JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 6, February 2009

Adaptive Algorithm for Classifying Launch and Recovery of the HH-60H Seahawk

Ankur Srivastava* and Andrew J. Meade†
Rice University, Houston, Texas, 77005

and

Jennifer Needham‡
Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139

DOI: 10.2514/1.37910

The pilot ratings from at-sea dynamic interface launch/recovery flight tests of the HH-60H Seahawk helicopter have been modeled satisfactorily by an adaptive, nonparametric classification scheme. Results were compared against those from three popular pattern classification tools and confirmed the authors' choice of algorithm. The resulting pilot rating classification model has also been used to evaluate the sensitivity to the 13 dependent parameters that include ship, aircraft, and environmental conditions.

Nomenclature

b_n	nth bias in the approximation
c_n	nth linear coefficient in the approximation
d	input dimension
f	arbitrary function
$F[v]$	proxy output of v
j	component index of $\boldsymbol{r}_{(n-1)}$ with the maximum absolute magnitude
n	number of bases
r_n	nth stage of the function residual
$\mathbb{R}$	real coordinate space
$\mathbb{R}^d$	d-dimensional vector space
s	number of training samples
$u(\xi)$	target function
$u_n^a(\xi)$	nth stage of the target function approximation
v	arbitrary function
$\{f, \ldots\}$	vector of the elements of f
$\langle f \rangle$	sum of the elements of f

Received 6 April 2008; revision received 12 December 2008; accepted for publication 15 December 2008.

* Graduate Student, Mechanical Engineering, Mail Stop 321, ankur.srivastava@rice.edu, AIAA Member.
† Professor, Mechanical Engineering, Mail Stop 321, meade@rice.edu, AIAA Associate Fellow.
‡ Graduate Student, Aeronautics and Astronautics, 77 Massachusetts Avenue, Cambridge, jneedham@mit.edu.

$\langle f, v\rangle_D$	$\sum_i^s (f_i v_i)$ = discrete inner product of f and v
$\lvert f\rvert$	absolute value of f
$\lVert f\rVert_2$	$\left(\int_\Omega f^2 d\xi\right)^{1/2} = L_2$ norm of f
$\lVert f\rVert_{2,D}$	$\left(\sum_i^s (f_i)^2\right)^{1/2}$ = discrete L_2 norm of f

Greek symbols

α	width parameter of Gaussian basis function
β	set of nonlinear optimization parameters
ε	objective function
θ	vector of parameters of an arbitrary data fitting model
λ	regularization parameter
$\Lambda[f]$	smoothing functional of f
ξ	d-dimensional input of the target function
ξ_i	ith sample input of the target function
ξ_n^*	sample input with component index j^* at the nth stage
σ	width parameter of Gaussian radial basis function
τ	user specified tolerance
ϕ	basis function
ρ	input feature sensitivity

Subscripts

i, j, k	dummy index
j^*	component index of $\boldsymbol{r}_{(n-1)}$ with the maximum magnitude
n	associated with the number of bases
s	associated with the number of samples

Superscript

d	associated with the input dimension

I. Introduction

SHIPBOARD helicopter flight operations constitute a challenging area of research. There are many factors that affect a helicopter pilot's ability to conduct safe shipboard flight operations. These include the size, shape, and location of the flight deck, its proximity to other shipboard structures, and often unpredictable ship motion or turbulent air-wake effects. All of these factors tend to increase the difficulty of conducting successful shipboard launch and recovery of rotorcraft. To evaluate the ability of helicopters to operate aboard various ship classes, and to determine any limitations that might be associated with such operations, the United States Navy (USN) conducts shipboard helicopter flight tests also known as dynamic interface (DI) tests. The most significant product of a typical DI test program for launch and recovery is the development of shipboard rotorcraft operational envelopes [1] as illustrated in Fig. 1. The arcs shown in this figure represent the speed of wind in knots and radial lines are drawn to show the direction of the wind relative to the ship. The shaded area in the envelope shows operating limits at night.

DI-derived operational envelopes summarize the relative wind and/or ship motion limits that will allow consistently safe shipboard rotorcraft operations by USN pilots as measured by the pilot rating scale (PRS) shown in Table 1. The greater the quantity and quality of DI data points, the greater the resolution of the envelope. This in turn provides increased helicopter operational flexibility and effectiveness. Likewise, a high-resolution DI envelope provides the ability to conduct safe rotorcraft launch/recovery operations from a wide variety of ship course/speed combinations, thereby increasing the ship's tactical flexibility.

However, DI tests require considerable effort and expense. Each DI test program typically consists of pretest planning and coordination, land-based buildup, at-sea shipboard testing, post-test data analysis, and finally, test debriefing/reporting. For example, a single at-sea aircraft DI test period lasting 4–5 days typically requires a test team of 4 pilots, 2 aircrewmen, 4 test engineers, and 5 maintenance personnel. During shipboard test operations, the

