JOURNAL OF AIRCRAFT
Vol. 36, No. 3, May–June 1999

Fault-Tolerant Neural Network Algorithm for Flush Air Data Sensing

Thomas J. Rohloff*
University of California, Los Angeles, Los Angeles, California 90095-1597
Stephen A. Whitmore†
NASA Ames—Dryden Flight Research Center, Edwards Air Force Base, California 93523
and
Ivan Catton‡
University of California, Los Angeles, Los Angeles, California 90095-1597

A fault-tolerant neural network algorithm was successfully developed for use with flush air data sensing systems. This algorithm is composed of a combination of aerodynamic and neural network models used to translate a discrete pressure distribution from the nose of an aircraft into a set of air data parameters, including static pressure, dynamic pressure, Mach number, angle of attack, and angle of sideslip. Techniques were developed to detect and eliminate the effect of a lost signal from the measured pressure distribution. This system was evaluated with archived data, and its performance was compared with a signal processing system based completely on aerodynamic models.

Nomenclature

M = Mach number
P_{fads} = individual pressure signals within P_{fads}, psf
$\bar{P}_{\mathrm{fads}}$ = array of pressure measurements from the flush air data sensing (FADS) instrumentation, psf
P_i = normalized FADS pressures
P_∞ = freestream static pressure, psf
$P_{\infty,NN}$ = normalized static pressure output from the neural network flush air data sensing (NNFADS) algorithm
$\boldsymbol{Q}_i$ = signal quality vector
q_c = dynamic pressure sensed by the FADS instrumentation, psf
$q_{c,NN}$ = normalized dynamic pressure output from the NNFADS algorithm
α_∞ = angle of attack (pitch) of the aircraft relative to the freestream, deg
α_{local} = angle of attack sensed at the surface of the FADS sensor, deg
β_∞ = angle of sideslip (yaw) of the aircraft relative to the freestream, deg
β_{local} = angle of sideslip sensed at the surface of the FADS sensor, deg
Γ = pressure difference from the triples algorithm, psf
γ = ratio of specific heats
$\Delta\alpha$ = up-wash factor, deg
$\Delta\beta$ = side-wash factor, deg
λ = cone angle of FADS pressure ports, deg
ϕ = clock angle of FADS pressure ports, deg

I. Introduction

ACCURATE measurements of air data parameters are important for both flight testing and control of aircraft. These parameters include the speed and direction of the air-mass velocity relative to the aircraft, as well as the freestream static pressure. According to Gracey,[1] air data measurements are typically performed using intrusive booms that extend beyond the local boundary layer. These booms have been found to be excellent at making steady-state measurements at low to intermediate angles of attack. However, the performance of these instruments deteriorates during high angles of attack and highly dynamic maneuvers. They are also sensitive to vibration and alignment error, and are susceptible to damage during both flight and maintenance.

Flush air data sensing (FADS) systems, described in Whitmore et al.,[2] were developed in response to problems associated with intrusive booms. These instruments infer the air data parameters from pressure measurements taken with an array of ports that are flush to the surface of the aircraft, and are thus completely nonintrusive. However, because the locations of the pressure measurements are on the outer surface of the aircraft, locally induced flowfields can seriously complicate the calibration of these devices. Additionally, the semiempirical models that have typically been used to process the FADS pressure signals have experienced numerical instabilities, which resulted in momentary degradations in the system performance. Current developments in FADS technology, described in Whitmore et al.,[3] have successfully improved algorithm stability, but at the cost of increasing the complexity of the algorithm. This paper seeks to apply a neural network approach to develop a FADS estimation algorithm that is inherently stable, and is easier to calibrate and implement than the existing FADS system.

The need to improve the FADS interpolation algorithms provides an ideal opportunity for the application of neural network techniques. FADS systems use an input vector composed of 11 pressure measurements, P_i, to estimate an output vector that includes four air data parameters (α_∞, β_∞, P_∞, and q_c). The relationship between these two vectors is complex and highly nonlinear. Computational fluid dynamics can be used to study the problem, but the need for a real time invertible model makes the application of this approach infeasible. Neural networks, which require large quantities of training data, are very well suited to situations such as this, where the more traditional approaches are either insufficient or too complex, but empirical data are plentiful. These neural network systems allow the correlation of complex nonlinear systems without requiring explicit knowledge of the functional relationship that exists between the input and output variables of the system.

Compared with existing semiempirical FADS techniques, neural networks have the advantage of being easier to develop while providing a greater level of detail in the mapping between the two vector spaces. The level of detail available with a neural network

---

Received Dec. 15, 1997; revision received Aug. 30, 1998; accepted for publication Sept. 10, 1998. Copyright © 1999 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

[*] Graduate Student, Department of Mechanical and Aerospace Engineering, 405 Hilgard Avenue. E-mail: rohloff@ucla.edu.

[†] Vehicle Dynamics Group Leader, Aerodynamics Branch, M/S D-2033, P.O. Box 273. Senior Member AIAA.

[‡] Professor, Department of Mechanical and Aerospace Engineering, 405 Hilgard Avenue. Senior Member AIAA.

flush air data sensing (NNFADS) system is limited only by the detail available in the flight data, and the computational time required to train those data to a network. Additionally, the difficult calibration process of the semiempirical approach is avoided, because the same information is characterized automatically during the neural network training process.

The development of the NNFADS system was initiated in response to the stability problems associated with the semiempirical system described by Whitmore et al.[2] A trained neural network provides a set of explicit calculations, and is therefore insusceptible to instabilities associated with iterative regression techniques. The first successful postflight demonstration of neural network techniques applied to the FADS system was reported in Rohloff and Catton.[4] The FADS air data estimation was shown to be adequately represented by a trained neural network. However, the data used to train this network only included a single flight profile, and the applicable range of this particular network was limited to similar flight conditions.

The next stage in the development of the NNFADS system, which was reported by Rohloff et al.,[5] used a wider range of flight data pieced together from multiple flight tests to provide a comprehensive training data set across the entire flight envelope of the F-18. Specific techniques were developed for extracting a proper set of neural network training patterns from an overly abundant archive of data. Additionally, the specific techniques used to train the neural networks for this project were reported, including the scheduled adjustments to learning rate parameters during the training process. Two neural networks were trained to estimate static and dynamic pressures across the entire domain on available flight data. The accuracy of these networks was shown to match the accuracy of the semiempirical system over a wide range of flight conditions from subsonic to supersonic speeds.

The results presented to date have focused on the development and evaluation of individual neural networks. The NNFADS system will require an array of these trained networks implemented within the structure of a fault-tolerant algorithm, which is able to detect and compensate for lost input signals from the FADS pressure readings. The details of this algorithm, including fault detection and management, are presented next.

