

Estimator Design in Jet Engine Applications*

Manfredi Maggiore[†], Raúl Ordóñez[†], Kevin Passino[†], and Shrider Adibhatla[‡]

[†] Department of Electrical Engineering, The Ohio State University 2015 Neil Avenue, Columbus, OH, 43210-1272

[‡] GE Aircraft Engines, One Neumann Way MD BBC-5, Cincinnati, OH 45215-6301

Abstract

Jet engines are nonlinear dynamical systems for which an exact mathematical model cannot be used for estimator design, because it is either not available or so complex that it does not fit the necessary assumptions. Thus, classical analytical tools for studying standard system properties like observability, which is very important in estimator design, cannot be directly applied. Generally, for practical jet engine applications, the designer faces two closely related problems: first, given a non-measurable parameter, find the minimal set of estimator inputs that facilitates achieving a satisfactory estimation performance (input selection); second, given a predetermined set of inputs, derive an “observability” measure that characterizes the *estimation feasibility* of a specific non-measurable parameter. In this paper, techniques for solving these two problems are developed and applied to estimator design for jet engine thrust, stall margins, and an unmeasurable state.

1 Introduction

Thrust regulation is often the primary objective in jet engine control; this quantity, however, cannot be measured, so the designer is forced to regulate closely related measurable variables such as the rotor speeds or pressure ratios. The resulting control designs must be conservative to ensure delivery of guaranteed thrust levels in the presence of engine-to-engine manufacturing differences and engine deterioration. The conservative nature of the control design results in operating the engine in a less efficient manner (e.g., at higher temperatures using more fuel) that shortens its life. A high quality thrust estimator can serve as a “virtual sensor” for thrust, allowing for more direct control over its value and resulting in less conservative designs that could lengthen engine life and improve its operating efficiency.

Another problem faced in jet engine control is how to avoid rotating stall [1]. Several unmeasurable “stall margin” variables are generally introduced to characterize how close the system is to a stall condition.

*This work was supported by NASA Lewis Research Center, Grant NAG3-2084. We wish to thank Dr. Ten-Huei Guo at NASA Lewis Research Center for his support on this project. Please address correspondence to K. Passino (k.passino@osu.edu)

Then, the designer constructs control laws so that these variables stay within certain intervals, even if there are engine-to-engine manufacturing differences and engine deterioration. In an analogous manner to the case for thrust discussed above, if good estimates of stall margins were available, one could reduce the design safety margins, increasing the overall efficiency and life of the engine.

In addition to thrust and stall margins, there are other engine parameters that one may want to estimate. For instance, there are internal variables that cannot be measured (e.g., the temperature at the combustor inlet) and can be viewed as unknown states, or actuators whose commanded input is often different from the actual one (e.g., the fuel flow actuator). Estimates of such variables can be useful in designing new control schemes or improving the performance of existing ones. Clearly, estimating engine thrust, stall margins, and other engine parameters (which we will later refer to as “engine states”) is a very important problem.

The particular engine we study here is the General Electric Aircraft Engine (GEAE) XTE46 which is a scaled unclassified version of GEAE’s variable cycle engine. It is a “component level model” implemented as a complex Fortran-based simulation of the nonlinear partial-differential equations that represent the engine.

The particular type of estimation problem we study in this paper is how to estimate thrust, stall margin, and an unmeasurable state while the engine is in steady-state operation. It is assumed that a suitable steady-state detection method is implemented and once steady-state is achieved, the estimator is given the engine data and it provides an estimate. Hence, the estimation problem is transformed into that of approximating an unknown function which maps the steady state engine measurements into the variables to be estimated. To perform this task we illustrate a systematic input selection scheme and estimation feasibility analysis, which turn out to be useful tools for computational complexity reduction and estimator redesign.

