed with Neural Works Professional Plus/II software.

First a NN model of the UH-60A helicopter was built and implemented, and several NN configurations analyzed. The reason for building a NN model for the UH-60A helicopter was to lay out a background to make a comparison of NN predictions using simulator data and NN predictions using real flight data. The results showed using simulator data potentially improves the accuracy of prediction significantly.

Three different methods were investigated to select the NN training data. The first one is a single condition data set in which the data belongs to one altitude only. The second one is called the baseline data set, and is formed by combining the data of two single condition data sets of different altitudes. The third set is obtained by applying principal component analyses to decrease the input space dimension and it is called a simplified data set. Results showed that the network trained by using a single condition data set proved to be the most successful and performance degraded only slightly with the baseline data. Moreover, the BPNN network produced more successful predictions than the RBFN implementation.

Among all BPNN architecture types considered, a two-hidden layer BPNN with an enabled pruning facility for the NCD learning rule showed the best performance. At sea level pressure altitude, the UH-60A low airspeed was predicted with one-sigma accuracy of ± 0.71 knots when the aircraft was out of ground effect. The accuracy of prediction was ± 0.88 knots when the aircraft was in ground effect.

The OH-6A low speed was predicted using a similar methodology based on the results obtained from the UH-60A model. The OH-6A analysis was performed at high pressure altitude as well as at seal level altitude. Results showed that at sea level, the OH-6A airspeed could be predicted with one-sigma accuracy of ± 0.75 knots when the aircraft is out of ground effect. The best performance for all high altitude analyses obtained from the pruned NCD network was with single condition data. For IGE conditions, the prediction accuracy was about ± 0.88 knots. High altitude analysis was performed at 6000 feet. The results showed that at high altitude, the OH-6A airspeed could be predicted with an accuracy of ± 0.64 knots when the aircraft is out of ground effect. For IGE conditions, the prediction accuracy was about ± 0.64 knots.

This study showed that a NN based approach to determine OH-6A helicopter airspeeds using a flight simulator is quite promising. The approach considered presents a mechanically simple alternative to current low airspeed measurement systems, and as a result contributes to increase the flight safety and combat effectiveness.

B. RECOMMENDATIONS FOR FURTHER RESEARCH

Several avenues are available to extend the work presented in this study. First, results obtained using real flight data are needed. These results should be compared to the results obtained here to evaluate the performance of our approach. Second, more sophisticated maneuvers need to be simulated and analyzed. The NN model might be further improved by using different network models, model parameters, etc. Another type of network to be investigated may be NN-based sideslip angle estimator, which would provide accurate wind direction for pilot-aid systems. Finally, the effects of measurement noise, sensor errors, and sensor failure should be investigated.

LIST OF REFERENCES

1. *Neural Network Design, M. Hagan, H., Demuth, M. Beale, PWS Publishing Company, 1996*

2. *Neural Network Toolbox For Use with Matlab, The Math Works, Inc., 2001*

3. "*A NN Based Approach To Helicopter Low Airspeed And Sideslip Angle Estimation," K.M. McCool, D.J. Haas, Proceedings of the AIAA Flight Simulation Technologies Conference, 1996*

4. "*A NN Based Approach To Helicopter Low Airspeed And Sideslip Angle Estimation," K.M. McCool, D.J. Haas, Proceedings of the AIAA Flight Simulation Technologies Conference, 1996*

5. *Using Neural Works, Neural Ware, Inc., 1993*

6. *Neural Computing, Neural Ware, Inc., 1993*

7. *FLIGHTLAB Manual Vol. II, Scope, Gscope, PilotStation, CSGE, Model Editor, Xanalysis*, *Advance Rotorcraft Technology, June 2001*

8. "*Flight Investigations of a Helicopter Low Airspeed Estimation System Based on Measurement of Control Parameters," A. J. Faulkner, F. Buchner, Sixth European Rotorcraft and Powered Lift Aircraft Forum, Sep. 1980.*

9. *Neural Networks: Tricks of the Trade, K-R. Muller (Editor), G. Orr (Editor), 1st Edition, May 1999*

10. *Artificial Neural Networks, D. Klerfors, St.Louis Uni., 1998* (http://www.hj.se/~de96klda/NeuralNetworks.htm), last accessed December 2001.

11. *Neural Network FAQ, Warren S. Sarle, 2001* (ftp://ftp.sas.com/pub/neural/FAQ2.html), last accessed January 2002.