II. Current FADS Technology

A prototype real time–flush air data sensing (RT–FADS) system has been developed at the NASA Dryden Flight Research Center.[2] This system was implemented and tested on the NASA F/A-18B Systems Research Aircraft (SRA) over the entire nominal flight envelope of the F-18 from takeoff to landing ($M < 1.6$, $\alpha_\infty < 50$ deg, -25 deg $< \beta_\infty < +25$ deg). The FADS system was compared with the ship system air data computer (ADC) measurements, and was shown to be robust to noise in the measured pressures. However, algorithm instabilities were encountered during the development of the RT–FADS system for certain flight conditions, which caused momentary degradations in the system performance. A new adaptation of the FADS system, currently being developed at NASA Dryden for the X-33 Single Stage to Orbit launch vehicle,[3] has successfully improved algorithm stability, but at the cost of increasing the complexity of the algorithm.

The hardware for the RT–FADS system is located in a modified radome of the SRA. The radome and the RT–FADS instrumentation are depicted in Fig. 1. The system consists of the FADS pressure port matrix and the associated measurement transducers. A matrix of 11 pressure orifices was integrated into a composite nosecap and attached in place of the noseboom. The locations of the ports on the nosecap are depicted in Fig. 2, and are defined in terms of clock and cone coordinate angles, ϕ and λ, respectively. The pressures at these locations are sensed by 11 digital absolute pressure transducers located on a palette inside the SRA radome.

The reference air data values, used in both the calibration of the RT–FADS model and in the training of the neural networks, were generated by combining information from various sources. These measurements included the onboard inertial navigation system attitudes, rates, and accelerations; radar tracking velocity and position data; and rawinsonde weather balloon sounding data. More detailed descriptions of the reference data set can be found in Whitmore et al.[2] and Rohloff et al.[5]

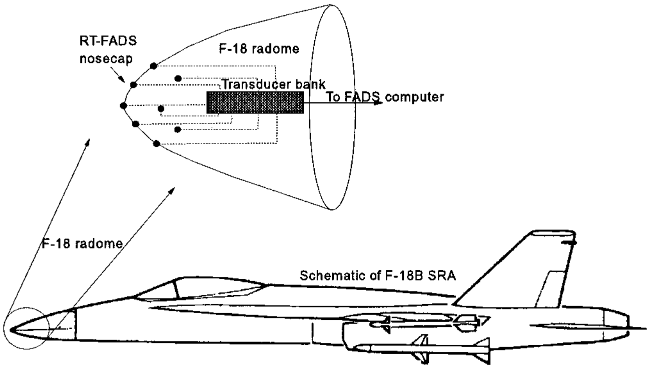

Fig. 1 FADS hardware.

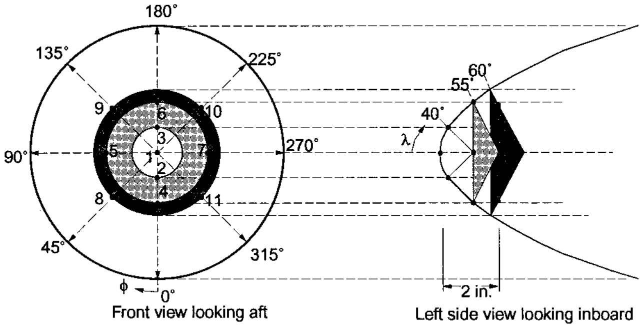

Fig. 2 FADS pressure port configuration.

III. NNFADS Processor

The NNFADS signal processor translates the pressure data from the FADS instrumentation (P_{fads}) into the more useful set of air data parameters (P_∞, q_c, M, α_∞, and β_∞). Unlike the preliminary neural network processor discussed in Rohloff and Catton,[4] the NNFADS processor does not rely on a single neural network to make the translation. Instead, this system was developed from an array of smaller components joined within the larger structure, outlined in the flow diagrams given in Figs. 3–7. The components of this system can be classified into two types: aerodynamic models and neural network models. The aerodynamic models, which were presented in Whitmore et al.,[3] have been adapted to the calculation of the local flow incidence angles (α_{local} and β_{local}), which are easily derived from the FADS pressure distribution. The neural network models are used to make the more complex extrapolation from local pressures (P_{fads}) and flow angles (α_{local} and β_{local}) to the corresponding freestream air data parameters (P_∞, q_c, M, α_∞, and β_∞).

A. General Structure of the NNFADS Processor

Understanding the details of the NNFADS processor first requires an understanding of its general structure. Figure 3 gives the broadest view of the signal processor. In this diagram the signals are received from the FADS pressure transducers and distributed to the first layer of components. This first layer includes the neural network-based calculations for static and dynamic pressures, as well as the aerodynamic model-based calculation of the local flow incidence angles. The outputs of this first layer are fed to the components of the second, which are used to calculate the Mach number and the freestream incidence angles. Each of the components shown in Fig. 3 is expanded on in Secs. III.A–III.D and in Figs. 4–7. Section III.A, in conjunction with Fig. 4, expands the portion of the NNFADS processor that comprises the neural network processor for static and dynamic pressure. This component of the NNFADS processor also has several subcomponents that must also be expanded upon. There are

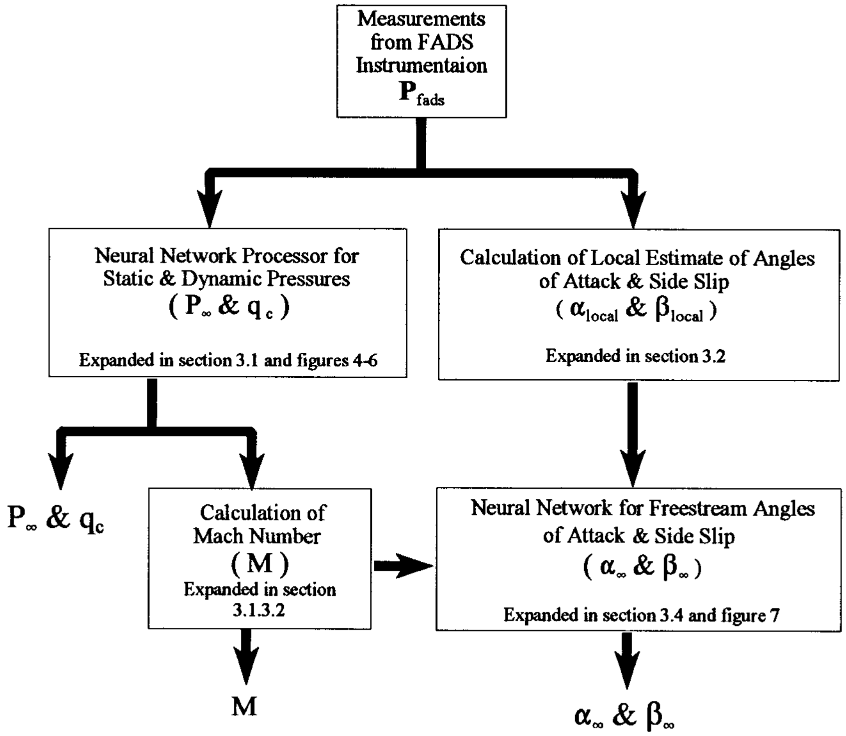

Fig. 3 General overview of the entire NNFADS processor.

