A PREDICTIVE AIRCRAFT LANDING SPEED MODEL USING NEURAL NETWORK

Ousmane N. Diallo, PhD, NASA Ames Research Center, Moffett Field, CA

Abstract

Expected increases in air traffic demand have stimulated the development of automated tools intended to assist the air traffic controller in accurately and precisely spacing aircraft landing at congested airports. Such tools require an accurate landing-speed prediction to increase throughput while decreasing the need for controller' interventions for avoiding separation violations. There are many practical challenges to developing an accurate landing-speed model that has acceptable prediction errors. This paper focuses on a near-term implementation, using readily available information, to model the final approach speed profile from the top-of-descent phase of flight to the landing runway.

The developed models accurately predicted the landing speed, for the MD-80 aircraft type operations at the Dallas/Fort Worth airport, 95% of times with error margins of 12.6% for the low-and-no gust and 12% for high gust conditions, respectively. Also, the models reduced the uncertainties of the landing speed predictions by at least 9.5% for both gust conditions from the current state-of-the-art predictions.

Introduction

The Federal Aviation Administration (FAA)'s plan to modernize the National Airspace System (NAS) in order to increase its capacity by 2025 is known as Next-Generation Air Transportation System (NextGen) [1]. One of the goals of NextGen is to accommodate the expected traffic-demand increase in the already congested terminal airspace. In the near term, this goal calls for tools to predict separation violations in order to plan conflict resolution strategies. Overall, there are two categories of tools for the final approach: tools that maximize throughput, such as the Traffic Management Advisor (TMA) [2] or the Terminal Area Precision Scheduling and Spacing System (TAPSS) [3], and those that predict separation violations such as the Automated Terminal Proximity Alert (ATPA) [4] or the Tactical Separation-Assisted Flight Environment (TSAFE) [5]. Both categories of tools, however, require accurate landing-speed estimates to provide accurate space between successive arriving aircraft because all the tools rely on predictions of the flight-path of the aircraft. The air traffic controllers' practices and procedures dictate aircraft speed throughout most of the airspace, but the flight crew (and aircraft procedures) decides the speed on approach in preparation for landing. Therefore, accurate landing-speed predictions are required for tools intended for use on final approach. Uncertainty in landing speed is manifested as either an increase in separation violations (i.e. a safety risk) or as increased separation buffers, a reduction in throughput.

There exist many different approaches to establish the landing speed of an aircraft to be used in the decision support tools. However, most of the options would require an upgrade to avionics equipment aboard the aircraft, increasing the airline's expenses, or verbal reports from the flight crew on landing speed, increasing the workload for both the flight crew and the controllers. Thus, to fulfill the goal of this work for near-term implementation with no additional equipment required, the focus of this research is to develop an accurate landing-speed model based on information that is readily available in today's air traffic control system.

In a previous study, the multivariable regression technique based on the response surface equation (RSE) is used to develop a statistical model of aircraft approach speed [6]. While the model yields an acceptable level of accuracy, the high number of variables it requires as inputs is unavailable at most airports. With the reduced number of input parameters, the RSE technique fails to provide an acceptable prediction model. To address this limitation, the alternative of the non-linear regression of the Neural Network modeling approach is used. The regression model is restricted to input variables that are more likely to be available (airlines internal data are usually difficult to obtain). The regression technique employed data for descent and approach phases of flight, as well as environmental and airport characteristics data.

Background

The forecasted increase in the traffic demand beyond the current capacity of the NAS may, in turn, lead to an increase in air traffic controllers' workload. This issue cannot be resolved by simply expanding the air traffic control (ATC) staff and facilities. The complexity of ATC, which central task is to ensure separation for all pairs of aircraft, grows quadratically with the number of aircraft [7]. Consequently, there is a need to create tools that can facilitate and strengthen the air traffic controllers' decision process, while still meeting the safety requirement. This paper focuses on such tools for the terminal environment, specifically those that address separation between successive arrivals. The specific challenge of such separation assurance lies in pursuing two competing goals: to maintain runway arrival throughput, and to ensure there are no separation violations between any pair of aircraft. The latter goal is set because separation violations carry the risk wake vortex hazards if too little separation is prescribed [8]. The task of separating aircraft becomes increasingly complex in highly congested terminal airspace [9].