12. "*The Measurement of Helicopter Air Data Using a Swivelling Pitot-Static Pressure Probe," J. Carter, Air Data Symposium, Monterey, CA, June 1976*

13. "*Omni-directional Low Range Airspeed, Display and System Requirements for Helicopters and V/STOL Aircraft," D. L. Green, Air Data Symposium, Monterey, CA, June 1976*

14. *FLIGHTLAB Manual Vol. I, Advance Rotorcraft Technology, April 2001*

15. "*An Electro-Optic Airspeed Sensor," G. A. Dubro, D. G. Kim, M. J. Rudd, Air Data Symposium, Monterey, CA, June 1976*

16. "*Practical NN Applications in the Mining Industry," L. Miller-Tait, R. Pakalnis, Proceedings of the Second International Conference, Vol.2, 1999*

17. *Helicopter Flight Dynamics: The Theory and application of Flying Qualities and Simulation Modeling, G. D. Padfield, AAIA Education Series, 1999*

18. *Linear Algebra and Its Applications, Third Edition, G. Strang, Saunders HBJ Publishing, 1988*

19. *Modeling the OH-6A Using FLIGHTLAB and Helicopter Simulator Considerations, Master's Thesis, Naval Postgraduate School, G. A. Ouellette, to be completed March 2002*

20. *"Predicting Airspeed and Sideslip Angle using an Artificial Neural Network", D. Goff, S. Thomas, P. Jones, C. Massey, AHS 55th Annual Forum, Montreal, Canada, 1999*

APPENDIX A. NEURALWORKS PROFESSIONAL PLUS/II PROGRAM SETUP

Neural Works Professional II/PLUS is a powerful and flexible development tool. It has over 20 different algorithms including common network types such as Back Propagation, Kohonen and Radical Basis Functions. The product includes a variety of diagnostic tools as well as options for quick and easy building networks.

Before creating the network, data files must be put under the directory of Neuralware Professional directory. The test and training input data files must be given different names but both of them must have extension of ".nna". The data files used in this thesis may be obtained from Professor R. W. Duren, Naval Post Graduate School Department of Aerospace and Aeronautical Engineering, by request.

Neural network development process begins with collecting and preprocessing input data. In this study data is collected from FLIGHTLAB simulation and prepared by using MATLAB. Preparation involves encoding data to a format that NN can deal with. For this problem data set was converted to a matrix with a dimension of 1114x14, where columns represent the inputs and rows represent the samples of each inputs. MATLAB is also used to separate the data into to sets, training set and test set, as well as to display the results. MATLAB codes for preprocessing and post processing the data are included in Appendix B.

After starting NeuralWorks program, Back-propagation command, under the InstaNet menu, must be selected to create a BP network. Selecting this command opens up BP dialog box, which allows user to build the main frame of the network by entering number of layers and number of PEs per layer. In addition to these, learning coefficient of each layer, momentum term, learning rule, transfer function, test data and training data of the network can be selected. Also by selecting "minmax" radio button, inputs can be mapped from the data file to a desired range, such as –1 to 1.

Activating this box opens up another window, which asks user how to display the performance of the network, such as RMS error, classification rate, etc. After these selections network is created which is presented in a window like in Figure 5. Created network can be saved either in binary format or in ASCII format by selecting the Save command under the File menu.

Selecting the Learn command under the Run menu starts training. At this point, number of learning iterations can be entered by the user. As each training example is presented to the network, the network produces an output, which is used to evaluate the training performance of the network. Another way to train the network is to use Savebest command under the Run menu. This commands opens up Run/Check dialog box, which makes pruning facility accessible. Based on a decision criteria specified by the user, pruning facility disables connections in a network as the network is training.

Network is tested using the test data sets and by selecting the Test command under the Run menu. The desired outputs, along with the actual network results are written to the results file which has ".nnr" extension.

Based on the above explanations and after preparing the data, OH-6A helicopter 14-25-25-1 BPNN NCD (Prune) network model created using the following steps:

Start Neural Works on the computer

Select Back-Propagation command from the InstaNet menu. This command pops up the following window.