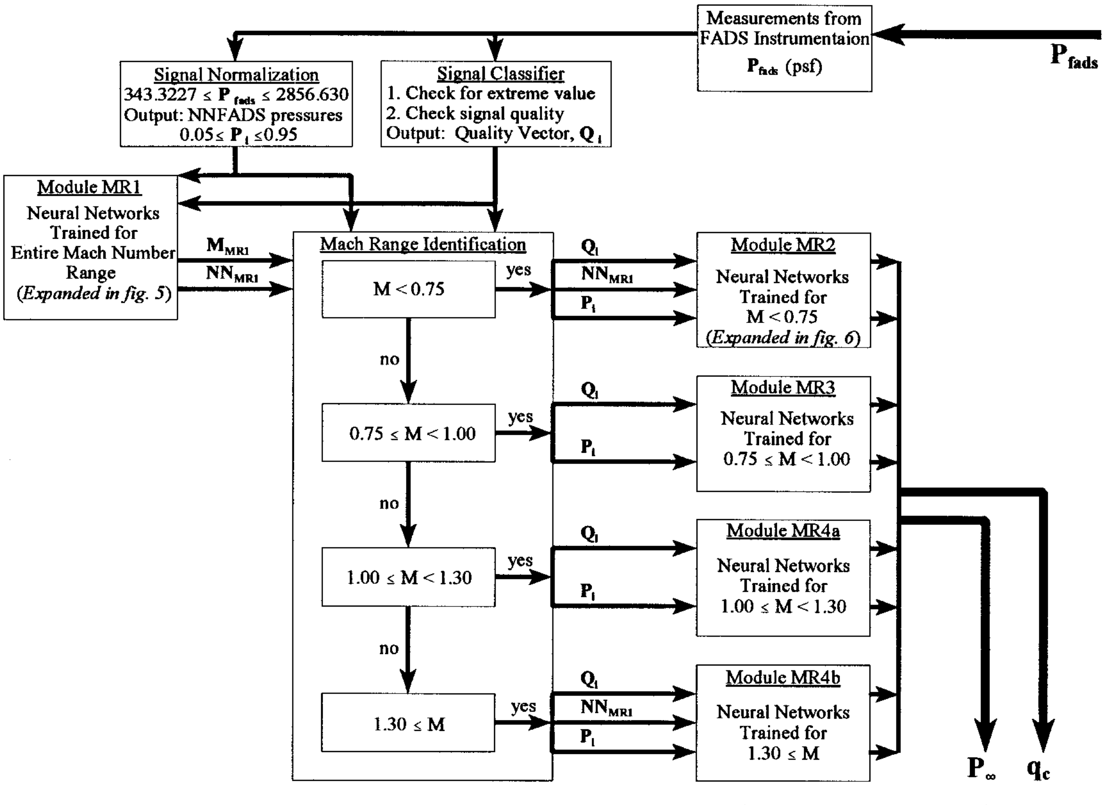

Fig. 4 Flow diagram for calculation of static and dynamic pressures (expanded view from Fig. 3).

five modules included in the diagram in Fig. 4. The first of these, module 1, is provided in more detail in Fig. 5. The remaining four modules are represented by the diagram of module 2 provided in Fig. 6. Referring once again to Fig. 3, the components used for the calculation of the local incidence angles and the Mach number are described in Secs. III.B and III.C, respectively. Finally, Sec. III.D and Fig. 7 outline the neural network processor for angles of attack and sideslip.

B. Neural Network Processor for Static and Dynamic Pressures

The neural network processor for static and dynamic pressures, which is outlined in Fig. 4, uses more than just one network to process the signals. In fact, this entire component of the NNFADS processor contains a total of 20 individual networks working cooperatively. Each of these individual networks was trained to represent a different piece of the FADS transformation, and so only a portion of the total set is needed for any time frame. All of the neural networks

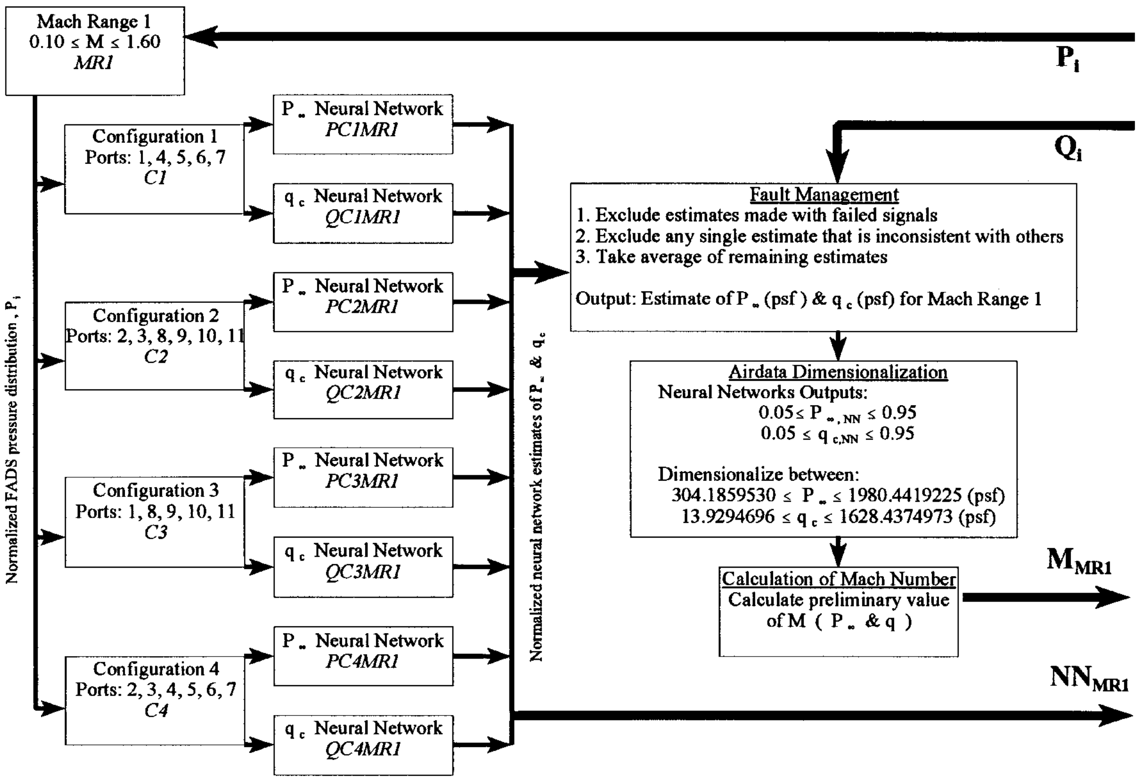

Fig. 5 Flow diagram for preliminary estimate of Mach number, module MR1 (expanded view from Fig. 4).