Among the requirements for a model that addresses this challenge is the capability to predict an aircraft landing speed. The role of knowing landing speed is as follows. As a pair of successive arriving aircraft approaches the landing runway, there is a natural process of compression of the distance between the pair when the leading aircraft is slowed ahead of the trailing aircraft [10]. To determine the approach-speed profile of the trailing aircraft, one must know the landing speed of the leading aircraft. Since the air traffic controller does not prescribe a landing speed, those data must either come from the aircraft or be estimated. Landing-speed estimation must be sufficiently accurate for effective use by the different ATC tools. Currently, to mitigate the impact of inaccurate landing-speed estimates and to account for uncertainty in trajectory predictions, air traffic controllers often add excess separation beyond the required standards between pairs of aircraft. This practice leads directly to loss of runway throughput.

Motivation

The main objective of this work is to build a landing-speed model based on variables readily available in today's air traffic automation systems. To make this model useful, the required predictions of landing speed must be computed in real-time operations, efficiently and with sufficient accuracy.

There are three options to consider when determining the landing speed of an aircraft:

1. The flight crew communicates the intended landing speed to controllers electronically (e.g., via ADS-B message data).
2. The controller requests the landing speed from the flight crew by voice communication and feeds it into an interactive tool.
3. The landing speed of aircraft is predicted using a mathematical model.

Of these three options, the first requires installations of costly avionics aboard the aircraft, while the second increases workload for the controllers and the flight crew. Also, these two options require changes to the ground equipment and to controllers' procedures, significant undertakings that could face resistance from various parties involved. For these reasons, the third option is the approach of choice in this paper. It is geared toward near-term implementation, using readily available information to estimate the landing speed.

The modeling methodology in this paper uses the neural network regression technique and can be found in section 2. The currently known and used methods of landing speed prediction are summarized in section 3. A set of recorded data is used to build a predictive landing speed model using the model in section and the results are analyzed in section 4. Finally, a summary of the findings and a future direction of the research are laid out in section 5.

Modeling Methodology

All data-driven modeling frameworks for building predictive models rely on the quality of the raw data and on the processing methods used. To be useful, raw data typically need thorough pre-processing steps.

Data Selection and Processing

The approach taken in this paper to predicting landing speeds is multi-variables nonlinear regression that fits contributing inputs to the corresponding historical landing speeds as outputs. From among the different approaches to collect and process historical data, the following methodical three-step approach was chosen because it leverages previous work [11]. The three steps are: data collection, screening, and parameters selection decision.

Data Collection step: The first step of a model development consists of identifying and collecting as many variables as possible to make sure no impactful variables are overlooked. Thus, for this landing-speed modeling, a model is built for a specific aircraft type at a given airport because each airport has its specific characteristics. The raw data are composed of all airline monitored flight parameters (e.g., fuel consumption, aircraft weight, etc.), radar recorded data (e.g., ground speed, etc.) and airport environmental conditions (e.g., wind, visibility, ceiling, etc.).

Screening step: This step centers the analysis on variables that directly impact the modeling task. It serves to detect the parameters that contribute the most to the landing-speed behavior. The screening is necessary to get rid of duplicates and redundancies in the input information for the model. Also, obvious parameters with no impact (e.g., departure airport, etc.) on the model are eliminated. Other variables with no or little perceivable effect on the overall model accuracy such as city-pair, route information such as domestic or foreign (i.e. landing weight may vary due to requirement of reserve fuel), are discarded.

Parameters selection decision step: In this step, final parameters to be including in the model are decided based on the result of the screening as well as engineering judgment. Hence, it is important to have a good understanding of the system to be modeled. For instance, it is common practice by airlines to track both the planned and actual variables for many key parameters. However, in this work, only the actual values of the variables are used. The use of actual variables would allow achieving more realistic models because planned variables are typically based on the predicted actual variables augmented by a margin value to account for the uncertainty of flight parameters (e.g., fuel burn, time, etc.).

Neural Network Mathematical Modeling

The neural network (NN) is a mathematical modeling technique based on regression. Intended to mimic the cognitive and inferential functions of the human brain, a NN is capable of modeling highly non-linear behavior. In general NNs can be an alternative modeling methodology when linear modeling approaches perform poorly [12]. One of the easiest types of neural network to build and implement is the *feed-forward back propagation (FFBP)*. Usually, FFBP is a good starting point for building a model because a feed-forward neural network can in theory fit any nonlinear relationship between inputs and target. However, its main disadvantages for this type of study is that it works like a black box for the end user, providing no qualitative insight on the impact or the variance of each input variable on the landing speed.

Neural Network Background

Architecturally a FFBP NN is composed of an input layer, hidden layers, and an output layer, as illustrated in Figure 1.