2 Engine Parameter Estimation

2.1 Estimator Construction Methods

As it was pointed out in the introduction, generally the mathematical model of a jet engine is either un-

available or too complex to be exploited by standard estimation methods. Rather, a more likely situation is that a numerical simulator that approximates the physical engine with high accuracy is available. In this case, the designer faces two options:

- (i) Develop, using the simulator, a simplified linear or nonlinear dynamical model that could be used in conjunction with available estimation techniques in order to recover the unknown parameters.
- (ii) From a finite number of simulator data, directly extrapolate an approximation of the *static relationship** between measurable variables and unknown parameters.

Here, we investigate case (ii) deriving the relationship described above by means of nonlinear approximators. Among various approximator structures that can be employed, we choose multilayer feedforward neural networks [2]. We recognize that other approximators may be more suitable for this application (e.g., fuzzy systems, polynomials, or wavelets), providing same or better estimation results with less computational complexity. Nevertheless, since our goal is to illustrate the estimation technique and the need of input selection methods and estimation feasibility analysis, we intentionally do not focus on this issue; the arguments and the ideas that we introduce in this paper do not depend on such a choice.

2.1.1 Assumptions: For the problem to be well-posed, the following assumptions are needed:

- (i) In steady state, there exists a diffeomorphism (i.e., a continuously differentiable, invertible map) between the set of sensor measurements and the unknown parameter we wish to estimate. In particular, $y = \mathcal{F}(\mathcal{S})$, $\mathcal{S} \in U_{\mathcal{S}} \subset \mathbb{R}^n$, where \mathcal{S} is a vector of n sensor measurements, y is the unknown parameter to be estimated, and $U_{\mathcal{S}}$ is a compact set of \mathbb{R}^n .
- (ii) \mathcal{S} is known or, in other words, we know the set of sensor measurements which is needed to estimate y .
- (iii) \mathcal{F} is not known analytically but, using an accurate simulator, it is possible to calculate the value of \mathcal{F} at a finite number of points, $y^i = \mathcal{F}(\mathcal{S}^i)$, $i = 1, \dots, M$.

If assumptions (i)-(iii) are satisfied, \mathcal{F} can be approximated by a tunable function (or *approximator*) $\hat{\mathcal{F}}(\mathcal{S}, \theta)$,

*We use the word *relationship* intentionally, to keep the idea as general as possible. Such a relationship might be represented, among others, by a continuous function, a logic function, or a probabilistic distribution, and it can be approximated by a tunable function (approximator), a lookup table, and a Bayesian belief network, respectively.

were $\theta \in \mathbb{R}^p$ is a vector of parameters to be optimized, provided that the candidate approximator possesses the universal approximation property or, in other words, $\hat{\mathcal{F}}$ can approximate any continuous function \mathcal{F} with arbitrary accuracy over a compact set. It has been proven [3] that feedforward neural networks enjoy this property and, since $U_{\mathcal{S}}$ is assumed to be a compact set, we conclude that there exists θ^* such that $\hat{y} = \hat{\mathcal{F}}(\mathcal{S}, \theta^*)$ can be used to estimate y , and can be made arbitrarily close to y by increasing the number of parameters p . Finding θ^* is, in general, a difficult task, particularly when the parametric dependence of the approximator is nonlinear.

2.2 Application to XTE46 Engine

In this section neural networks are applied to the estimation of thrust, compressor stall margin, and an unmeasurable state for the XTE46 engine. We stress that, at this point, assumption (ii) is not satisfied, in that we do not know which variables are needed to estimate the unknown parameters, therefore \mathcal{S} will be chosen according to intuition. In Section 3 we will introduce a systematic method for choosing \mathcal{S} , showing its effectiveness as compared to the results presented here.

In order to build estimators, it is first necessary to generate a training set. We define the engine *operating condition* at fixed environmental temperature as the triple (**altitude**, **mach number**, **power code**)[†]. This triple, together with the set of unmeasurable states[‡], completely specifies the simulation parameters of the engine when the environmental temperature is constant. The inputs to thrust and stall margin estimators come from sensed parameters of the engine in steady state and actuators values; for the engine state, we use the error between the engine sensed parameters and the equivalent outputs of an embedded model representing a nominal engine. The embedded model is currently used by GEAE for control purposes; refer to [4] for a more detailed description.