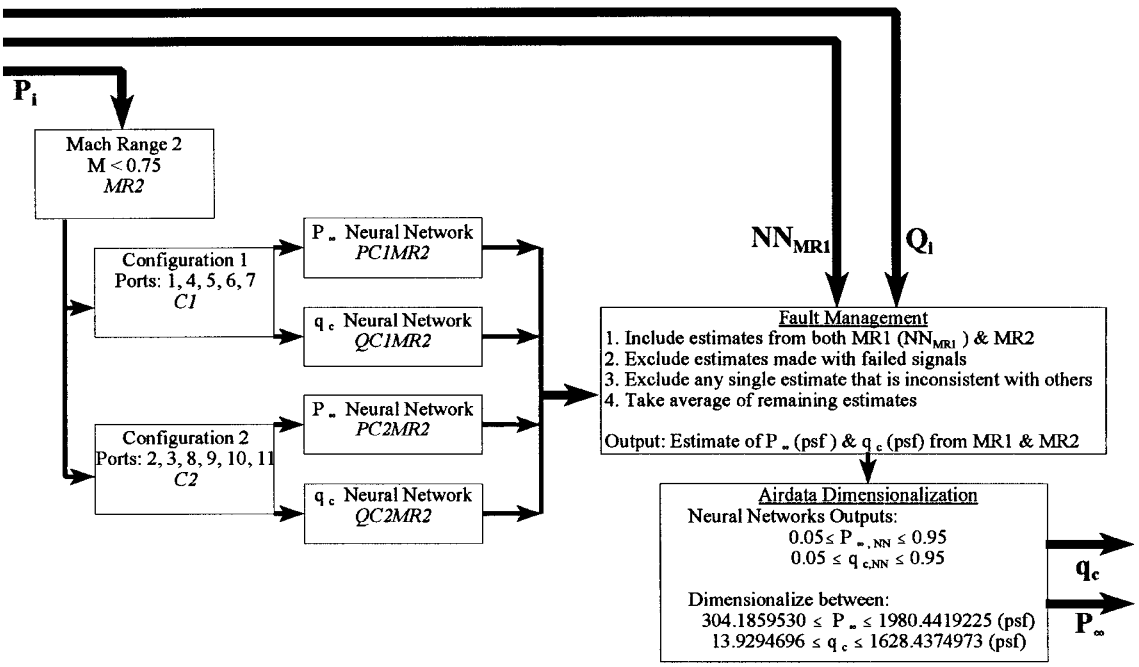

Fig. 6 Flow diagram for refined estimate of static and dynamic pressures for $M < 0.75$, module MR2 (expanded view from Fig. 4).

used here included two hidden layers of 20 nodes each, using sigmoidal transfer functions. The details of the input and output layers are specified in Tables 1 and 2.

The calculations for each frame of data are performed in three steps. First, the pressure data are normalized using the procedure outlined in Sec. III.B.1. Next, the input signals are checked for physically improbable values, as described in Sec. III.B.2. Finally, the techniques described in Sec. III.B.3 are used to calculate estimates for static and dynamic pressure.

1. Signal Normalization

The neural networks trained for this system were developed to accept sets of normalized pressure signals, as depicted in Fig. 4. During the compilation of the training set, the flight data archives were scanned for the extreme values recorded from the FADS pressure transducers. The FADS pressure signals were found to fall in the following range:

$$343.3227 \leq P_{\mathrm{fads}} \leq 2856.630 \text{ psf}$$

These extreme values were used as the maximum and minimum values in the normalization of FADS pressures to values between 0.05 and 0.95. The normalized values are given by

$$P_i = 0.05 + 0.9\left(\frac{P_{\text{fads}} - P_{\text{fads,min}}}{P_{\text{fads,max}} - P_{\text{fads,min}}}\right) \tag{1}$$

Normalizing between 0.05 and 0.95, instead of 0 and 1, guarantees that future pressure readings that happen to fall slightly outside the original range will still have normalized values between 0 and 1.

Table 1 Pressure port configurations used in NNFADS

Configuration name	Port indices
C1	1, 4, 5, 6, 7
C2	2, 3, 8, 9, 10, 11
C3	1, 8, 9, 10, 11
C4	2, 3, 4, 5, 6, 7

This should improve the performance of the system for short-range extrapolation.

2. *Signal Classification*

The signal classification algorithm, referred to in Fig. 4, is currently used only as a check for physically improbable values in the FADS pressure data. The routine receives the 11 raw pressure signals, P_{fads}, and returns the signal quality vector Q_i, which has 11 components corresponding to each of the incoming signals. If a signal is within the physically reasonable range, then its corresponding component in the quality vector is set to 1. Otherwise, that component of Q_i is set to 0. Considering the values of P_{fads} used for signal normalization, the physically reasonable range has been arbitrarily set at the slightly broader range of

$$300 \le P_{\text{fads}} \le 2900\,\text{psf}$$

This range may be adjusted in the future, after testing of the NNFADS system is complete. The final values of the physically

Table 2 Neural networks used in the calculation of static and dynamic pressures[a]

Module				
MR1 $0.10 \le M \le 1.60$	MR2 $0.10 \le M \le 0.75$	MR3 $0.75 < M \le 1.00$	MR4a $1.00 < M \le 1.30$	MR4b $1.30 < M \le 1.60$
PC1MR1	PC1MR1	——	——	PC1MR1
QC1MR1	QC1MR1	——	——	QC1MR1
PC2MR1	PC2MR1	——	——	PC2MR1
QC2MR1	QC2MR1	——	——	QC2MR1
PC3MR1	PC3MR1	——	——	PC3MR1
QC3MR1	QC3MR1	——	——	QC3MR1
PC4MR1	PC4MR1	——	——	PC4MR1
QC4MR1	QC4MR1	——	——	QC4MR1
——	*PC1MR2*	*PC1MR3*	*PC1MR4a*	*PC1MR4b*
——	*QC1MR2*	*QC1MR3*	*QC1MR4a*	*QC1MR4b*
——	*PC2MR2*	*PC2MR3*	*PC2MR4a*	*PC2MR4b*
——	*QC2MR2*	*QC2MR3*	*QC2MR4a*	*QC2MR4b*

[a] The entries in italics are part of the set of second-phase neural nets, and the others are from the first phase.