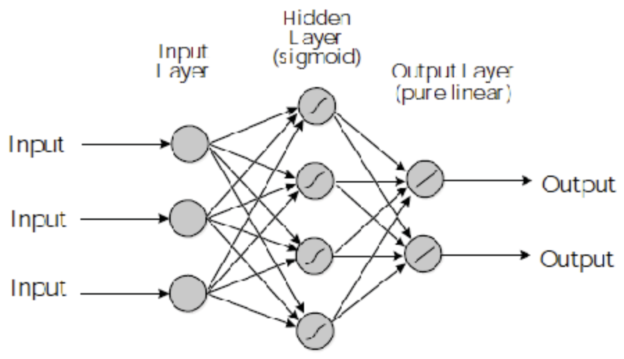

Figure 1. Feed-Forward Neural Network

While the hidden layer is the data processing center of the network, the output layer is the response to the simulation. A layer is composed of neurons. In a layer, each input element is connected to each neuron through a weight (w), and a bias (b) is added to the weighted input.

Mathematically, the value of the hidden node is computed by composing the logistic function

$$S(z)=\frac{1}{1+e^{-z}} \tag{1}$$

With a linear function that fits the design variables X_i's. The hidden node has the form

$$H_j = S\left(d_j + \sum_{i=1}^{N}\left(C_{ij}X_i\right)\right) \quad (2)$$

where:

d_j: intercept term for the jth hidden node

c_{ij}: coefficient for the ith design variable

X_i: value of the ith design variable

H_j: value of the jth hidden node

N: number of input variables

The logistic sigmoid function S defined by equation (1) is used to "squish" the inputs so that its output is a value between 0 and 1.

Typically, there are three main steps to building a neural network: training, validation and testing. Consequently, a dataset composed of an input set and corresponding target output are divided into those three groups. Usually there is a pre-processing step to get the data into a useful form. Then the training step consists of tuning the values of weights and biases of the network to optimize network performance [13]. The validation step consists of using the second (validation) set of the data to adjust the quality of the regression. Finally, the test set serves to check the network on a set different from the one used for its construction.

Among the many existing NN training algorithms, the one by Levenberg-Marquardt [14] is the most commonly used for feed-forward networks when rapid training is desired. Based on the goal of this research, to model a regression of the input variables to predict the corresponding landing speed, the network is trained for function approximation as opposed to pattern recognition. A NN training requires a training set based on known inputs and corresponding target outputs. It is also customary for neural networks to undergo the training more than once (a technique known as retraining) to improve the quality of the fit. Then to stop the training process, the Mean Square Error (MSE) performance function defined in equation (3) is used as the stopping criteria.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(e_i\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(t_i - a_i\right)^2 \quad (3)$$

where:

MSE: Mean Square Error, the average squared error between the network outputs and target outputs

e_i: ith error between the network outputs and target outputs

t_i: ith target or actual output

a_i: ith network predicted output

Model Evaluation

After the model is built, it needs to be evaluated to assess its effectiveness. The model evaluation consists of determining the predicted landing speed errors for the neural network model and the baseline (i.e. recommended target landing speed). To make this comparison the following two metrics are defined:

$Error_{Vtarget}$: defined as the absolute difference between the actual landing speed and the baseline speed (V_{Target}):

$$Error_{V_{Target}} = \left|\left(\frac{V_{Actual} - V_{Target}}{V_{Target}}\right) \times 100\right| \quad (4)$$

$Error_{Model}$: defined as the absolute difference between the actual landing speed and the landing speed predicted by the neural network model:

$$Error_{V_{Model}} = \left|\left(\frac{V_{Actual} - V_{Model}}{V_{Model}}\right) \times 100\right| \quad (5)$$

With:

V_{Actual}: Actual speed or true speed is defined as the sum ground speed (obtained from the terminal radar approach control facilities (TRACON) radar data) and the reported wind at the airport (from METAR),

V_{Target}: Target speed or flight manual-recommended landing speed is defined as the flight manual-recommended approach speed based on aircraft type and aircraft weight. This speed is used as the baseline landing speed for comparison.

V_{Model}: Landing speed of the model is defined as the neural network model predicted landing speed.

Flight Operating Manual – Baseline

There exists a flight-operating manual for each aircraft type. In this work, the flight operating manual recommendation for landing-speed prediction based on the landing-aircraft weight represents the baseline.

To assess the quality of the proposed landing-speed model, its predictions are compared to the state-of-the art prediction or baseline (i.e. the flight-operating manual recommendations).

The example of an aircraft type used here is the MD80, also known to as the Super 80, one of the most common narrow-body air-transport aircraft. For this aircraft type, the flight operating manual has two recommendations: one for low-and-no-gust condition defined as gust wind below 10 knots, and another for high-gust condition for gust wind greater than 10 knots.