We perform “regional” estimation, i.e., the validity of our estimator will be confined to a specific region of the operating space, which we choose to be “takeoff” (i.e., **altitude** in [0, 5000], **mach number** in [0.2, 0.4], **power code** in [45, 50]). This choice allows for a good estimation performance, which can be difficult to obtain on the whole operating space. On the other hand,

[†]Altitude is in the interval [0, 50000] feet, mach number is in [0, 1.7], and power code denotes the throttle angle which ranges between 20 and 50 degrees. Variations in day temperature are not taken in account in this study, but the techniques introduced here can as well be applied to the case when the environmental temperature is allowed to vary.

[‡]We generically refer to “unmeasurable states” of XTE46 without describing them in detail, since the real scope of this paper is introducing input selection and estimation feasibility analysis for jet engines, without restricting ourselves to XTE46. Rather, this engine is used as a testbed for our techniques.

this restriction is not conservative, since the majority of the real flight conditions can be covered by three or more regions of the same size, e.g., “takeoff,” “climb,” and “cruise.”

The training and testing sets are formed by $M = 1000$ and $M_{test} = 2000$ data pairs, respectively. The inputs for the three estimators were chosen according to physical considerations and the intuition derived by experience. Generally, this is the approach that a designer would first follow when dealing with a complex dynamical system. Due to space limitations we do not include the list of these inputs, it suffices to say that 14 sensor values were used for the estimation of thrust and stall margin, and 21 for the engine state. The estimation results are shown in Table 1, where the neural networks used to estimate stall margin and engine state have 5 neurons, while the one used to estimate thrust has 13 neurons. The indices shown in the table refer to the estimation error on 2000 points of the testing set, with the exception of thrust, for which the indices refer to the percentage estimation error: $100 \cdot (\text{thrust} - \text{estimated thrust}) / \text{thrust}$. The symbol σ denotes the error standard deviation, and the weighted mean error $W(e)$, not used for the engine state, is defined as $W(e) = \frac{1}{M} \sum_{i=1}^M |e_i \left(\frac{PC_i}{50}\right)|$, where e_i and PC_i denote the estimation error and power code values for the i -th data point.

Table 1: Estimation results.

	% Thrust	Stall marg.	Flow eff.sc.
$MAX(e)$	0.103	0.95	$4.64 \cdot 10^{-3}$
$MEAN(e)$	0.020	0.27	$6.43 \cdot 10^{-4}$
$MED.(e)$	0.016	0.24	$5.08 \cdot 10^{-4}$
$2\sigma(e)$	0.032	0.39	$1.09 \cdot 10^{-3}$
$W(e)$	0.019	0.26	-

These results are adequate for practical applications, with the drawback that too many sensor measurements are needed for the estimation. A systematic input selection technique is needed in order to pick the minimal set of inputs for our estimators, and is explained in next section.

3 Estimator Input Selection

3.1 Correlation Analysis Approach

In a correlation analysis approach, we view the engine as a stochastic system that generates some output variables as statistical functions of other variables. These variables may or may not be measurable, and the goal of the estimator design is to characterize the *measurable* variables that are most highly correlated to the parameters that we would like to estimate. The correlation can be viewed as an indicator of the quantity of information that two random variables carry on together, and is defined as follows:

$$\rho_{xy} = \frac{E[(X - \bar{X})(Y - \bar{Y})]}{\sigma_X \sigma_Y} \quad (1)$$

where \bar{X}, \bar{Y} denote the expected values of X, Y , respectively; $E(\cdot)$ denotes is expectation operator, and σ_X, σ_Y are the standard deviations of X, Y , respectively. Note that $0 \leq |\rho_{xy}| \leq 1$. Given M points, we will approximate ρ_{XY} by means of the following unbiased estimator [5]

$$\hat{\rho}_{xy} = \frac{\frac{1}{M} \sum_{i=1}^M (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\frac{1}{M} \sum_{i=1}^M (X_i - \bar{X})^2 \frac{1}{M} \sum_{i=1}^M (Y_i - \bar{Y})^2}} \quad (2)$$

Below, we outline a systematic method for the selection of reasonable sets of inputs to be used in the estimation of an engine parameter.