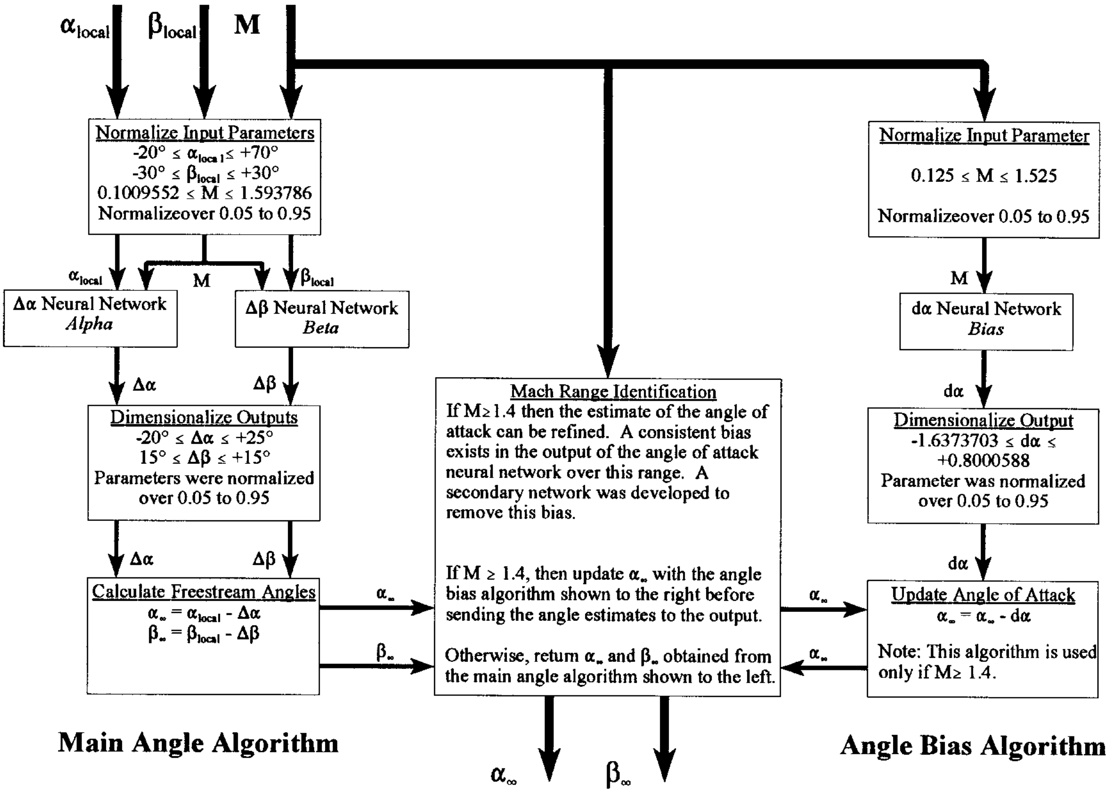

Fig. 7 Flow diagram for calculation of angles of attack and sideslip (expanded view from Fig. 3).

reasonable range will be dictated by the ability of NNFADS system to extrapolate outside the domain of the original training set.

A more comprehensive signal classification scheme, similar to the system described in Guo and Nurre,[6] is currently under development. This new approach will use a system of neural networks to check for inconsistencies between the raw FADS pressure signals. The 11 pressures are not all independent. It should therefore be possible to develop a system to determine if any single pressure reading is inconsistent with the remaining 10. The system being proposed will take the 11 signals and pass them through a system of neural networks trained to evaluate the quality of the signals relative to others. The output of such a network would be a quality vector similar to that described earlier.

3. *Calculation of Static and Dynamic Pressure*

The calculation of static and dynamic pressures is performed in two steps. The first step is used to get a rough estimate of the air data values including Mach number, and the second uses the information from the first to refine the air data estimates.

The nondimensional signals are first processed through module MR1, as shown in Fig. 5. The name of this module refers to the range of Mach numbers used in training the associated neural networks. The range included for MR1 covers the entire domain available in the data archives, $0.1 \leq M \leq 1.6$. The neural nets included in module MR1 are therefore generally applicable for any set of FADS data. However, these nets must represent a very broad range of flow conditions, including both the subsonic and supersonic regimes, and the accuracy of the associated air data estimates is not always ideal.

A second step in the air data calculations was added to improve the accuracy of the NNFADS processor. In this step, the estimates of module MR1, including the initial estimate of the Mach number, are passed on to one of the four remaining modules. Each of these four modules includes neural networks developed to handle a different flow regime. Subsonic flow, $M \leq 0.75$, is processed through module MR2, as shown in Fig. 6. Transonic flow, $0.75 \leq M \leq 1.00$, is processed through module MR3. Low supersonic flow, $1.00 \leq M \leq 1.3$, is processed through module MR4a, and high supersonic flow, $1.30 \leq M$, is processed through module MR4b. Modules MR4a and MR4b actually use the same secondary neural networks, but the signals are processed slightly different. The initial estimate of M provided by module MR1 is thus used to determine the appropriate secondary module.

a. Multiple configurations. A major concern during the implementation of this neural network-based FADS system will be the robustness to signal failure. The preliminary study, discussed in Rohloff and Catton,[4] centered on a single neural network constructed to use all 11 FADS pressure signals in the calculation of the four air data parameters. Thus, all 11 pressure signals had to be available for this preliminary network to operate properly. A single lost signal could cause that initial system to fail. However, it was discovered in Rohloff et al.[5] that neural networks with smaller input layers could also be trained to make the air data estimates from a reduced set of the FADS pressure signals. More specifically, networks were successfully developed to make air data estimates from as few as five or six of the pressure signals. This discovery was taken advantage of by training several different neural networks for groupings of five or six of the FADS signals each. The estimates from the different pressure groupings can then be averaged to give a best estimate. If any single pressure signal is identified as corrupt, then any neural net trained with a grouping that includes the failed signal can simply be excluded from the vote. Given that only a limited number of configurations can be reasonably included, it is not possible to guarantee that the system of networks will be able to handle the loss of two or more signals simultaneously, but there will be some combinations of two, three, and even four lost signals that will not affect all of the available networks. The groupings used in this study, which are listed in Table 1, refer to the port index system provided in Fig. 2.

Note that each grouping in Table 1 is symmetric about both the vertical and horizontal axes. Furthermore, each configuration includes both points from the center region ($\lambda \leq 40$ deg), and from the outer rings ($\lambda \geq 55$ deg). These configurations were chosen to mimic other multiport air data instruments that have been successfully developed in the past.[7–10]

All four configurations are used for the estimates of the two air data parameters calculated in module MR1, so that a total of eight neural nets were trained for use in this module. The remaining modules only use the first two configurations, and thus include only four neural nets each. Considering the repeated use of Mach number range MR4 for both module MR4a and module MR4b, a total of 20 networks was developed for the calculation of static and dynamic pressures. The names given each of these 20 neural networks, which are listed in Table 2, are derived from the ranges to which they are applicable, including the output parameter (P_∞ or q_c), the configuration, and the Mach number range.

b. Fault management. The introduction of multiple configurations in the neural net calculations of static and dynamic pressure has two main advantages. First, if the quality of the signal from any one pressure is suspect, then groupings that included that signal can simply be excluded during the final calculations. Second, after corrupted estimates have been excluded, the remaining estimates provide a redundant set of estimates of the same parameter. Because some of these remaining estimates will be from independent sets of input signals, agreement between the estimates will imply a high degree of confidence in the final output.

As shown in Fig. 4, the calculations of static and dynamic pressure were formed in two phases. The first phase was used to get an initial estimate of the Mach number, and the second was used to refine the estimates of P_∞ and q_c. During the first phase, outlined in Fig. 5, four pressure port configurations were used in calculating each of the two different parameters. That means that up to four different estimates were available in the initial calculations. During the second phase, an example of which is outlined in Fig. 6, only two configurations were used for the refined estimates of each of the two parameters. However, the accuracy of these secondary estimates was not always better than those from the preliminary stage, module MR1. In fact, for $M < 0.75$ and $M > 1.3$, the accuracy of both phases was about the same. Thus, estimates from both phases could be used in the final calculations. The sets of neural network estimates used in the calculation of the output of each of the modules are listed in Table 2.

Each of the five modules uses the same routine in the error management phase, but the neural network estimates being implemented vary between the modules according to the schedule listed in Table 2. The steps used in the error management algorithm include the following:

1) Exclude any estimate with a failed input signal in its grouping. The quality of each of the signals is given by $\boldsymbol{Q}_i$ from the signal classification algorithm.