Landing with Low-and-no-Gust Condition

In the low-and-no-gust condition on the final approach, pilots are recommended to target an approach speed of:

$$V_{Target} = V_{ref} + 5knots \qquad (6)$$

Where:

V_{Target} : Final approach target speed (knots - indicated airspeed (IAS))

V_{ref} : Reference speed (knots IAS)

The reference speed is provided in the flight manual as a function of aircraft weight and aircraft type.

Landing with High-Gust Condition

In the high-gust condition, the operations manual recommends that on the final approach the pilot target the speed of:

$$V_{Target} = V_{ref} + \frac{1}{2}SW + GW \qquad (7)$$

Where:

V_{Target}: Final approach target speed (knots IAS)

V_{ref} : Reference speed (knots IAS)

SW : Steady wind defined as the headwind component of the reported winds

GW : Gust wind defined as the headwind differential between the reported winds and the reported gusts

With the restriction that the total wind additive to the V_{ref} should not exceed 20 knots.

Proof of Concept and Results

Background on the Choice of Data Used

Previous studies [10, 15] suggested that airport-specific parameters (airport elevation, runway length, etc.) contribute to the final approach performance of a flight. Also, each aircraft type (e.g., B737-800 or MD80) has specific performance parameters that would directly impact its landing speed. To account for all these variations that may affect the fidelity of any landing-speed model, an airport-specific and aircraft-specific landing model is created using historical data.

Data Processing: Variables Used in the Study

As a proof of the concept, the proposed neural network is applied to the MD80 aircraft type operations at the Dallas/Fort Worth International Airport (DFW). DFW was chosen as a test case for the availability of data and because the SP80 is the highest number of a single aircraft type in service at this airport.

Step 1: Data collection

The raw dataset used for this study contained over 300 variables. These data consisted of airport topological (e.g., runway length and configuration), environmental information (e.g., visibility, wind condition, ceiling), flight-specific parameters (e.g., city pair flown, fuel burn rate, landing weight, etc.), and aircraft-specific parameters (e.g., empty gross weight, aircraft maximum payload, etc.). For the remainder of this paper the collected data are considered as the true value or actual value. Therefore these variables are assumed known exactly.

Step 2: Screening

The term *screening* is used in this paper to mean identification of the predictive values of the parameters. Screening is done here in two steps: elimination of irrelevancies and elimination of duplicate information to the model. Variables identified at the outset as irrelevant are automatically eliminated (e.g., aircraft tail number, city of departure, etc.). The elimination of redundancies is based on the statistical correlation between pairs of variables. If two or more variables provide the same information, only one of them is used for the modeling. For example, as shown in Figure 2, the variables *pax* (number of passengers) and *LF_pax* (Load Factor for passengers) have a correlation coefficient of 1. For modeling purposes these two variables provide the same information, so the use of one of them is sufficient. Consequently, after dropping the linear dependent parameters, *LF_pax*, *FBR* (Fuel Burn Rate), and *ESAD_nm* (Equivalent Still Air Distance) are among the retained variables for modeling, while *FGFH* (Flight Gallons/Flight Hours) and *GCD_sm* (Great Circle Distance (statute mile)) are dropped.

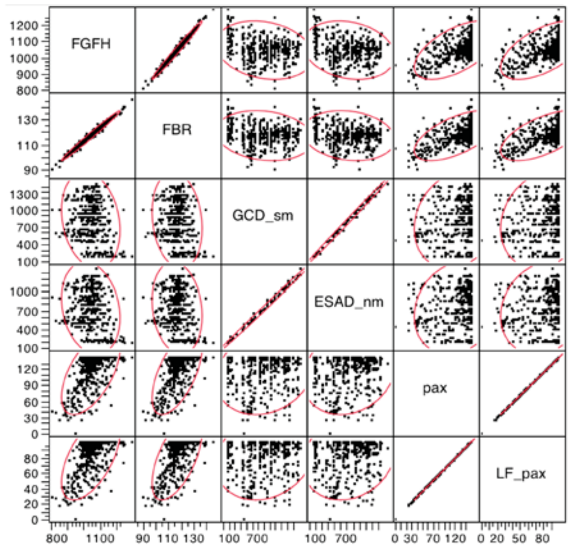

Figure 2. Example of Correlated Variables Matrix

Step 3: Selection decision

The result of the screening and physical relevance steps, based on the principle of using the minimum number of variables possible, yields a reduced set of thirteen parameters as the most relevant to aircraft landing speed. Then, In order to broaden the modeling approach to other major US airports, variables that are deemed difficult to obtain (i.e. airline proprietary data) are eliminated from the consideration. Finally, the result of the data processing is a reduced set of parameter of five variables shown in Table 1. These data are typically available to individuals through normal acquisition means (e.g., Federal Aviation Administration (FAA) websites).