Data Set Generation: Generate a large number of data points (generally, a large data set will provide more information) in a way analogous to what has been done in Section 2.2. The resulting data will be stored in an $n \times M$ matrix, where n is the total number of engine variables.

Form Correlation Coefficient Matrix: Generate, by using (2), an $n \times n$ correlation coefficient matrix and take its element-wise absolute value and call it C . The correlation matrix will be symmetric, therefore row i is equal to column i for all $i = 1, \dots, n$. The (i, j) -th element of C will be the correlation coefficient between the i -th and j -th variables of the engine.

Eliminate Input Variables with Low Correlation: Take the row of C corresponding to the variable to be estimated, say the l -th one. Fix a threshold $\underline{\delta}$ between 0 and 1 and eliminate all the variables that have correlation coefficient less than $\underline{\delta}$ and the ones that are not measurable (in particular, use the sensed outputs of the measurable variables). The choice of $\underline{\delta}$ can be made in many ways; a very simple one is to plot the l -th row of C and decide heuristically a reasonable value for $\underline{\delta}$ that keeps a sufficiently large set of variables, while discarding the ones with very low correlation coefficients.

Study Cross-Correlation: For every element of this subset with correlation coefficient greater than $\underline{\delta}$, look at its correlation coefficient with all the other elements of the subset, form a matrix (which is a sub-matrix of C) containing all these cross-dependencies, and call it C' . A direct examination of this matrix will show the variables that are highly cross-correlated, and therefore carry redundant information.

Eliminate Redundant Input Variables: Looking at C' , discard the variables with cross-correlation greater than $\bar{\delta}$, (a typical value for $\bar{\delta}$ is 0.95) keeping the one with highest correlation with the l -th variable of the engine.

Construct Estimator: Train the estimator with the set of inputs given by the above correlation analysis procedure, and testing it on the test set. Store the error vector.

Find Correlation Between Inputs and Estimation Errors: Calculate and plot the correlation coefficients of this error vector with respect to the engine

variables.

Add Input Variables that Affect Estimation Error: If some measurable variable is significantly correlated to the estimation error, this means which we are not exploiting the statistical information contained in this variable. The ideal situation is when the error is uncorrelated to all the variables of the engine, meaning that all the possible information has been used. Form a subset made up of those variables which are measurable, significantly correlated with the estimation error, and that are not already included in our set of inputs.

Eliminate Redundant Variables: Discard from this subset all the variables that are redundant, similarly to how this is done above, and include the remaining variables in the set of inputs \mathcal{S} .

3.2 Case Study: Estimation of Thrust and Stall Margin

Let us now apply the correlation analysis to input selection for thrust and stall margin estimators. The estimation is performed during “takeoff” using multi-layer feedforward neural networks, and the results will be compared to the ones obtained in Section 2.2. Our data set is formed from 4000 data points generated in the manner explained before; this is enough points to render the samples statistically representative. The total number of variables contained in this data set is 73, therefore $n = 73, M = 4000$. We number the variables from 1 to 73. Thrust is variable number 2, therefore $l = 2$. By plotting the correlation coefficients versus the variable index number we found that a good choice of the first threshold $\underline{\delta}$ appears to be $\underline{\delta} = 0.7$. The application of the successive steps of correlation analysis leads to the choice of the following set of inputs for the estimator: `combustor fuel flow, fan rotor speed, core rotor speed, altitude, power code`. Calculating the testing error after training the neural network, and estimating the correlation coefficient between this error and all the other variables, it is observed that the error is significantly correlated with the measurable variables (we do not consider variables that we cannot measure) with indexes 39, 56, 72, corresponding to `exhaust nozzle area, fan inlet total temperature, mach number`. Noticing that there is no redundant variable in this subset, and therefore including these variables in our set of inputs, we get the set $\mathcal{S} = \{\text{combustor fuel flow, fan rotor speed, core rotor speed, altitude, power code, mach number, exhaust nozzle area, fan inlet total temperature}\}$ and the estimation performance results in Table 2, which are better than the ones shown in Table 1, with a significant reduction in the number of inputs (8 versus 14) and of the parameters (51 versus 209).