2) Calculate the average value from the remaining estimates.

3) Exclude any individual estimate that varies by more than 0.003 from the mean. This value of the tolerance represents the expected variance between the estimates multiplied by a factor of 3. The expected variance was determined for a set of test cases. A value of three times the expected variance was used as the cutoff for statistically reasonable variations.

4) Recalculate the mean from the remaining signals, and return this estimate. If there are no signals left after steps 1–3, then calculate the mean from all of the available estimates and return that value.

The fault management scheme described here guarantees that the system will continue to operate after any single signal turns off. Additionally, this procedure will enhance the tolerance of the system to large-scale noise in the input signals. There are two sets of two independent groupings of the available pressure ports. There are therefore at least four estimates of the static and dynamic pressures. The consistency between these estimates is evaluated with the preceding procedure, so that any single estimate that is corrupted can be removed from the calculations. As noted earlier, the expected variance between the estimates was evaluated using 40,000 samples.

c. Air data dimensionalization. The output of the neural networks is a nondimensional value from 0.05 to 0.95, and it must therefore be dimensionalized. The following values were used as the extreme values over which static and dynamic pressures were normalized:

$$304.19 \le P_\infty \le 1980.44 \text{ psf}, \qquad 13.93 \le q_c \le 1628.44 \text{ psf}$$

The following equations can be used to transform the neural network outputs into a dimensional form:

$$\begin{aligned} P_\infty &= P_{\infty,\min} + \frac{(P_{\infty,NN} - 0.05)(P_{\infty,\max} - P_{\infty,\min})}{0.9} \\ q_c &= q_{c,\min} + \frac{(q_{c,NN} - 0.05)(q_{c,\max} - q_{c,\min})}{0.9} \end{aligned} \tag{2}$$

d. Mach number calculation. The following relation for isentropic flows, available in a variety of thermodynamics texts, e.g., Anderson,[11] can be used to calculate M from the static and dynamic pressures for subsonic flows:

$$M = \left\{\left[(q_c/p_\infty + 1)^{(\gamma-1)/\gamma} - 1\right][2/(\gamma-1)]\right\}^{\frac{1}{2}} \tag{3}$$

The value of the dynamic pressure used in this study is measured at the nose of the aircraft, and is therefore only valid behind the shock for supersonic flow. The relationship between static and dynamic pressures and the Mach number for supersonic flow is described by the Rayleigh pitot equation, which is derived from adiabatic normal shock-wave relationships[12]:

$$\frac{q_c}{P_\infty} = \frac{\left\{[(\gamma+1)/2]M_\infty^2\right\}^{\gamma/(\gamma-1)}}{\left\{[2\gamma/(\gamma+1)]M_\infty^2 - [(\gamma-1)/(\gamma+1)]\right\}^{1/(\gamma-1)}} - 1 \tag{4}$$

Over the range of Mach numbers and flow conditions observed in this study, γ can be considered constant and equal to 1.4.

C. Local Incidence Angles

The calculations of the local angles of attack and sideslip are based on the meridian triples algorithm.[3] Using any three pressures along the sensor meridian, i.e., $\phi = 0$ or 180 deg, the local angle of attack can then be calculated. Once the proper solution for angle of attack has been determined, the local angle of sideslip can then be calculated using any three pressure signals from the set. The reader is referred to Whitmore et al.[3] for more details on the meridian triples algorithm. The specific steps used by the NNFADS processor to implement the estimates from the triples algorithm are given next.

Several combinations of three pressure measurements can be used during the angle-of-attack and angle-of-sideslip calculations. These different combinations will produce a set of redundant estimations of the two angles from which a best estimate can be derived. The following algorithm is used to filter out inconsistent values of the two angles from the redundant set before the average is calculated:

1) Calculate the average of all physically reasonable estimates.

2) Calculate the rms error of the full set.

3) If the rms error is less than 1 deg, then return the average from the full set, else

4) Exclude all estimates for which the absolute error is greater than half the rms error.

5) If there are no values remaining, then return the average of the full set, else

6) Return the average value of the reduced set.

This routine ensures that any substantially inconsistent values of the incidence angles will not corrupt the overall estimates. The cutoff values of 1 deg, used in step 3, and half the rms error, used in step 4, were determined heuristically. However, the results of this algorithm were found to be consistent with the full RT–FADS system only after the previously mentioned filter was implemented.

This routine assumes that the flow remains attached over the nose of the aircraft. When flow separation occurs, the reading for the pressure transducer at the top, P_6, will be read much higher than if the flow remained attached. This would consequently corrupt the calculated estimate of both α_{local} and β_{local}. To remove the effects of flow separation on these calculations, the readings from port number 6 were excluded if it appeared that separation was indeed occurring. If the initial estimate of α_{local} was greater than 10 deg, and $P_6 > P_1$, then the component of the quality vector associated with P_6 was set to zero, and the suspect pressure reading was excluded.

D. Final Mach Number Calculation

This section is included to maintain consistency with the general structure presented in Fig. 3. A detailed description of the Mach number calculations is given in Sec. III.B.3.d. The same procedures are again used here.

E. Neural Network Processor for Angles of Attack and Sideslip

The characteristics of the neural processor for angles of attack and sideslip, which are outlined in Fig. 7, are somewhat different from those for static and dynamic pressures. Instead of using the array of 11 FADS pressures as input to the networks, this processor takes advantage of some of the results from the aerodynamic model. Evidence from the FADS literature[2] suggested that the freestream incidence angles could be related to the calculated values of the local incidence angles in combination with the Mach number. So the neural networks trained for this component of the NNFADS processor use M, α_{local}, and β_{local} as inputs for the estimation of α_∞ and β_∞. The local incidence angles can be related to the freestream values by

$$\alpha_\infty = \alpha_{\text{local}} - \Delta\alpha, \qquad \beta_\infty = \beta_{\text{local}} - \Delta\beta \tag{5}$$

where $\Delta\alpha$ and $\Delta\beta$ are functions of M, α_{local}, and β_{local}. The structure around the two main neural networks is very similar to networks discussed earlier. The inputs must be normalized before being sent to the neural network, and the outputs must be dimensionalized. The limits used in transforming these parameters are included in Fig. 7, and the calculations follow directly from the examples given earlier. The neural networks used here included two hidden layers of 20 nodes each, using sigmoidal transfer functions.

One additional improvement can be made to the estimate of angle of attack. If $M \ge 1.4$, then the estimate can be refined. A consistent bias was discovered in the output of the angle-of-attack neural network for these high Mach numbers. This bias can be seen in Fig. 8, which graphs the mean error in the estimate of α_∞ as a function of Mach number. The markers in Fig. 8 represent the bins in Mach number used in calculating the local statistics. A secondary neural network was developed to remove the bias. The curve fit shown in Fig. 8 is the output of the trained neural network. Using this curve fit, the bias can simply be subtracted out.