Table 1. Variables Used for Modeling

Parameter	Description
Head wind	Head Wind
Gust wind	Gust Wind
Ceiling_ft	Forecast Ceiling
Vis_ft	Forecast Visibility
Act_Land_Wgt	Actual Landing Weight

Only the selected five variables are used as inputs variables to build the neural network model, where the output is the landing speed.

Neural Network Modeling

Because the baseline landing speed prediction is divided according to the landing gust condition, the data are divided into two groups as well: low-and-no-gust in and high-gust conditions.

Then, the five inputs variables shown in Table 1 are used as inputs to build a feed-forward neural network for each gust condition using the Matlab software. The neural network schematic illustrated in Figure 3 has three hidden layers and one output layer.

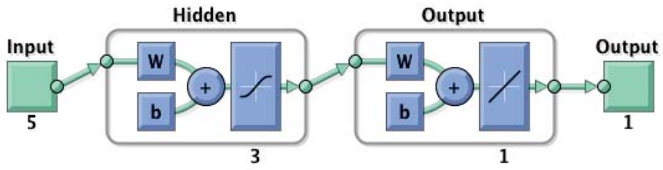

Figure 3. Neural Network Illustration

The dataset is randomly divided into three groups as follows: 70% used for training, 20% for validation, and the remaining 10% for testing. The rational for this percentage split of the dataset is to allow a large enough training set size, given the limited number of data point available. Then, the training process consists of tuning the weights and biases until the validation dataset converges as illustrated in Figure 4 for the high gust condition.

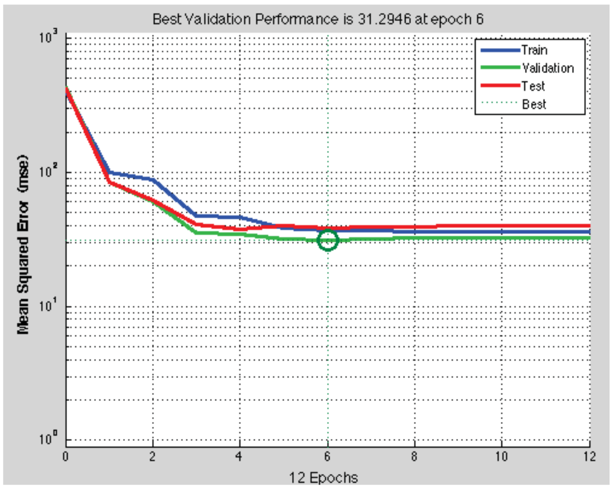

Figure 4. MSE Plot for High Gust Condition

In the example shown in Figure 4 for high gust case, converge occurs after the sixth iteration.

Neural Network Models Evaluation

After the models are built, and the landing speeds predicted, the quality of the predictions is evaluated.

Figure 5 and Figure 6 represent the graph of actual landing speed as a function of the predicted landing speed for the low-and-no gust and high gust respectively. If the models were perfect predictors of the actual recorded landing speeds, all the red stars would have fallen on the blue lines (also called the 45-degree line).

The quality of a fit can be assessed using the goodness-of-fit metrics of the coefficients of determination R^2 and the root mean square (RMS) error. For the above dataset, the R^2 and RMS error turn out to be 0.43 and 3.5 for the low-and-no gust, and 0.56 and 3.5 for high gust conditions, respectively.

Also, it can be inferred from Figure 5 and Figure 6 that there is sizable dispersion in the landing-speed prediction for the low-and-no-gust as well as the high gust conditions, despite the use of the same dataset for modeling and testing. This large dispersion is reflected in the lower R^2 values (ideal value is 1.0).