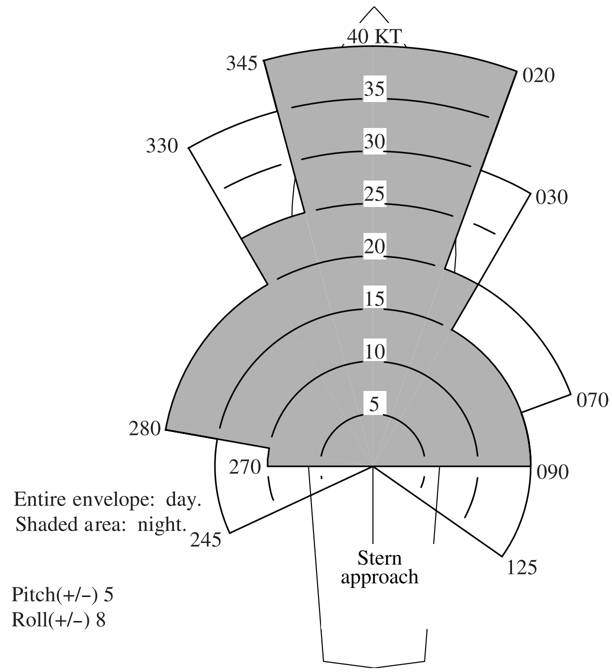

Fig. 1 DI launch/recovery operational envelope [1].

Table 1 Pilot rating scale [1]

PRS # - Pilot effort	Rating description
1-Slight	No problems; minimal pilot effort required to conduct consistently safe shipboard evolutions under these conditions.
2-Moderate	Consistently safe shipboard evolutions possible under these conditions. These points define fleet limits recommended by the Naval Air Warfare Center Aircraft Division (NAWCAD) Patuxent River, MD.
3-Maximum	Evolutions successfully conducted only through maximum effort of experienced test pilots using proven test methods under controlled test conditions. Loss of aircraft or ship system likely to raise effort beyond capabilities of average fleet pilot.
4-Unsatisfactory	Pilot effort and/or controllability reach critical levels. Repeated safe evolutions by experienced test pilots are not probable even under controlled test conditions.

DI test team conducts multiple flight tests under a wide range of ship conditions, aircraft configurational conditions, and environmental conditions. Any innovative flight or analytical technique is welcome that can the reduce time and expense of these DI test programs. With that in mind, the current authors believe machine learning tools can find considerable use in DI tests.

The problem of predicting the rating for a flight test falls under the category of classification problems in machine learning research. Classification has been widely used in flight envelope prediction [2] and numerous aircraft applications [3,4]. Popular classification algorithms include support vector machines (SVM) [5] and k-nearest neighbor (k-NN) [6,7] classifiers. These classifiers require the use of user-determined control parameters and/or kernel hyperparameters. The user must find the optimum values of the control parameters for the entire data set either by cross-validation or grid search approach. In such approaches the data must be used to generate numerous randomly selected subsets for training and testing. The values of the control parameters must then be optimized on each of these testing subsets and the optimum control parameters averaged.

We believe that for the purpose of this and other types of at-sea Navy flight tests, a multi-dimensional classification tool should address the previously mentioned hyper-parameter selection problem, operate as easily on high-dimensional data as it does on low, and provide the user an assurance that the classifier will give its best performance on an unseen test set. The tool should also require a minimum of storage and require as little user interaction as possible. Finally, for the purposes of generalization and data compression, the scheme should construct an accurate approximation that uses as few basis functions as possible, preferably less than the number of sample points. We believe we have developed just such a classification tool for Navy DI tests, which we call sequential function approximation (SFA). SFA was originally introduced to solve differential equations [8]. It was later used to provide kernel based solutions to regression problems [9]. In this work SFA was adapted for the solution of a classification problem.

We have two objectives in this paper. Our first objective is to determine whether 13 dependent parameters that include ship, helicopter, and environmental conditions can be accurately and efficiently mapped to empirical pilot ratings of launch and recovery. It is thought that this mapping can be used as a surrogate for the human pilot evaluations of the HH-60H in future computer simulations used to develop further the DI launch/recovery operational envelope. Our second objective is to determine if the pilot rating model constructed from at-sea test runs can be evaluated for input parameter sensitivity. This sensitivity analysis will help to determine whether the pilot rating model can be used to eliminate the need to sample certain test parameters and reduce the complexity of the flight tests.

In Sec. II we discuss the difficulty of classification problems in machine learning and the development of the SFA algorithm. In Sec. III the classification scheme is used to model the pilot rating of an actual at-sea DI launch/recovery flight test of the HH-60H Seahawk. For the sake of completeness, the performance of the scheme is compared with that of popular classifiers to confirm our choice of tool. The constructed launch/recovery pilot rating model is then evaluated to determine its sensitivity to the 13 parameters that include ship, aircraft, and environmental conditions. Conclusions and future work are given in Sec. IV.

II. Approach

A. Classification

In statistics, classification of a data set involves the construction of a discriminant function, which predicts the labels of the unlabeled data set as accurately as possible. In constructing the discriminant function one makes the assumption that all the samples are independent and identically distributed from the same probability distribution function. Unfortunately, this problem is not trivial for a real world data set because the probability distribution from which the samples are drawn is unknown. A number of statistically based and function approximation based classification tools exist in the literature for real-world problems and are discussed and compared in detail in [10]. Scattered-data function approximation techniques in machine learning are also used to model either the discriminant function directly or the multidimensional manifold that separates the classes (e.g., separating hyperplanes from the classic form of SVM).