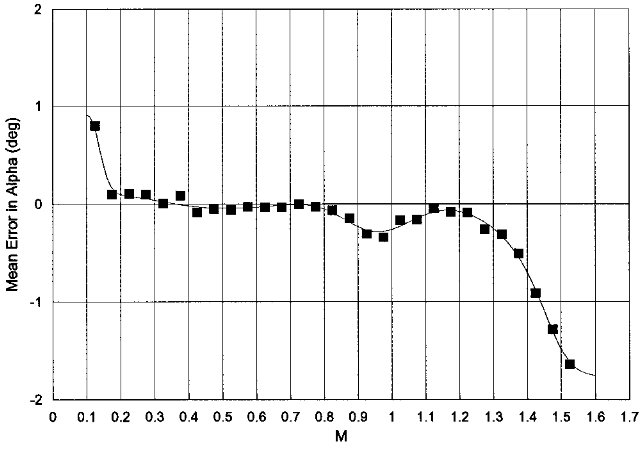

Fig. 8 Curve fit for bias in angle-of-attack calculations: ■, NNFADS, and ——, curve fit.

IV. Results

The performance of the NNFADS processor was evaluated with the FADS flight data available from the NASA Dryden storage facilities. The flight profiles, which were discussed in Rohloff et al.,[5] were extracted at 20 samples per second, to provide a total of nearly 1.5 million frames of data. This data was processed through the NNFADS system, and the results were summarized in Figs. 9–15.

A high-resolution graph of a high-Mach-number maneuver is provided in Fig. 9. During this maneuver, the aircraft accelerated from $M = 0.8$ up to 1.53, and then back again. The estimates for both NNFADS and the ADC were graphed as a function of time. The only significant variation between the two curves occurred at Mach numbers near 1.0. This is the transition point between subsonic and supersonic flow, and the characteristics of the flowfield change drastically during this region. However, the NNFADS predictions are within an acceptable range of error (less than 3%).

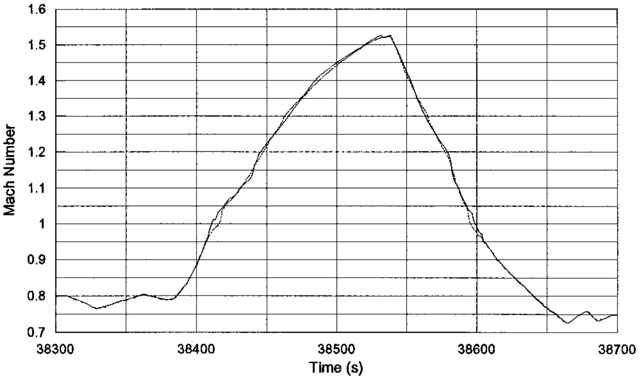

Fig. 9 NNFADS estimates for a high-Mach-number maneuver: ——, ADC, and ·····, NNFADS.

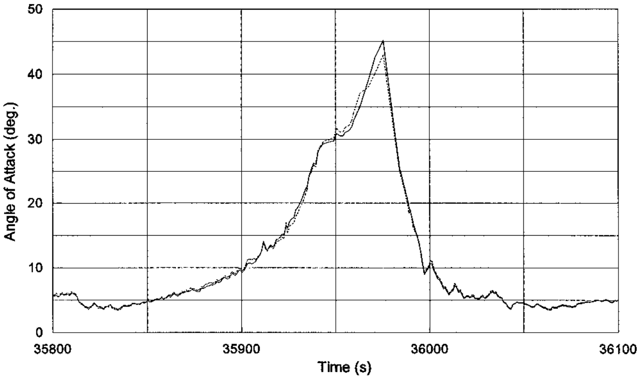

Fig. 10 NNFADS estimates for a high-angle-of-attack maneuver: ——, ADC, and ·····, NNFADS.

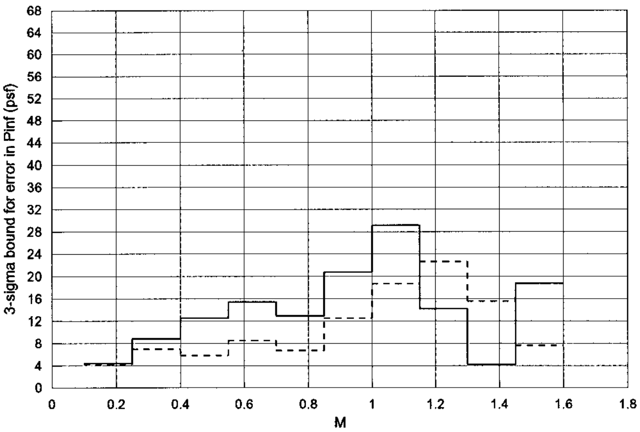

Fig. 11 Error bounds on P_∞ evaluated for RTFADS and NNFADS: ——, RTFADS, and - - -, NNFADS.

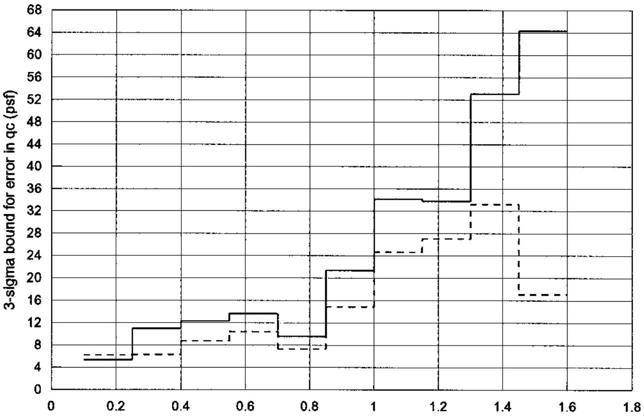

Fig. 12 Error bounds on q_c evaluated for RTFADS and NNFADS: ——, RTFADS, and - - - , NNFADS.

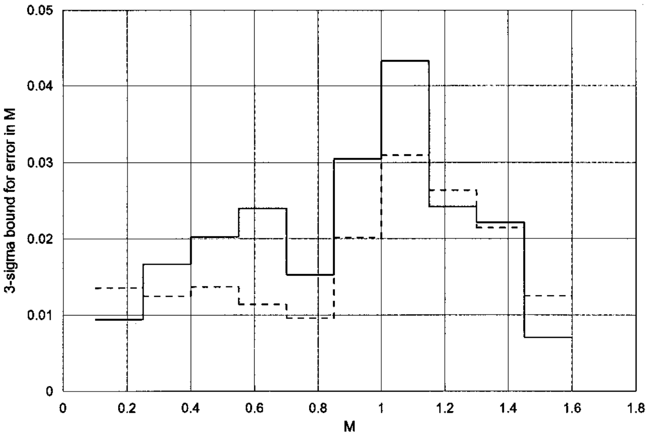

Fig. 13 Error bounds on M evaluated for RTFADS and NNFADS: ——, RTFADS, and - - -, NNFADS.

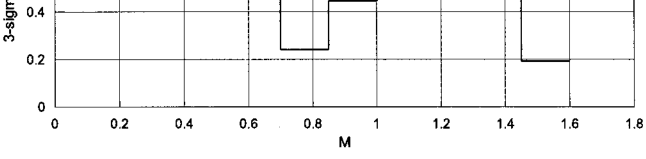

Fig. 14 Error bounds on α_∞ evaluated for RTFADS and NNFADS: ——, RTFADS, and - - -, NNFADS.