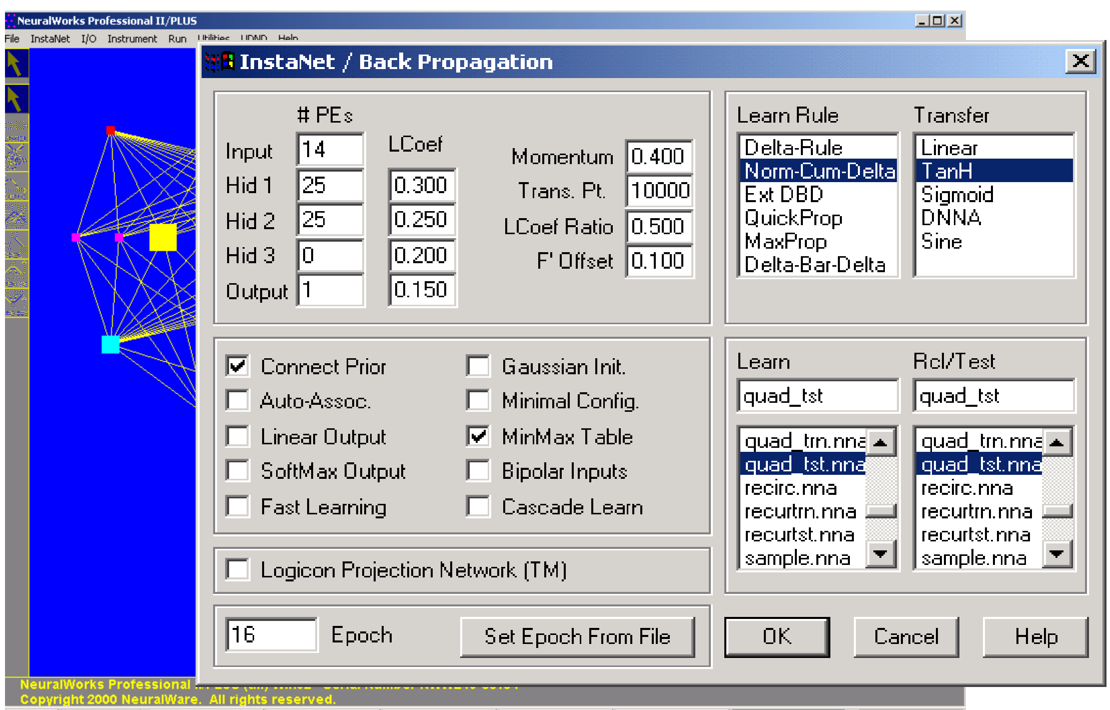


Figure 63. Back-propagation network setup window

1. Select “trainfile_trn” from the training input scroll window.
2. Select “testfile_tst” from the testing input scroll window.
3. In the number of PEs section, enter the following numbers:

   -Input: 14

   -Hid1: 25

   -Hid2: 25

   -Output: 1
4. Enter 0.4 for momentum and 0.5 for LCoef ratio.
5. Select Norm-Cum-Delta for the learning rule.
6. Select TanH for the transfer function.
7. Check MinMax table box.
8. Click the OK button.
9. After clicking OK the following window opens automatically. Select RMS Error.

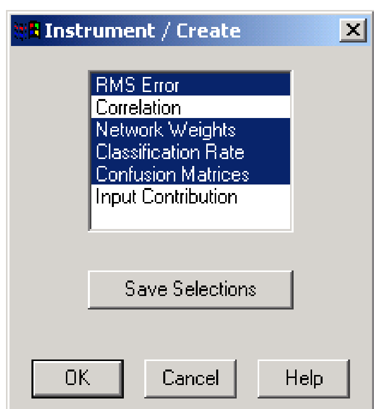


Figure 64. Instrument /Create menu

10. Click OK.
11. Start training by selecting the SaveBest command under the Run menu. SaveBest command setup is shown in Figure #. Training can also be started by selecting Learn command under the same menu. SaveBest command allows user to use Pruning facility.
12. Type the name of the file that results are to be written.
13. Enter 50000 in the For field.
14. Enter 1000 for the Test Interval.

15. Check the pruning radio button.

16. Enter 0.975 for the Tolerance field.

17. Select Classification Rate for the Objective Function list.

18. Click OK. to start training.

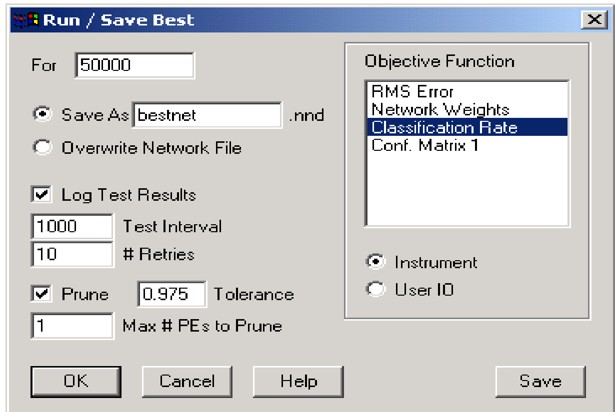


Figure 65. SaveBest command window

19. After training is completed select Test in the Run menu.

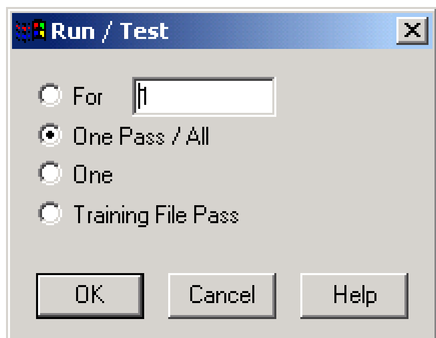


Figure 66. Test command window

20. Select One Pass/All.

21. Click OK.

After test process has been completed the performance of the network is displayed on the main window in terms of selected options, such as RMS error, network weights, etc. and the results are stored in the test file with .nnr extension. In this work, the result files were saved as text files in order to be exported into the MATLAB.