The SFA tool used in this paper models the discriminant function u directly. We note that a continuous d-dimensional function can be arbitrarily well approximated by a linear combination of basis functions ϕ. Because scattered data approximation is an ill-posed problem, we can use concepts from regularization and construct the discriminant function by minimizing the following objective function

$$\varepsilon = \sum_{i=1}^{s} \|F[u(\xi_i)] - F[u^{\mathrm{a}}(\xi_i)]\|^2 + \lambda \Lambda[u^{\mathrm{a}}(\xi)]$$

The objective function ε accounts for two terms, the error to approximate the true surface realized by the discrete experimental data and the smoothness of the approximation. The regularization parameter λ controls the trade-off between the interpolation error and the smoothness of the solution. We use the approach outlined by Poggio and Girosi [11] who suggested in the solution of ill-posed problems that, “If nothing else is known about a high dimensional function to be approximated, the only option may be to assume a high degree of smoothness.” We interpret this as constructing a smooth continuous approximation that satisfies the regularization functional identically with as few basis functions as possible. We also assume the discriminant itself is smooth but can only be measured in discrete values (the proxy function). Consequently, we only require that the approximation fit the scattered data within accepted tolerance, i.e.

$$\sum_{i=1}^{s} \| F[u(\xi_i)] - u^{\mathrm{a}}(\xi_i) \|^2 \leqslant s^2 \tau^2$$

$u(\xi_i)$ is the true human label of the data, $u^{\mathrm{a}}(\xi_i)$ is the approximated solution. The sign function is a good example of the proxy function F [ ]. To guarantee smoothness second derivative of the approximation error may be chosen for Λ.

B. Sequential Function Approximation

SFA was developed from mesh-free finite element research but shares similarities with the Boosting [12] and Matching Pursuit [13] algorithms. We start our approximation of u using the Gaussian radial basis function (RBF) ϕ

$$u_n^{\mathrm{a}}(\xi) = \sum_{i=1}^{n} c_i(\phi(\xi, \beta_i) + b_i) \tag{1}$$

with

$$\phi(\xi, \beta_i) = \exp\left[-\frac{(\xi - \xi_i^*) \cdot (\xi - \xi_i^*)}{\sigma_i^2} \right] = \exp\left[\ln[\alpha_i](\xi - \xi_i^*) \cdot (\xi - \xi_i^*) \right] \quad \text{for } 0 < \alpha_i < 1$$

where $\xi \in \mathbb{R}^{\mathrm{d}}$ represents the d-dimensional data point, $c_i, b_i \in \mathbb{R}$ are the linear coefficients and bias, and $\beta_i = \{\alpha_i, \xi_i^*\} \in \mathbb{R}^{d+1}$ represents the ith basis function parameters, the width α_i and the center $\xi_i^* \in \mathbb{R}^{\mathrm{d}}$. We write the RBF as Eq. (1) to set up the optimization problem for α_i as a bounded nonlinear line search instead of an unconstrained minimization problem. The basic principles of our greedy algorithm are motivated by the similarities between the iterative optimization procedures by Jones [14,15] and Barron [16] and the method of weighted residuals (MWR), specifically the Galerkin method [17]. We can write the function residual r at the nth stage of approximation as the following

$$\begin{aligned} r_n &= F[u(\xi)] - u_n^{\mathrm{a}}(\xi) \\ &= F[u(\xi)] - u_{n-1}^{\mathrm{a}}(\xi) - c_n(\phi(\xi, \beta_n) + b_n) \\ &= r_{n-1} - c_n(\phi(\xi, \beta_n) + b_n) = r_{n-1} - c_n(\phi_n + b_n) \end{aligned} \tag{2}$$

Using the Petrov–Galerkin approach, we select a coefficient $c_n$that will force the function residual to be orthogonal to the basis function and b_n using the discrete inner product given by Eq. (3)

$$r_n \cdot (\phi_n + b_n) = \sum_{j=1}^{s} r_n(\xi_j)(\phi_n(\xi_i) + b_n) = \langle r_n, (\phi_n + b_n) \rangle_D = -\left\langle r_n, \frac{\partial r_n}{\partial c_n} \right\rangle_D = -\frac{1}{2} \frac{\partial \langle r_n, r_n \rangle_D}{\partial c_n} = 0 \tag{3}$$

which is equivalent to selecting a value of c_n that will minimize $\langle r_n, r_n \rangle_D$ or

$$c_n = \frac{(s\langle \phi_n, r_{n-1} \rangle_D - \langle r_{n-1} \rangle_D \langle \phi_n \rangle_D)}{(s\langle \phi_n, \phi_n \rangle_D - \langle \phi_n \rangle_D \langle \phi_n \rangle_D)} \tag{4}$$

with

$$b_n = \frac{(\langle r_{n-1} \rangle_D \langle \phi_n, \phi_n \rangle_D - \langle \phi_n \rangle_D \langle \phi_n, r_{n-1} \rangle_D)}{(s\langle \phi_n, r_{n-1} \rangle_D - \langle r_{n-1} \rangle_D \langle \phi_n \rangle_D)} \tag{5}$$