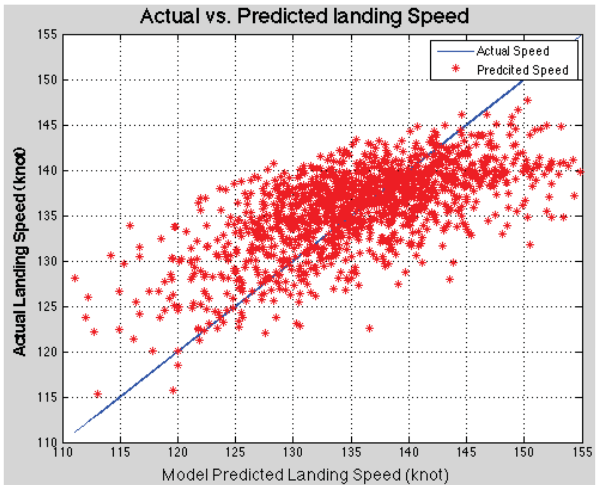

Figure 5. Actual vs. Predicted Plot for Low-and-No Gust

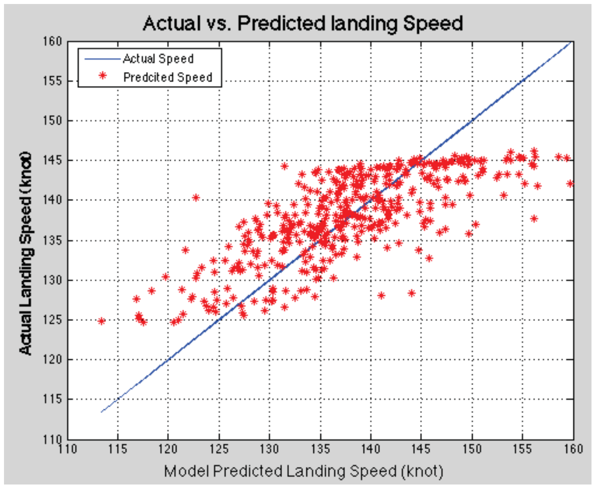

Figure 6. Actual vs. Predicted Plot for High Gust

Despite the lower number of data points (e.g. 426), the goodness-of-fit metrics show that overall the model for the high-gust condition is a better predictor than the model obtained for the low-and-no-gust wind condition (e.g. 1373 data points).

Results Discussion

Based on the clustered nature of the data points shown on Figure 5 and Figure 6, intuition suggests that the actual and predicted landing speeds present normal distribution characteristics. Consequently, the models errors distributions are more likely to have normal distribution characteristics as well.

As illustrated by the normal probability plot shown on Figure 7, the assumption of a normal distribution is reasonable for the low-and-no gust condition because all the data point fall near the line. For the high gust condition, a similar trend is observed. Therefore normal distributions statistics can be used to compare the models to their corresponding baselines. Table 2 is the summary of the calculated statistical parameters.

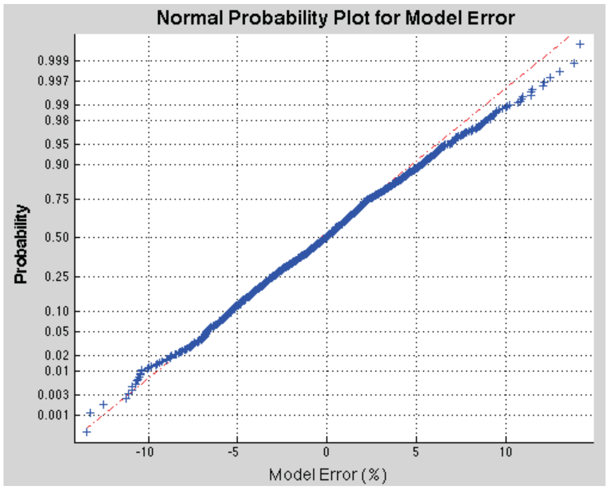

Figure 7. Normal Probability Plot for Low-and-No Gust Condition

Table 2. Summary of Landing Speed Errors Statistics

Statistical Parameters	Low-and-No Gust Error		High Gust Error	
	Model	Baseline	Model	Baseline
Mean	-0.1	4.2	-0.2	-6.9
std Dev	4.2	4.9	4.0	4.4
Min	-13.5	-14.5	-12.5	-19.2
Median	-0.1	4.3	-0.4	-7.3
Max	14.2	19.6	13.4	7.6
25%	-3.0	0.8	-3.0	-9.9
75%	2.3	7.4	2.3	-4.2

Table 2 indicates that the mean values of the percentage error of the models for both low-and-no-gust and high-gust conditions are an order of magnitude smaller than those of the baseline values. Furthermore, the standard deviation values are smaller for the models (4.2 and 4.0 respectively) than for the baseline (4.9 and 4.4 respectively). These statistics demonstrate that the neural network landing-speed prediction models outperform the baseline predictions of the actual landing speed for both the low-and-no-gust and high-gust conditions.

For both gust conditions, the error distributions for the models and for the baseline approximate a normal distribution as shown in Figure 8 (low-and-no gust case) and Figure 9 (high gust condition). A direct comparison between the distributions shows lower variance for the neural network models. For low-and-no gust condition the model error has a spread from the twenty-fifth percentile to seventy-fifth percentile of 5.3% (-3% to 2.3%) while the baseline has a spread of 6.6% (0.8% to 7.4%). Similarly, for the high gust condition, the spread from the twenty-fifth percentile to seventy-fifth percentile of model error is 5.3% (-3% to 2.3%) and the baseline spread is 5.7% (-9.9% to -4.2).