APPENDIX B. MATLAB® M-FILES

MATLAB m-files were developed to prepare the simulation outputs for NN and to process the NN results.

a. FILE Name: Gregcode

The first code included in this appendix is the FLIGHLAB scope language, an interpretive language that uses the industry standard MATLAB syntax. This routine loads the specified model and runs the simulator. The outputs of this routine are used as inputs to the NN model.

```
/*********************************************************************
        This code was developed by LT. Gregory OUELLETTE, USN. for the Naval
Postgraduate School in partial fulfillment of the requirements for a Masters Degree in
Aeronautical Engineering.
        September 2001.
/*********************************************************************

        exec("$FL_DIR/flme/models/articulated/arti-rgd-3iv-qs-simeng.def",1)
        world_model_airframe_cpg_testcond_poszic = -85;
        exec("xatestcond.exc",1)
        exec("xamodeltrim.exc",1)

        goto world
        group test

        gw = [16000:1000:24000]';
        hdg = [30:30:360]';
        vel = [0:5:30]';
        column = 0;

        utrim = @trimvariable;
        statesave = savestates(world_topsolve);

        for ngw = 1:prod(size(gw))
          for nhdg = 1: prod(size(hdg))
            for nvel = 1: prod(size(vel))

        world_model_airframe_cpg_testcond_poszic = -85;
        world_model_airframe_cpg_testcond_veq = vel(nvel);
        world_model_airframe_cpg_testcond_gamh = hdg(nhdg);
```

```
world_model_data_vweight = gw(ngw);

exec("xatestcond.exc",1)
exec("xamodeltrim.exc",1)

outputs(column+nvel,1) = world_model_control_data_xatrm;
outputs(column+nvel,2) = world_model_control_data_xbtrm;
outputs(column+nvel,3) = world_model_control_data_xctrm;
outputs(column+nvel,4) = world_model_control_data_xptrm;
outputs(column+nvel,5) = world_model_airframe_cpg_xaout_p;
outputs(column+nvel,6) = world_model_airframe_cpg_xaout_q;
outputs(column+nvel,7) = world_model_airframe_cpg_xaout_r;
outputs(column+nvel,8) = world_model_airframe_cpg_xaout_phi;
outputs(column+nvel,9) = world_model_airframe_cpg_xaout_psi;
outputs(column+nvel,10) = world_model_airframe_cpg_xaout_radralt;
outputs(column+nvel,11) = world_model_airframe_cpg_xaout_vclimb;
outputs(column+nvel,12) = world_model_rotor1_rotor_cpg_xaout_omega;
outputs(column+nvel,13) = world_model_propulsion_cpg_xaout_etorq;
outputs(column+nvel,14) = world_model_data_vweight;
outputs(column+nvel,15) = world_model_airframe_cpg_xaout_tas;
outputs(column+nvel,16) = world_model_airframe_cpg_testcond_gamh;
outputs(column+nvel,17) = world_model_airframe_cpg_testcond_veq;
outputs(column+nvel,18) = 0;

end

column = column + prod(size(vel));
savestates(world_topsolve,statesave);
@trimvariable = utrim;

end
end

trainset = outputs';
save("velhead.sav",trainset)


exec("$FL_DIR/flme/models/articulated/arti-rgd-3iv-qs-simeng.def",1)
world_model_airframe_cpg_testcond_poszic = -85;
exec("xatestcond.exc",1)
exec("xamodeltrim.exc",1)

goto world
group test
```

```
gw = [16000:1000:24000]';
hdg = [-60:30:60]';
vel = [35:5:40]';
column = 0;

utrim = @trimvariable;
statesave = savestates(world_topsolve);

for ngw = 1:prod(size(gw))
  for nhdg = 1: prod(size(hdg))
    for nvel = 1: prod(size(vel))

world_model_airframe_cpg_testcond_poszic = -85;
world_model_airframe_cpg_testcond_veq = vel(nvel);
world_model_airframe_cpg_testcond_gamh = hdg(nhdg);
world_model_data_vweight = gw(ngw);

exec("xatestcond.exc",1)
exec("xamodeltrim.exc",1)

outputs(column+nvel,1) = world_model_control_data_xatrm;
outputs(column+nvel,2) = world_model_control_data_xbtrm;
outputs(column+nvel,3) = world_model_control_data_xctrm;
outputs(column+nvel,4) = world_model_control_data_xptrm;
outputs(column+nvel,5) = world_model_airframe_cpg_xaout_p;
outputs(column+nvel,6) = world_model_airframe_cpg_xaout_q;
outputs(column+nvel,7) = world_model_airframe_cpg_xaout_r;
outputs(column+nvel,8) = world_model_airframe_cpg_xaout_phi;
outputs(column+nvel,9) = world_model_airframe_cpg_xaout_psi;
outputs(column+nvel,10) = world_model_airframe_cpg_xaout_radralt;
outputs(column+nvel,11) = world_model_airframe_cpg_xaout_vclimb;
outputs(column+nvel,12) = world_model_rotor1_rotor_cpg_xaout_omega;
outputs(column+nvel,13) = world_model_propulsion_cpg_xaout_etorq;
outputs(column+nvel,14) = world_model_data_vweight;
outputs(column+nvel,15) = world_model_airframe_cpg_xaout_tas;
outputs(column+nvel,16) = world_model_airframe_cpg_testcond_gamh;
outputs(column+nvel,17) = world_model_airframe_cpg_testcond_veq;
outputs(column+nvel,18) = 0;
end

column = column + prod(size(vel));
savestates(world_topsolve,statesave);
@trimvariable = utrim;

end
end
```

```
trainset = outputs';
save("velheadfast.sav",trainset)
```