The discrete inner product $\langle r_n, r_n\rangle_D$, which is equivalent to the square of the discrete L_2 norm, can be re-written, with the substitution of Eqs. (4) and (5), as

$$\sum_{j=1}^{s} r_n(\xi_j)r_n(\xi_j) = \|r_n\|^2_{2,D} = \langle r_n, r_n\rangle_D$$
$$= \left\langle r_{n-1} - \frac{\langle r_{n-1}\rangle_D}{s}, r_{n-1} - \frac{\langle r_{n-1}\rangle_D}{s}\right\rangle_D$$
$$\times\left(1 - \frac{\left\langle r_{n-1} - \frac{\langle r_{n-1}\rangle_D}{s}, \phi_n - \frac{\langle \phi_n\rangle_D}{s}\right\rangle^2_D}{\left\langle \phi_n - \frac{\langle \phi_n\rangle_D}{s}, \phi_n - \frac{\langle \phi_n\rangle_D}{s}\right\rangle_D \left\langle r_{n-1} - \frac{\langle r_{n-1}\rangle_D}{s}, r_{n-1} - \frac{\langle r_{n-1}\rangle_D}{s}\right\rangle_D}\right) \tag{6}$$

Recalling the definition of the cosine given by Eq. (7), using arbitrary functions f and v and the discrete inner product

$$\cos(\theta) = \frac{\langle f, v\rangle_D}{\langle f, f\rangle^{1/2}_D \langle v, v\rangle^{1/2}_D} \tag{7}$$

Eq. (6) can be written as

$$\|r_n\|^D_{2,D} = \left\langle r_{n-1} - \frac{\langle r_{n-1}\rangle_D}{s}, r_{n-1} - \frac{\langle r_{n-1}\rangle_D}{s}\right\rangle \sin^2(\theta_n) = \left(\|r_{n-1}\|^D_{2,D} - \frac{\langle r_{n-1}\rangle^2_D}{s}\left[2 - \frac{1}{s}\right]\right)\sin^2(\theta_n) \tag{8}$$

where θ_n is the angle between ϕ_n and r_{n-1} because $\langle r_{n-1}\rangle_D/s$ and $\langle \phi_n\rangle_D/s$ are scalars. With Eq. (8) we note that $\| r_n \|^2_{2,D} < \| r_{n-1} \|^2_{2,D}$ as long as $\theta_n \neq \pi/2$, which is a very robust condition for convergence. By inspection, the minimum of Eq. (8) is $\theta_n = 0$, implying Eq. (9)

$$c_n(\phi(\xi_i, \beta_n) + b_n) = r_{n-1}(\xi_i) \text{ for } i = 1, \ldots, s \tag{9}$$

Therefore, to force $\| r_n \|^2_{2,D} \to 0$ with as few stages n as possible, a low dimensional function approximation problem must be solved at each stage. This involves a bounded nonlinear minimization of Eq. (6) to determine the two variables $\alpha_i (0 < \alpha_i \leqslant 1)$ and index $j^*(\xi^*_n = \xi_j \in \mathbb{R}^d)$ for the basis function center taken from the training set. The dimensionality of the nonlinear optimization problem is kept low because we are solving only one basis at a time.

C. Unknown Parameters

Any regression or classification problem can be treated as a function approximation problem. In this work we constructed our approximation as a linear sum of Gaussian radial basis functions given by Eq. (1). The approximation is constructed via a four dimensional optimization problem for the coefficients (c_n), the bias (b_n), the width parameter (α_n), and the center of the basis function (ξ^*_n) at each stage. Incorporating the MWR we calculated the optimum value of the coefficients and bias given by Eqs. (4) and (5). This left us with a two-dimensional optimization problem for the center and the width at each stage.

To reduce computational time and expense we used a heuristic rather than optimization to locate the center of the RBF at each stage. For each basis function the center was chosen to be the training vector that corresponds to the maximum absolute value of the residual at that stage. The rationale behind choosing this approach is to place the basis function at peaks in the residual so that the maximum residual errors are taken care of first. It has proven to be a good approach for a number of regression and classification problems tested in [9] and [18]. However, developing a computationally efficient way of finding optimal basis centers is a part of the future research. We are then left with a one-dimensional optimization problem given by Eq. (6) for the width of the basis function which we solve at each stage in the algorithm.

D. Implementation of the Algorithm

Although our SFA scheme allows the basis center (ξ_n^*) to be located anywhere in $\mathbb{R}^d$, the practical application to problems with multiple inputs constrains the centers to the set of sample points$\{\xi_1, \ldots, \xi_s\}$. At each stage we determine ξ_n^* such that $|r_{n-1}(\xi_n^*)| = \max |r_{n-1}|$. The remaining optimization variable α_n is continuous and constrained to $0 < \alpha_n < 1$.

A benefit to using RBFs is that in practical applications we can ignore the denominator in the discrete inner product formulation of Eq. (6). As a result, the determination of β_n requires only

$$\min_{\beta_n} \left[-\left\langle r_{n-1} - \frac{\langle r_{n-1}\rangle_D}{s}, \phi_n - \frac{\langle \phi_n \rangle_D}{s} \right\rangle_D^2 \right] \tag{10}$$

The algorithm terminates when $\max |r_n| \leqslant \tau$, where τ is the tolerance desired by the user. For binary classification where $F[u] = +1$ or -1, we set $\tau = 1.0$.

In this work multi-class classification problems are tackled by applying several binary classifiers in parallel. Approaches to combining binary classifiers include the *one vs. one* combination, the *one vs. all* pairing, and the error correcting output coding. There is no best approach to combining binary classifiers. Even though *one vs. all* has its own shortcomings, this approach was selected by the authors because of its simplicity and acceptable accuracy. Rifkin and Klautau [19] compared the *one vs. all* approach with other existing methods and concluded that *one vs. all* classification will produce results as good as any other approach if the underlying classifiers are well tuned. Comparison of other methods used to combine binary classifiers is reserved for a separate study.