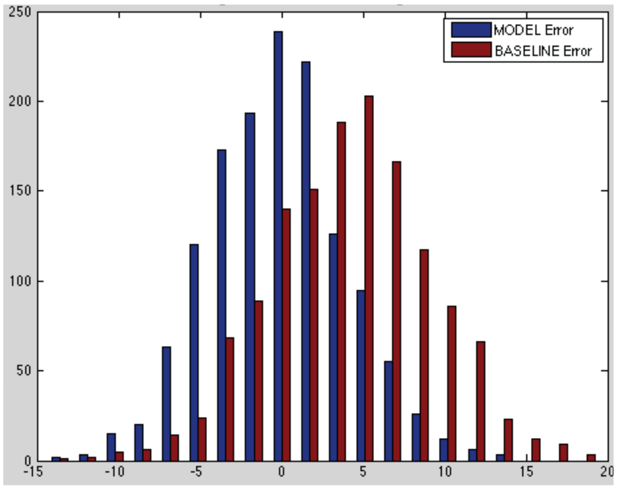

Figure 8. Error Distributions for Low-and-No Gust

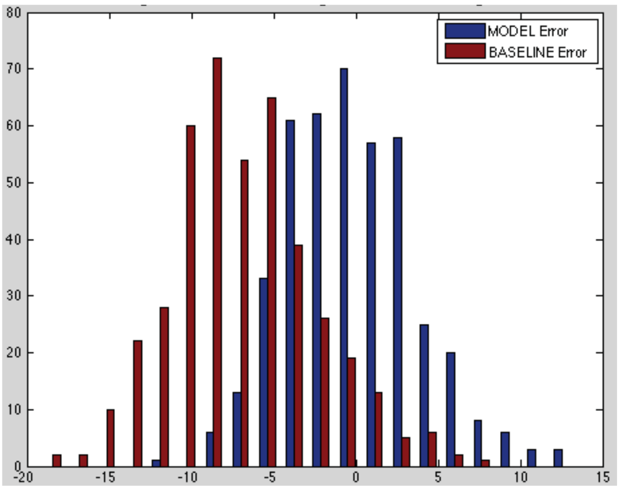

Figure 9. Error Distributions for High Gust

Another indication of the error distributions is that for the low-and-no gust case condition while the peak of the model is 0, the baseline's probability density function (pdf) attains a maximum at approximately 5%, i.e. the baseline exhibits a systematic error, with the distribution shifted to the right. By contrast, for the high gust case the baseline error distribution is shifted to the left with a pdf maximum at approximately 8%.

Results Interpretation

A quantitative analysis of the low-and-no-gust condition suggests that the neural network modeling approach provides a better landing-speed predictor than the baseline. First, because the mean values of the neural network models' error distributions are around 0%, it implies that they have superior accuracies than the baseline. Then, because the neural network models errors have a standard deviation of 4.2 and 4.0 for low-and-no gust and high gust conditions respectively compared to 4.9 and 4.4 for the baseline.

In essence, the error distributions of the neural network models represent a reduction of 18% and 9.5% from the baseline error distributions standard deviation for the low-and-no gust and the high respectively. In other words, the neural network models reduce the landing speed errors by 18% and 9.5% for the low-and-no gust and for the high gust respectively from the baseline landing speed prediction errors. In turn, these reductions in dispersion (standard deviation) can be directly translated in uncertainty reduction.

Based on the fact that the error distributions approximate normal distributions as illustrated on Figure 7, other interesting deductions can be made. For instance, since the standard deviation values are known, it can be said that the neural network models predict accurately 95% of the landing speed within an error margin of 12.6% and 12% (i.e. three sigma) for low-and-no gust and the high gust conditions respectively.

Another observation from Figure 8 is that for the low-and-no-gust condition, the flight-operating manual tends to underestimate the landing speed; that is, it under-predicts the landing speed (see equation (4)). In other words, at the low-and-no-gust condition, pilots overshoot (i.e. actual landing speed is higher than recommended) the landing speed compared to the flight manual recommendation. By contrast, in the high-gust condition error distribution shown in Figure 9, the flight-operating manual overestimates the landing speed. Thus, the predicted landing speed is over predicted compared to pilots' achieved landing speed.

Overall, for the normal landing-speed range, neural network modeling approach is capable of predicting the landing speed within a few knots for most data points.