By applying correlation analysis to the estimation of compressor stall margin we get \mathcal{S} to be the

Table 2: Estimation results for thrust: 8 variables chosen as inputs using correlation analysis.

$\% \text{ MAX}$	$\% \text{ MEAN}$	$\% \text{ MED.}$	$\% 2\sigma$	$\% W(e)$
0.088	0.017	0.014	0.027	0.016

following set of seven inputs: `{compressor inlet total temperature, fan inlet total pressure, combustor inlet static pressure, combustor fuel flow, exhaust nozzle area, rear VABI, power code}`, with estimation results shown in Table 3. The results are almost indistinguishable from the

Table 3: Estimation results for stall margin: 7 variables chosen as inputs after correlation analysis.

MAX	MEAN	MEDIAN	2σ	$W(e)$
1.05	0.29	0.25	0.41	0.27

ones of Table 1, but the computational complexity has been reduced significantly (7 inputs and 46 parameters versus 14 inputs and 81 parameters).

The two cases illustrated above are quite representative, and they illustrate the principles of the correlation analysis approach. The application of this technique to engine state estimator design would provide similar results that, due to space constraints, we do not include.

4 Estimation Feasibility Analysis (EFA)

In EFA we include a set of techniques aimed at providing some measure of “observability” of the unknown engine parameters with respect to a given sensor set. If we were able to predict the estimation performance accurately enough, we could explore the advantages and disadvantages of a given sensor set without needing to build estimators. Note the difference between EFA and input selection: given an unknown variable and a set of inputs, input selection discards the unnecessary inputs for the estimation, whereas EFA indicates the feasibility of estimating the variable using that set of inputs, as is. In the next section we introduce two different approaches for estimation feasibility analysis; in our description, without loss of generality, we will refer to the “operating space” as the 3 dimensional space for `altitude, mach number, and power code` defined for the XTE46 engine described earlier.

4.1 Data Based Approach

Inspired by the correlation analysis introduced in Section 3, we introduce here a method that does not use any direct information about the nominal model, relying totally on the available input-output engine data[§].

[§]The input-output engine data, however, are generated by a simulator, as pointed out by assumption (iii).

Operating Space Partitioning: Partition the operating space into local regions. In practice, one can choose to partition the operating space into cubes whose size is chosen according to physical intuition. We will denote each of the cubes by $C_{alt,mach,pc}$, where $alt,mach,pc$ are integers determining the position of the cube in the operating space.

Data Matrix Generation: In a way completely analogous to the standard correlation analysis, given a local cube $C_{alt,mach,pc}$ and a sensor set \mathcal{S} , form a data matrix $D^{alt,mach,pc}$ of size $N^{alt,mach,pc} \times n$, where n is the number of measurable outputs associated with the sensor set, and $N^{alt,mach,pc}$ is the number of data points available for a specific region $C_{alt,mach,pc}$. Next, create the matrix $E^{alt,mach,pc} = [D^{alt,mach,pc}, Y^{alt,mach,pc}]$, where each row of the $N^{alt,mach,pc} \times n_y$ matrix $Y^{alt,mach,pc}$ contains n_y unmeasurable parameters associated with one data point. The i -th row of $E^{alt,mach,pc}$ contains the i -th data point consisting of n measurable outputs, and n_y unmeasurable parameters.

Correlation Vectors Generation: Using (2) generate a $(n + n_y) \times (n + n_y)$ correlation coefficient matrix. Each row (or column) of this matrix contains the correlation coefficients between the corresponding variable and the remaining ones. Define $c^{alt,mach,pc}$ as the sub-matrix formed by the first n rows and the last n_y columns of such matrix.

Correlation Measure: Choose a function $\mu : \mathbb{R}^n \rightarrow \mathbb{R}$ to map each of the columns of $c^{alt,mach,pc}$ into a number $\rho^{alt,mach,pc}$ representative of the total correlation between the specific unmeasurable parameter and the measurable outputs. Let v be a generic column of $c^{alt,mach,pc}$, then good candidates for μ are $\text{mean}(v)$, $\|v\|$, and $\max(v)$.