To implement the SFA algorithm the user takes the following steps:

1. Initiate the algorithm with the labels of the training data $r_0 = \{F[u(\xi_1)], \ldots, F[u(\xi_s)]\}$.
2. Search the components of r_{n-1} for the maximum magnitude. Record the component index j^*.
3. $\xi_n^* = \xi_{j^*}$.
4. With ϕ_n centered at ξ_{j^*}, minimize Eq. (10) initializing the optimization parameter with $\alpha_n \cong 0$.
5. Calculate the coefficient c_n and b_n from Eqs. (4) and (5) respectively.
6. Update the residual vector $r_n = r_{n-1} - c_n(\phi_n + b_n)$. Repeat steps 3 to 6 until the termination criterion $\max |r_n| \leqslant 1.0$ has been met.
7. Use the constructed binary classifier to predict on the test set.
8. Repeat steps 1 to 7 for remaining classes.
9. Use *one vs. all* scheme to obtain the final prediction on the test set.

Our binary method is linear in storage with respect to s because it needs to store only $s + sd$ vectors to compute the residuals: one vector of length s (r_n) and s vectors of length $d(\xi_1, \ldots, \xi_s)$ where s is the number of samples and d is the number of dimensions. To complete the SFA model requires two vectors of length $n(\{c_1, \ldots, c_n\}$ and $\{b_1, \ldots, b_n\})$ and n vectors of length $n + 1(\beta_1, \ldots, \beta_n)$.

III. Results

A total of 369 at-sea dynamic interface tests were conducted by the US Navy. Figure 2 shows the variation of the DI test inputs for all the tests. This figure does not include the Ship Type input because only two ships were used during the entire testing. The input abbreviations are explained in Table 2. Pilot ratings for launch and recovery were noted for each test. The mapping of 13 dynamic flight test parameters to the pilot ratings of the launch and recovery of a HH-60H Seahawk helicopter was investigated using the SFA algorithm.

A. Predictive Accuracy on Launch and Recovery Data Sets

Launch/recovery pilot ratings were mathematically modeled as a function of thirteen dynamic interface parameters by SFA using Gaussian radial basis functions. The model was trained on one-half of the available points and tested on the remainder. Percentage accuracy is calculated by Eq. (11) shown below

$$\text{Percentage accuracy} = \left(\frac{Ntest - mtest}{Ntest} \right) \times 100\% \tag{11}$$

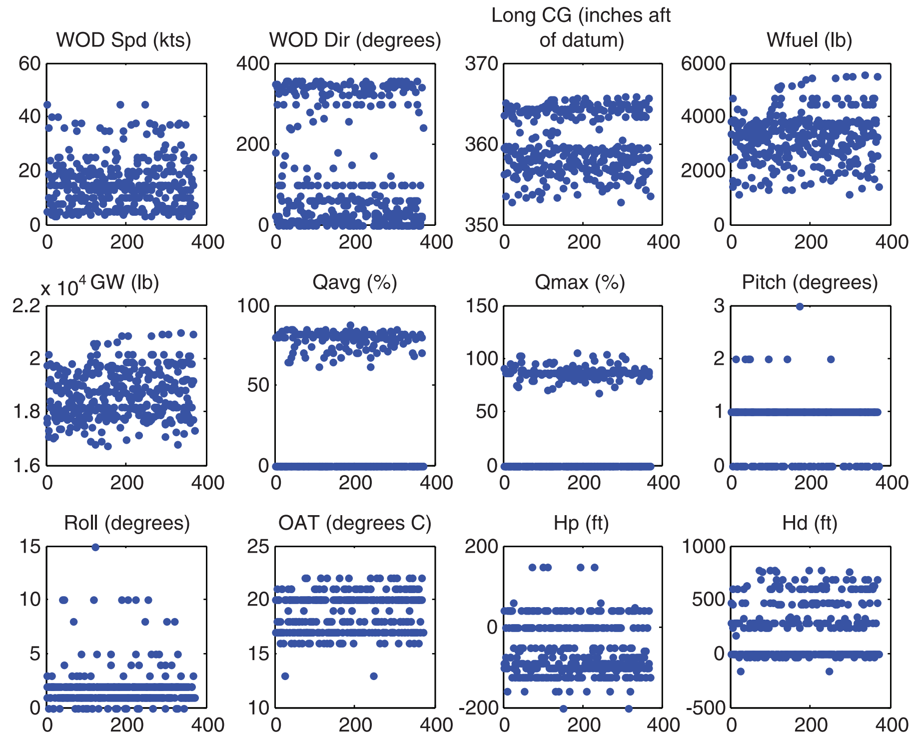

Fig. 2 DI test inputs for 369 tests.