The NNFADS system is shown to be stable through the Mach number transition points in the estimation of static and dynamic pressure, discussed in Sec. III.A.3.*b*. The system switches from one set of neural networks to another at these points, and there was a concern that there might be discontinuities in the air data estimates as the system traversed these regions. The switchover points included M equal to 0.75, 1.0, and 1.3. It can be seen in Fig. 9 that the estimate for Mach number was continuous through these points. The same observation was made for the estimates of both static and dynamic

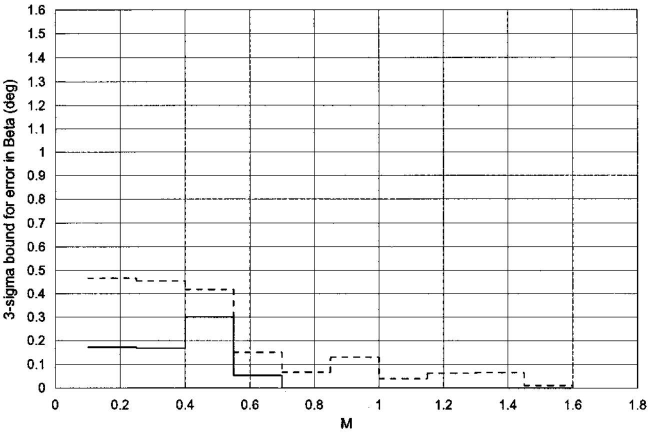

Fig. 15 Error bounds on β_∞ evaluated for RTFADS and NNFADS: ——, RTFADS, and - - -, NNFADS.

pressure over a wide range of example flight profiles. While this is not a conclusive proof of absolute stability, it is a strong indication that the NNFADS system will not have problems at the Mach number transitions.

A graph of a high-angle-of-attack maneuver is provided in Fig. 10. During this maneuver, the aircraft pitched up from $\alpha = 5$ to 45 deg, and back again. Comparing the NNFADS estimates with the ADC curve, it can be seen that the two agree quite well for $\alpha < 30$ deg. At higher angles, the agreement is not quite as good, but the variation is always less than 5%. It may be possible that flow separated off the nose of the aircraft during this maneuver. This could have caused more than one of the upper pressure ports to have higher than normal readings. This would result in an underprediction of α similar to that shown in Fig. 10.

The full set of 1.5 million data frames were processed through both the RT–FADS and NNFADS processors, and the results were compared with the ADC measurements. The rms error of estimate of each of the air data parameters was evaluated as a function of Mach number. Defining the 3-sigma error bound for the air data estimates as three times the rms error, the error bounds for both the RT–FADS and NNFADS estimates were graphed as a function of M in Figs. 11–15. A total of 10 bins were used in summing up the rms error for each air data parameter.

It can be seen from Figs. 11–13 that the performance of the NNFADS system matched RT–FADS for most predictions of static pressure, dynamic pressure, and Mach number. Furthermore, NNFADS has been shown in Figs. 14 and 15 to perform close to the same level of accuracy as RT–FADS for the prediction of angles of attack and sideslip. These results, more than anything else, validate the performance of the NNFADS processor. The level of accuracy is definitely within the acceptable range for air data measurements. Additionally, the system developed here uses a set of explicit calculations to estimate the air data state, and the instabilities inherent to the RT–FADS aerodynamic models are avoided.

Fault detection and management schemes were tested by setting individual pressure signals out of range, one at a time. Before the FADS signals were processed, one of the pressures would be set to 100 psf. As a result, any of the configuration that included that pressure was excluded from the air data calculations. Over the range of data tested, there was no noticeable degradation in system performance. The NNFADS system is, therefore, currently tolerant to the failure of any single pressure signal. Work is currently under way to extend the fault management schemes to include multiple port failure and erroneous spikes in the pressure signals.

V. Conclusions

A fault-tolerant neural network-based algorithm was successfully developed for use with FADS systems. The performance of this system was found to match the aerodynamic model-based processor in the estimation of the air data parameters. The system was also found to be stable, with no major deviations in the air data estimates, throughout the domain for which it was developed. The fault detection and management schemes of the neural network-based algorithm were evaluated for the loss of any single input signal. These schemes successfully eliminated the effect of the lost signals without any reduction in the accuracy of the system.

Acknowledgment

This work was supported under Grant NCC 2-374 from NASA Dryden Flight Research Center, Edwards Air Force Base, California.

References

[1] Gracey, W., "Measurement of Aircraft Speed and Altitude," NASA RP 1046, May 1980.

[2] Whitmore, S. A., Davis, R. J., and Fife, J. M., "In-Flight Demonstration of a Real-Time Flush Airdata Sensing (RT-FADS) System," NASA TM 104314, Oct. 1995.

[3] Whitmore, S. A., Cobleigh, B. R., and Haering, E. A., "Architecture, Algorithms, and Calibration of the X-33 Flush Airdata Sensing (FADS) System," AIAA Paper 98-0201, Jan. 1998.

[4] Rohloff, T. J., and Catton, I., "Development of a Neural Network Flush Airdata Sensing System," *Proceedings of the 1996 ASME International Mechanical Engineering Congress and Exposition*, FED-Vol. 242, American Society of Mechanical Engineers, Fairfield, NJ, 1996, pp. 39–43.

[5] Rohloff, T. J., Whitmore, S. A., and Catton, I., "Air Data Sensing from Surface Pressure Measurements Using a Neural Network Method," *AIAA Journal*, Vol. 36, No. 11, 1998, pp. 2094–2101.

[6] Guo, T. H., and Nurre, J., "Sensor Failure Detection and Recovery by Neural Networks," *International Joint Conference on Neural Networks*, Vol. 1, 1991, pp. 221–226.

[7] Gallington, R. W., "Measurements of Very Large Flow Angles with Non-Nulling Seven-Hole Probe," *Aerodynamic Digest*, USAFA-TR-80-17, U.S. Air Force Academy, Spring/Summer 1980.

[8] Everett, K. N., Gerner, A. A., and Durston, D. A., "Seven-Hole Cone Probes for High Angle Flow Measurements: Theory and Calibration," *AIAA Journal*, Vol. 21, No. 7, 1983, pp. 992–998.

[9] Zilliac, G. G., "Calibration of Seven-Hole Probes for Use in Fluid Flows with Large Angularity," NASA TM 102200, Dec. 1989.

[10] Rediniotis, O. K., Hoang, N. T., and Telionis, D. P., "The Seven-Hole Probe: Its Calibration and Use," *Forum on Instructional Fluid Dynamics Experiments*, Vol. 152, June 1993, pp. 21–26.

[11] Anderson, J. D., *Modern Compressible Flow*, 2nd ed., McGraw–Hill, New York, 1990.

[12] Shapiro, A. H., *The Dynamics and Thermodynamics of Compressible Fluid Flow*, Vol. 1, Ronald, New York, 1953, pp. 83–88, 154.