Concluding Remarks

Summary

The quality of fits of the neural network developed for the variables listed in Table 1 are assessed with coefficient of determination R^2 of 0.43 for the low-and-no-gust model and 0.56 for the high-gust model. Based on the normal distribution nature of the error distributions, the neural network models are found to predict the landing speed in 95% of cases with an error margin of 12.6% for the low-and-no gust condition, and with an error margin of 12% for high gust condition.

The landing speed modeling technique presented in this paper is a promising research direction toward addressing the challenges in the prediction of final approach speed. The obtained predictions of landing speed achieve a higher accuracy (zero mean error) and higher precision (smaller variance) than do the current state-of-the art predictions of the flight operating procedure recommendations. Specifically, in the case of the narrow-body aircraft type used as test-bed, neural network models reduced the uncertainty of the landing speed prediction by 18% for the low-and-no gust and by 9.5% high gust conditions, respectively.

Furthermore, the study indicates that pilots tend to overcompensate landing speed in the low-and-no-gust condition and to under-shoot landing speed under high-gust conditions. These findings are very important in the modeling of pilots' behavior to improve existing and future terminal area tools.

Directions for Future Research

Among the reasonable next steps of this research is the application of these modeling approaches to other airports and aircraft types. Though a new network needs to be trained for each aircraft type and airport configuration.

Finally, the proposed modeling approach can be improved by identifying parameters more relevant (other than those used here) and finding a way to quantify pilots' decision-making processes for the landing procedure.

References

[1] Joint Planning and Development Office, 2007, "Concept of Operations for the Next Generation Air Transportation System," Version 2.0

[2] Swenson, H., T. Hoang, S. Engelland, D. Vincent, T. Sanders, B. Sanford, K. Heere, 1997, "Design and Operational Evaluation of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center," 1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France

[3] Swenson, H.N.; J. Thipphavong, A. Sadovsky, L. Chen, C. Sullivan, L. Martin, 2011, "Design and Evaluation of the Terminal Area Precision Scheduling and Spacing System", 9th USA/Europe ATM R&D Seminar, Berlin, Germany

[4] Functional Description Narrative N32422, "Automated Terminal Proximity Alert (ATPA)-Final Approach Course", ATO0T-CARTS-1055, U.S. Department of Transportation, Federal Aviation Administration, Washington, D.C.

[5] Paielli, R., E. Heinz, D. Chiu, K. Heere, 2009, "Tactical Conflict Alerting Aid for Air Traffic Controllers", Journal of Guidance, Control, and Dynamics, Vol. 32, No. 1

[6] Diallo, O., J. Robinson, 2012, "A Statistical Approach to Landing Speed Modeling", Second Aerospace Systems Conference, Los Angeles, California

[7] Sadovsky, A., D. Davis, D. Isaacson, 2012, "Efficient Computation of Separation-Compliant Speed Advisories for Air Traffic Arriving in Terminal Airspace", NASA/TM-2012-216033, Moffett Field, California

[8] FAA, 2010, "Radar Separation Minima", Order JO 7110.65T, Air Traffic Control, Ch. 5

[9] Robinson, J., M. Kamgarpour, 2010, "Benefits of Continuous Descent Operations in High-Density Terminal Airspace Under Scheduling Constraints", ATIO Conference, Fort Worth, Texas

[10] Robinson, J., O. Diallo, R. Reisman, 2011, "Comparison of Trajectory Synthesis Algorithms for Monitoring Final Approach Compression", ATIO Conference, Virginia Beach, Virginia

[11] Diallo, O. N., 2010, "A Data Analytics Approach to Gas Turbine Prognostics and Health Management", Ph.D. Thesis, Georgia Institute of Technology, Thesis office, Atlanta, Georgia

[12] Johnston, C., J. Schutte, 2009, "Basic Regression Analysis for Integrated Neural Networks (BRAINN) Documentation", Version 2.3, Georgia Institute of Technology, Atlanta, Georgia

[13] Beale, M., M. Hagan, H. Demuth, 2011, "Neural Network Toolbox User's Guide", Version R2011b

[14] Lera, G., M. Pinzolas, 2002, "Neighborhood Based Levenberg-Marquardt Algorithm for Neural Network Training", IEEE Transactions on Neural Networks, Vol. 13, No. 5

[15] Gong, C., A. Sadovsky, 2010, "A Final Approach Trajectory Model for Current Operations", ATIO Conference, Fort Worth, Texas

Acknowledgements

The author thanks Alex Sadovsky and Confesor Santiago for providing great recommendations that made this paper so much clearer.

31st Digital Avionics Systems Conference
October 14-18, 2012