Table 2 HH-60H DI flight test inputs [1]

Input index	Abbreviation	Definition (units)
1	Ship type	USN ship type: DD 967 (1), DDG 61 (2)
2	WOD Spd	Wind over deck speed. Relative wind speed (kts)
3	WOD Dir	Wind over deck direction. Relative wind direction (degrees)
4	Long CG	Longitudinal CG station of helicopter. Length aft of datum (inches)
5	Wfuel	Weight of fuel aboard helicopter (lb)
6	GW	Gross weight of helicopter (lb)
7	Qavg	Average hover torque during evolution (%)
8	Qmax	Maximum torque required during evolution (%).
9	Pitch	Pitch angle of ship during evolution (°)
10	Roll	Roll angle of ship during evolution (°)
11	OAT	Outside air temperature (°C)
12	Hp	Pressure altitude (ft)
13	Hd	Density altitude (ft)

where $Ntest =$ number of points in the test set and $mtest =$ number of misclassifications. Random launch and recovery training and test sets were generated using Matlab's *randperm* function. Using the *randperm* function 369 distinct random integers were generated between 1 and 369. The first 185 integers were used as indices that gave the training set and the remaining integers were used to select the test set. In this way twenty random training and test sets were generated. Using the *one vs. all* procedure, the SFA algorithm was implemented using the MATLAB programming environment on a Windows-configured PC with a Pentium 4 2.66 GHz processor and 1.0 GB of RAM. The SFA algorithm's mean and standard deviation of percentage classification accuracy were 81 ± 2 for launch and 73 ± 3 for recovery.

For the sake of completeness, we compared the performance of SFA on the pilot rating data against the optimal performance of the popular classification schemes mentioned earlier. We detail these classification algorithms below:

1. **LS-SVM**: LS-SVMlab1.5, a Matlab/C toolbox, was used to implement the least squares support vector machines [5]. The basis functions used to construct the approximation are also known as support vectors in the area of machine learning research, hence the name of the algorithm. The *one vs. all* scheme was used to combine different binary classifiers and perform multi-class classification. It uses two user-controlled parameters, the regularization parameter (γ) and the width of the Gaussian radial basis function (σ).
2. ***k*-NN**: SPIDER [7], a MATLAB based machine learning toolbox was used to implement k-nearest neighbor algorithm on this problem. This algorithm also uses *one vs. all* approach to construct multi-class classifier. Gaussian radial basis functions were used to compute the distances between training/test points. It uses two user-controlled parameters, the number of nearest neighbors (k) and the width of the RBF (σ).
3. **EDC**: A simple Euclidean distance classifier [8] with one nearest neighbor and the entire training set defined as the neighborhood was also used to obtain classification accuracies on the Launch and Recovery data sets. EDC is different from k-NN because unlike k-NN, EDC does not use a RBF to compute the distances between the training and test points. Also, EDC does not combine binary classifiers in the solution of multi-class problems.

To obtain the optimal classification accuracy of LS-SVM and k-NN an exhaustive grid search for the user-controlled parameters of the algorithms was done using the *one vs. all* procedure mentioned at the end of Sec. II D. These optimal parameter values are given in Table 3. Like SFA, the use of EDC included only loading of training and test sets because it has no user-determined parameters. All algorithms were implemented using the MATLAB programming environment on a Windows-configured PC with a Pentium 4 2.66 GHz processor and 1.0 GB of RAM.

Results in Table 4 show that the mean classification accuracy of SFA and EDC on the launch and recovery test sets is among the highest of the classifiers and display relatively small standard deviations. The remainder of this subsection will concentrate on comparing SFA and EDC.

Although EDC has shown to be attractive for our pilot rating classification problem, the algorithm may run into difficulty as the number of input dimensions is increased. That is, the EDC breaks down in higher dimensions because training points sparsely populate the input space. If s data points sufficiently fill up one dimension, s^d are needed to fill up a d-dimensional input space. The reduced density of training points decreases the mean prediction accuracy of the classifier and increases its variance. Authors in [10] demonstrated this effect using several simulated examples and present a more complete understanding of this problem. The convergence rate of SFA using RBFs, however, does not depend explicitly on the dimensionality of the approximation [20]. It sidesteps the "curse of dimensionality."

Table 3 Optimal control and kernel hyperparameters used by LS-SVM and *k*-NN on launch and recovery pilot rating test sets using half of the data set to train the classifier

	LS-SVM		k-NN	
Data sets	γ	σ	k	σ
Launch	107.82	0.11	11	29.77
Recovery	215.66	0.11	1	15.48

Table 4 Mean and standard deviation of percentage accuracies on twenty random launch and recovery pilot rating test sets using half of the data set to train the classifier

Classifier	Launch (%)	Recovery (%)
LS-SVM	71 ± 5	69 ± 3
k-NN	71 ± 8	68 ± 14
EDC	76 ± 2	73 ± 3
SFA	81 ± 2	73 ± 3

It is accepted within the machine learning community that sparse network topologies provide redundancy and robustness in the approximation of data. Redundancy in the constructed approximation improves its predictive ability while robustness allows the approximation to perform within certain bounds of its nominal performance in the presence of bounded uncertainty. The EDC needs to store all of the training set (s vectors of length d) to predict the class of a test point resulting in a memory storage requirement of sd. SFA needs to store only the basis functions and the function parameters to predict the class of a test point. More precisely, SFA needs memory storage for n vectors of length d, n coefficients, n width parameters, and a bias variable, which results in a total of $2n + nd + 1$. In predicting the launch pilot ratings SFA used an average 14% of the training data as RBF centers to construct the classifier and 20% of the training data as RBF centers. Inputting the appropriate numbers of $n < 0.2s$, $s = 185$ and $d = 13$, we can say that SFA used less than 23% of the memory required by the EDC.

SFA was selected for modeling the pilot rating problem because of its high accuracy, low storage during learning, low number of basis functions required for prediction, and minimum user interaction. For a more detailed evaluation of SFA, readers can refer to [18] where SFA was compared against other popular classification algorithms over many real-world and synthetic data sets.