b. FILE Name: Ozcan.m

```
%*****************************************************************
%  This routine takes simulation output files as input and makes them applicable for NN
%  model by forming a matrix, where columns represent the input parameters to NN and
%  rows represent the samples.
%  mat1 is the output matrix of the routine where all the data are stored and to be used as
%  input data to the NN.
%  The last part of the routine may be used to obtain the simplified data using
%  eigenvalues and  eigenvectors. Dataeig is the resultant matrix for simplified data.
%  September 2001
%*****************************************************************

clear all;
format short e
load velhead.txt;                  % loading ascii file (simulation output file)
load velheadfast.txt;              %loading ascii file

m=1;

for i = 1:length(velhead)/6

slat_stick(i)  =   velhead(m,1);
slong_stick(i) =   velhead(m,2);
scoll_pos(i)   =   velhead(m,3);
sped_pos(i)    =   velhead(m+1,1);
sroll_rate(i)  =   velhead(m+1,2);
spitch_rate(i) =   velhead(m+1,3);
syaw_rate(i)   =   velhead(m+2,1);
spitch_att(i)  =   velhead(m+2,2);
sroll_att(i)   =   velhead(m+2,3);
salt(i)        =   velhead(m+3,1);
sclimb_rate(i) =   velhead(m+3,2);
smrb_rpm(i)    =   velhead(m+3,3);
seng_torque(i) =   velhead(m+4,1);
sgw(i)         =   velhead(m+4,2);
stas_trim(i)   =   velhead(m+4,3)*(360/608);      % ft/sec is converted into knots
sheading(i)    =   velhead(m+5,1);
stas_trgt(i)   =   velhead(m+5,2);
```

```
szero(i)       =   velhead(m+5,3);
m=m+6;
end
m=1;
for i = 1:length(velheadfast)/6

flat_stick(i)  =   velheadfast(m,1);
flong_stick(i) =   velheadfast(m,2);
fcoll_pos(i)   =   velheadfast(m,3);
fped_pos(i)    =   velheadfast(m+1,1);
froll_rate(i)  =   velheadfast(m+1,2);
fpitch_rate(i) =   velheadfast(m+1,3);
fyaw_rate(i)   =   velheadfast(m+2,1);
fpitch_att(i)  =   velheadfast(m+2,2);
froll_att(i)   =   velheadfast(m+2,3);
falt(i)        =   velheadfast(m+3,1);
fclimb_rate(i) =   velheadfast(m+3,2);
fmrb_rpm(i)    =   velheadfast(m+3,3);
feng_torque(i) =   velheadfast(m+4,1);
fgw(i)         =   velheadfast(m+4,2);
ftas_trim(i)   =   velheadfast(m+4,3)*(360/608);     % ft/sec is converted into knots
fheading(i)    =   velheadfast(m+5,1);
ftas_trgt(i)   =   velheadfast(m+5,2);
fzero(i)       =   velheadfast(m+5,3);
m=m+6;
end
lat_stick=[slat_stick,flat_stick];
long_stick=[slong_stick,flong_stick];
coll_pos=[scoll_pos,fcoll_pos];
ped_pos=[sped_pos,fped_pos];
roll_rate=[sroll_rate,froll_rate];
pitch_rate=[spitch_rate,fpitch_rate];
yaw_rate=[syaw_rate,fyaw_rate];
pitch_att=[spitch_att,fpitch_att];
roll_att=[sroll_att,froll_att];
alt=[salt,falt];
climb_rate=[sclimb_rate,fclimb_rate];
mrb_rpm=[smrb_rpm,fmrb_rpm];
eng_torque=[seng_torque,feng_torque];
gw=[sgw,fgw];
tas_trim=[stas_trim,ftas_trim];
heading=[sheading,fheading];
tas_trgt=[stas_trgt,ftas_trgt];

%***for UH-60A helicopter add parameter "eng_torque" to the matrix below****
```

```
mat1=[lat_stick;long_stick;coll_pos;ped_pos;roll_rate;pitch_rate;yaw_rate;pitch_
att;roll_att;alt;climb_rate;mrb_rpm;gw;heading;tas_trgt]'

% To simplify the data by eigenvalues and eigenvectors use the following part
    % mat2=mat1'*mat1;
    % [v,d]=eig(mat2);
    % After examining the eigenvalues create submatrix u
    % u=[mat1(:,9),mat1(:,10),mat1(:,11),mat1(:,12),mat1(:,13),mat1(:,14)];
    % dataeig=mat1'*u;
```