B. Input Sensitivity

One of the tasks of this work was to find the inputs of Table 2 that display low sensitivity. Determining the least sensitive input components could help the DI test engineers to focus resources on fewer parameters and accelerate through the test matrix. In the field of classification the problem of finding the most sensitive inputs is known as feature selection. In the current work, a partial derivative method was chosen because the authors view the problem of classification as a function approximation problem. Partial derivative of the approximating function, given by Eq. (1), was calculated with respect to each input dimension as shown in Eq. (12).

$$\frac{\partial u}{\partial \xi_{k,j}} = \sum_{i=1}^{n} 2\left|\xi_{k,j} - \xi^*_{i,j}\right| c_i \phi_{i,k} \ln(\alpha_i) \text{ where } i = 1, 2, 3, \ldots, n,\ j = 1, 2, 3, \ldots, d \text{ and } k = 1, 2, 3, \ldots, s \tag{12}$$

The input sensitivities ρ_j were determined by summing the squares of the derivatives over the number of training points as shown in Eq. (13).

$$\rho_j = \frac{1}{s}\sum_{k=1}^{s}\left(\frac{\partial u}{\partial \xi_{k,j}}\right)^2 \tag{13}$$

This was done once for each binary classifier and an average taken to get the final input sensitivities. The input sensitivity values were calculated after training and testing was completed over one set of data sets. To eliminate any bias resulting from the choice of the training and testing set, the sensitivity calculations were repeated 20 times over randomly chosen training and test points from the available data. The sensitivity results shown in Figs. 3a and 3b are computed in an average sense over twenty randomly chosen training and test sets from the available data.

Based on the available data, we conclude the following:

1. Launch/recovery pilot ratings of HH-60H Seahawk differ little between DD 967 and DDG 61.
2. Longitudinal CG station of the helicopter does not affect launch or recovery.
3. Pitch angle of the ship below 2° does not affect recovery.
4. Roll angle of the ship below 15° does not affect recovery.
5. Outside air temperature between 13 and 22 °C does not affect launch or recovery. Outside air temperature sensitivity values are low because the outside air temperature varied in a smaller range. Most of the values vary between 16 and 22° with just two measurements of 13 °C. However, there was significant variation in the pressure and density altitude corresponding to this variation in the outside air temperature.

Variation of these inputs can be neglected in future DI tests of this aircraft, with time and resources devoted to the remaining inputs. In our work, the three least significant inputs (1, 4, and 11) were eliminated and training and testing was repeated over the reduced data set. It was verified that the remaining inputs achieved similar mean percentage accuracies as those shown in Table 2 for both launch and recovery pilot rating data set. This exercise verified that elimination of these features did not decrease the prediction accuracies of SFA over launch and recovery pilot rating test sets.

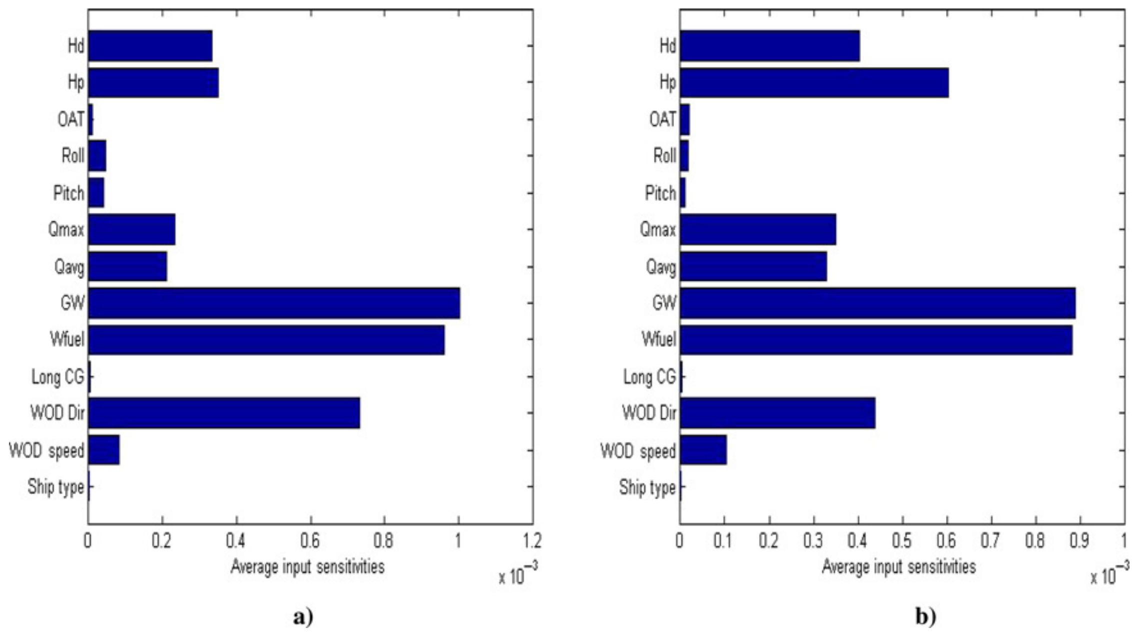

Fig. 3 Average input sensitivities in the prediction of a) launch pilot ratings and b) recovery pilot ratings.

IV. Conclusions

This paper has demonstrated that classification tools from the field of machine learning can model the pilot ratings from the launch and recovery test of a U.S. Navy helicopter. We propose that the mathematical model can be used as a surrogate for the human pilot evaluations of the HH-60H in future computer simulations used to further develop the DI launch/recovery operational envelope. In particular, a nonparametric and adaptive classification tool accurately and efficiently mapped 13 dependent parameters, which included ship, helicopter, and environmental conditions, to empirical pilot ratings of HH-60H launch and recovery. Results were compared against three popular classification tools (LS-SVM, *k*-NN, and EDC) from the machine learning literature and confirmed our choice of algorithm. This work has also demonstrated that the pilot rating model constructed from at-sea launch/recovery tests can be evaluated for input parameter sensitivity. It is believed that with this ability ship-based engineers can plan remaining tests with an estimate on how fine the test parameter changes should be made. This sensitivity analysis may also help determine whether the pilot rating model can be used to eliminate the need to sample certain test parameters and thereby reduce the complexity of the flight tests.

Acknowledgments

Support for this work was provided by the NASA Ames Research grant NCC-2-8077 and NASA Cooperative Agreement No. NCC-1-02038.

References

[1] Williams, S. L., and Long, K. R., "Dynamic Interface Flight Tests and the Pilot Rating Scale," *Proceedings of the 53rd American Helicopter Society Annual Forum*, American Helicopter Society, Alexandria, VA, April 29–May 1, 1997.

[2] Yavrucuk, I., Prasad, J. V. R., and Calise, A. J., "Adaptive Limit Detection and Avoidance for Carefree Maneuvering," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA, Reston, VA, AIAA Paper 2201-4003, 2001.

[3] Letourneau, S., Famili, F., and Matwin, S., "Data Mining for Prediction of Aircraft Component Replacement," *IEEE Intelligent Systems Journal*, Vol. 14, No. 6, Nov.–Dec. 1999, pp. 59–66.
doi: 10.1109/5254.809569

[4] Wroblewski, D., Grabill, P., Berry, J., and Branhof, R., "Neural Network System for Helicopter Rotor Smoothing," *IEEE Aerospace Conference Proceedings*, IEEE Publications, Piscataway, NJ, Vol. 6, 2000, pp. 271–276.

[5] Suykens, J. A. K., Gestel, T. V., De Brabanter, J., De Moor, B., and Vandewalle, J., *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.

[6] Dasarathy, B. V., *Nearest Neighbor Pattern Classification Techniques*, IEEE Computer Society Press, 1991.

[7] Weston, J., Elisseeff, A., Bakir, G., and Sinz, F., Software available electronically at http://www.kyb.mpg.de/bs/people/spider/main.html. Max Planck Institute for Biological Cybernetics.

[8] Kokkolaras, M., "Utilizing Parallel Optimization in Computational Fluid Dynamics," PhD Thesis, Mechanical Engineering Department, Rice University, November 1997.

[9] Meade, A. J., "Application of Scattered Data Approximation to a Rotorcraft Health Monitoring Problem," *Journal of Aerospace Computing, Information, and Communication*, Vol. 1, No. 4, 2004, pp. 175–188.
doi: 10.2514/1.1978

[10] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning*, Springer Series in Statistics, 2001.

[11] Poggio, T., and Girosi, F., "Networks for Approximation and Learning," *Proceedings of the IEEE*, Vol. 78, No. 9, 1990, pp. 1481–1497.
doi: 10.1109/5.58326

[12] Freund, Y., "Boosting a Weak Learning Algorithm by Majority," *Information and Computation*, Vol. 121, No. 2, 1995, pp. 256–285.
doi: 10.1006/inco.1995.1136

[13] Mallat, S. G., and Zhang, Z., "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, December 1993.
doi: 10.1109/78.258082

[14] Jones, L. K., "Constructive Approximations for Neural Networks by Sigmoidal Functions," *Proceedings of the IEEE*, Vol. 78, No. 10, 1990, pp. 1586–1589.
doi: 10.1109/5.58342

[15] Jones, L. K., "A Simple Lemma on Greedy Approximation in Hilbert Space and Convergence Rates for Projection Pursuit Regression and Neural Network Training," *The Annals of Statistics*, Vol. 20, No. 1, 1992, pp. 608–613.
doi: 10.1214/aos/1176348546

[16] Barron, A. R., "Universal Approximation Bounds for Superpositions of a Sigmoidal Function," *IEEE Transactions on Information Theory*, Vol. 39, No. 3, 1993, pp. 930–945.
doi: 10.1109/18.256500

[17] Fletcher, C. A. J., *Computational Galerkin Methods*, Springer-Verlag, New York, NY, 1984.

[18] Srivastava, A., "A Greedy Algorithm for Learning Pilot Ratings from Helicopter Shipboard Dynamic Interface Tests," Masters Thesis, Mechanical Engineering Department, Rice University, April 2007.

[19] Rifkin, R., and Klautau, A., "In Defense of One-Vs-All Classification," *Journal of Machine Learning Research*, Vol. 5, 2004, pp. 101–141.

[20] Meade, A. J., and Zeldin, B. A., "Approximation Properties of Local Bases Assembled from Neural Network Transfer Functions," *Mathematical and Computer Modeling*, Vol. 28, No. 9, 1997, pp. 43–62(20).
doi: 10.1016/S0895-7177(98)00144-7

Christopher Rouff
Associate Editor