c. FILE Name: Train.m

```
%*****************************************************************
% This routine takes the output matrix of Ozcan.m code, mat1 or dataeig, by loading the
% ascii file and filters it so that train data set can be obtained from the whole data.
% September 2001
%*****************************************************************

    clear all;
    format short e
    load oh6sl_100ft.txt;            % loading ascii file

    [rmax,cmax]=size(oh6sl_100ft);
    dmax=924  ;   % Enter the # of rows  associated with slow velocity data
    s=1;
    x=0;
    y=8;
    for d=1:dmax/14
        for i=1:2:7
            trndt(s,:)=oh6sl_100ft(i+x,:);
            s=s+1;
        end
        for j=0:2:6
          trndt(s,:)=oh6sl_100ft(j+y,:);
          s=s+1;
        end
        x=x+14;
        y=y+14;
    end
    for d1=(dmax+2):2:rmax
        trndt(s,:)=oh6sl_100ft(d1,:);
        s=s+1;
    end
    trndt   % training data set
```

d. FILE Name: Test.m

```
%*******************************************************************
% This routine takes the output matrix of Ozcan.m code, mat1 or dataeig, by loading the
% ascii file and filters it so that test data set can be obtained from the whole data.
% September 2001
%*******************************************************************

      clear all;
      format short e
      load oh6sl_100ft.txt;                        % loading ascii file

      [rmax,cmax]=size(oh6sl_100ft);
      dmax=924  ;            %   Enter the #of rows  associated with  slow velocity data
      s1=1;
      x1=0;
      y1=9;
      for d2=1:dmax/14
         for l=2:2:6
            tstdt(s1,:)=oh6sl_100ft(l+x1,:);
            s1=s1+1;
         end
         for j1=0:2:4
            tstdt(s1,:)=oh6sl_100ft(j1+y1,:);
            s1=s1+1;
         end
         x1=x1+14;
         y1=y1+14;
      end
      for d3=(dmax+1):2:rmax
         tstdt(s1,:)=oh6sl_100ft(d3,:);
         s1=s1+1;
      end
      tstdt                   %   test data set
```

e. FILE Name: Bersan.m

```
%*****************************************************************
% This program used to process the outputs of the NN. In this routine the output file of
% the NN is taken as input and vectors of NN predicted speeds are created related to each
% gross weight and sideslip angle of the helicopter. This file also produces the figures
 % and the evaluation of the NN results.
% October 2001
% Note: For baseline data and single data use the specified sections of the program. Also
% depending on the helicopter type activate and deactivate the stated lines of the code.
%*****************************************************************

        clear all;
        load oh6_sl_ncdprune.txt;          % loading ascii file
        a=oh6_sl_ncdprune(:,2);            % NN outputs(predicted speeds)
        ilk=oh6_sl_ncdprune(:,1);          % target values(actual speeds)

        %set initial variables
        g=1;
        j=1;
        k=1;
        l=1;
        m=1;

        n=length(a);
        aci=[30:30:360];                   % angles for slow speed
        aci1=[-60:30:60];                  % angles for fast speed
        ai=length(aci);

        %  x=1.6:0.1:2.4;                  %  gross weight range of uh-60a helicopter(10^4 lb)
        x=1.500:.100:2.550;                % gross weight range of oh-6a helicopter(10^3 lb)
        %(for UH-60A:16000:1000:24000  - for OH-6A: 1500:100:2550 )

        gw=length(x);
        na=(ai*gw*6);                      % slow velocity vector dimension (for baseline data
%  multiply by 6, for single data multiply by 3)

% Classify the speeds and get each speed vectors
        for i=1:3:na
          a1(j,1)=a(i,1);                  %vector of 5 kts
          fark5(j)= abs(a1(j,1)-ilk(i,1));
          j=j+1;
        end
```

```
for i=2:3:na
  a2(k,1)=a(i,1);                    %vector 15 kts
  fark15(k)= abs(a2(k,1)-ilk(i,1));
  k=k+1;
end

for i=3:3:na
  a3(l,1)=a(i,1);                    %vector 25 kts
  fark25(l)= abs(a3(l,1)-ilk(i,1));
  l=l+1;
end

for t=(na+1):2:n
  a4(m,1)=a(t,1);                    % vector 35 kts
  fark35(m)= abs(a4(m,1)-ilk(t,1));
  m=m+1;
end
 for t=(na+2):2:n
   a5(g,1)=a(t,1);                   %vector 45 kts
   fark45(g)= abs(a5(g,1)-ilk(t,1));
   g=g+1;
 end

% classify speeds according to gross weights

   y=length(a1);
   b=y/gw;

 % for speed = 5 kts
   c=1;
   for i=1:gw
     if i>=2
       z=1;
       for j=(((i-1)*b)+1):i*b
         a11(z,i)=a1(j,1);
         z=z+1;
       end
     else
       for v=1:b
         a11(c,1)=a1(v,1);
         c=c+1;
       end
     end
   end
    figure(1)
```

```
          subplot(2,2,1);plot(x,a11,'ro')

 %for speed = 15 kts
          c=1;
          for i=1:gw
            if i>=2
              z=1;
              for j=(((i-1)*b)+1):i*b
                a22(z,i)=a2(j,1);
                z=z+1;
              end
            else
              for v=1:b
                a22(c,1)=a2(v,1);
                c=c+1;
             end
            end
          end

          hold on
          subplot(2,2,1);plot(x,a22,'g<')

 %for speed = 25 kts
          c=1;
          for i=1:gw
            if i>=2
              z=1;
              for j=(((i-1)*b)+1):i*b
                a33(z,i)=a3(j,1);
                z=z+1;
              end
            else
              for v=1:b
                a33(c,1)=a3(v,1);
                c=c+1;
              end
            end
          end
          hold on
          subplot(2,2,1);plot(x,a33,'bo')

% for airspeed = 35 kts
          y1=length(a4);
          b1=y1/gw;
          c=1;
          for i=1:gw
```

```
      if i>=2
        z=1;
        for j=(((i-1)*b1)+1):i*b1
          a44(z,i)=a4(j,1);
          z=z+1;
        end
      else
        for v=1:b1
          a44(c,1)=a4(v,1);
          c=c+1;
        end
      end
    end
    hold on
    subplot(2,2,1);plot(x,a44,'m>')

% for airspeed 45 kts
    c=1;
    for i=1:gw
      if i>=2
        z=1;
        for j=(((i-1)*b1)+1):i*b1
          a55(z,i)=a5(j,1);
          z=z+1;
        end
      else
        for v=1:b1
          a55(c,1)=a5(v,1);
          c=c+1;
        end
      end
    end
    hold on
    subplot(2,2,1);plot(x,a55,'ko')
    axis([1.4 2.6 0 50])
    title('Predicted Speed vs Grossweight','FontSize',10)
    xlabel('Gross Weight (lb*10000)','FontSize',7)                     %for UH-60A
    % xlabel('Gross Weight (lb*1000)','FontSize',7)                    % for OH-6A
    ylabel('Predicted Speed (Kt)','FontSize',7)
    set(gca,'xtick',[1.6 1.8 2.0 2.2 2.4],'ytick',[0 5 15 25 35 45])       %for UH-60A
    % set(gca,'xtick',[1.5 1.8 2.0 2.2 2.5],'ytick',[0 5 15 25 35 45])     % for OH-6A
    hold off

% Plotting predicted speed vs actual speed

    subplot(2,2,2);plot(ilk,a,'o')
```

```
        axis([0 55 0 55])
        title('Predicted Speed vs Actual Speed','FontSize',10)
        xlabel('Actual Speed (Kt)','FontSize',7)
        ylabel('Predicted Speed (Kt)','FontSize',7)
        set(gca,'xtick',[0 5 15 25 35 45],'ytick',[0 5 15 25 35 45])
        hold off
 %  plotting airspeed vs. angle (for single data only)

      %    subplot(2,2,3);plot(aci,a11,'ro')
      %    hold on
      %    subplot(2,2,3);plot(aci,a22,'g<')
      %    hold on
      %    subplot(2,2,3);plot(aci,a33,'bo')
      %    axis([0 400 0 35])
      %    title('Slow Speed vs Angle','FontSize',10)
      %    xlabel('Side Slip Angle (Deg)','FontSize',7)
      %    ylabel('Predicted Speed (Kt)','FontSize',7)
      %    set(gca,'xtick',[0 60 120 180 240 300 360],'ytick',[0 5 15 25 30])
      %    hold off
      %    subplot(2,2,4);plot(aci1,a44,'m>')
      %    hold on
      %    subplot(2,2,4);plot(aci1,a55,'ko')
      %    axis([-100 100 30 50])
      %    title('Fast Speed vs Angle','FontSize',10)
      %    xlabel('Side Slip Angle (Deg)','FontSize',7)
      %    ylabel('Predicted Speed (Kt)','FontSize',7)
      %    set(gca,'xtick',[-60 -30 0 30 60],'ytick',[30 35 40 45 50])
      %    hold off

%   For Baseline data plotting speed vs. angle use the following section of the routine

        for df=1:12
           a1m1(df,:)=a11(df,:);
           a2m1(df,:)=a22(df,:);
           a3m1(df,:)=a33(df,:);
        end
        for vc=1:5
           a4m1(vc,:)=a44(vc,:);
           a5m1(vc,:)=a55(vc,:);
        end
        nd=1;
        for gf=13:24
           a1m2(nd,:)=a11(gf,:);
           a2m2(nd,:)=a22(gf,:);
           a3m2(nd,:)=a33(gf,:);
           nd=nd+1;
```

```
            end
            nd1=1;
            for gf1=6:10
               a4m2(nd1,:)=a44(gf1,:);
               a5m2(nd1,:)=a55(gf1,:);
              nd1=nd1+1;
            end
            subplot(2,2,3);plot(aci,a1m1,'ro')
            hold on
            subplot(2,2,3);plot(aci,a1m2,'ro')
            hold on
            subplot(2,2,3);plot(aci,a2m1,'g<')
            hold on
            subplot(2,2,3);plot(aci,a2m2,'g<')
            hold on
            subplot(2,2,3);plot(aci,a3m1,'bo')
            hold on
            subplot(2,2,3);plot(aci,a3m2,'bo')
            axis([0 400 0 35])
            title('Slow Speed vs Angle','FontSize',10)
            xlabel('Side Slip Angle (Deg)','FontSize',7)
            ylabel('Predicted Speed (Kt)','FontSize',7)
            set(gca,'xtick',[0 60 120 180 240 300 360],'ytick',[0 5 15 25 30])
            hold off

            subplot(2,2,4);plot(aci1,a4m1,'m>')
            hold on
            subplot(2,2,4);plot(aci1,a4m2,'m>')
            hold on
            subplot(2,2,4);plot(aci1,a5m1,'ko')
            hold on
            subplot(2,2,4);plot(aci1,a5m2,'ko')
            axis([-100 100 30 50])
            title('Fast Speed vs Angle','FontSize',10)
            xlabel('Side Slip Angle (Deg)','FontSize',7)
            ylabel('Predicted Speed (Kt)','FontSize',7)
            set(gca,'xtick',[-60 -30 0 30 60],'ytick',[30 35 40 45 50])
            hold off

% Finding max errors for the worst case
            for i=1:n
              fark(i)=(abs(ilk(i)-a(i)));
            end
            maxfark=max(fark)
            maxfark5=max(fark5)
            maxfark15=max(fark15)
```

```
maxfark25=max(fark25)
maxfark35=max(fark35)
maxfark45=max(fark45)

% Computation of SD and percent error for each speed:
mn5=mean(a1)
sd5=std(a1)
per5=(sd5/5)*100
mn15=mean(a2)
sd15=std(a2)
per15=(sd15/15)*100
mn25=mean(a3)
sd25=std(a3)
per25=(sd25/25)*100
mn35=mean(a4)
sd35=std(a4)
per35=(sd35/35)*100
mn45=mean(a5)
sd45=std(a5)
per45=(sd45/45)*100

% SD for whole data set

sdw=sqrt((((na/3)-1)*(sd5^2+sd15^2+sd25^2)+(((n-na)/2)-1)* sd35^2+sd45^2)) /(n-1))
```

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman
   Department of Aerospace and Aeronautics, Code AA
   Naval Postgraduate School,
   Monterey, California

4. Dr. Russell Duren
   Department of Aerospace and Aeronautics, Code AA/Dr
   Naval Postgraduate School,
   Monterey, California

5. Dr. Monique P. Fargues
   Department of Electrical and Computer Engineering, Code EC/Fa
   Naval Postgraduate School,
   Monterey, California

6. Ozcan Samlioglu
   7nci. Kor. Tk. Hv. Gr. K.ligi, Hlkp. Tb. K.ligi
   Diyarbakir, TURKEY

7. K K.K.ligi. Kara Harp Okulu Kutuphanesi
   Kara Harp Okulu, Bakanliklar,
   Ankara, 06200 TURKEY

8. Kara Kuvvetleri K.ligi Kutuphanesi
   Bakanliklar,
   Ankara, 06200 TURKEY

9. Hasan Akkoc
   Kara Harp Okulu Ogr. Bsk.ligi,
   Bakanliklar, Ankara, 06200 TURKEY