The procedure described above, given a region $C_{alt,mach,pc}$ and a sensor set \mathcal{S} , will generate $\rho^{alt,mach,pc}$ for all the non-measurable parameters.

4.2 Simulation Model Based Approach

This technique exploits the availability of a simulation model of the engine, and provides an ‘‘observability’’ measure without using any measurements from the engine. The simulation model can be used to approximate the partial derivative of each element[¶] of \mathcal{S} with respect to each non-measurable parameter. The resulting Jacobian matrix is manipulated to generate the ‘‘observability index.’’

[¶]The partial derivative, however, can be calculated only if the non-measurable variable parameterizes the engine model, i.e., if a variation of this variable generates a variation in the engine outputs. Thrust and stall margins, being unmeasurable outputs of the engine, do not parameterize it and, therefore, the model based approach cannot be used to perform estimation feasibility analysis on these parameters. On the other hand the technique is particularly suitable for variables such as the engine states or unreliable actuators.

Operating Space Partitioning: same as for the data based approach above.

Partials Generation: Given a sensor set \mathcal{S} , for each cube $C_{alt,mach,pc}$ calculate N Jacobians of the non-measurable variables with respect to \mathcal{S} corresponding to N different engines each in a different operating condition within $C_{alt,mach,pc}$. Each Jacobian will be a matrix $J^i, i = 1, \dots, N$ of dimension $n \times \bar{n}_y$, where \bar{n}_y is the number of non-measurable variables for which the partials can be generated. Set $\bar{J}^{alt,mach,pc} = \frac{1}{N} \sum_{i=1}^N (J^i)$, to be the averaged Jacobian corresponding to region $C_{alt,mach,pc}$.

Observability Index Calculation: Given the matrix $\bar{J}^{alt,mach,pc}$ calculate the following observability index, associated with the sensor set \mathcal{S} and the k -th non-measurable variable:

$$o_{alt,mach,pc}^k = \frac{\|\bar{\mathcal{J}}^k\|_1}{\sum_{l=1}^{\bar{n}_y} \|\bar{\mathcal{J}}^l\|_1} \text{cond}(\bar{J}^{alt,mach,pc})^{-1} \quad (3)$$

$$\|\bar{\mathcal{J}}^k\|_1 \triangleq \sum_{i=1}^n |\bar{\mathcal{J}}^k|_i, \quad k = 1, \dots, \bar{n}_y \quad (4)$$

$$\text{cond}(\bar{J}^{alt,mach,pc}) = \sqrt{\frac{\lambda_{max}(\bar{J}^T \bar{J})}{\lambda_{min}(\bar{J}^T \bar{J})}} \quad (5)$$

where $\mathcal{J}^k, k = 1, \dots, \bar{n}_y$, is the k -th column of $\bar{J}^{alt,mach,pc}$ and $|\mathcal{J}^k|_i, i = 1, \dots, n$, is the absolute value of the i -th element in the above column. Notice that $\text{cond}(\bar{J}^{alt,mach,pc})$, as defined here, is the condition number of the matrix $\bar{J}^{alt,mach,pc}$. The procedure described above, given a region $C_{alt,mach,pc}$ and a sensor set \mathcal{S} , will generate $o_{alt,mach,pc}^k, k = 1, \dots, \bar{n}_y$ for each non-measurable parameter. Due to space constraints we do not include the mathematical considerations leading to the choice of o^k .

4.3 Global Linear Estimation

In order to evaluate the results of estimation feasibility analysis and show the effectiveness of the techniques that have been introduced, we develop, for each region $C_{alt,mach,pc}$, a linear least squares estimator. The data points employed in the data based approach (see Section 4.1) can be used to calculate the linear model $\mathcal{Y} \approx R\mathcal{S}$, where $R = Y^{alt,mach,pcT} D^{alt,mach,pc} (D^{alt,mach,pcT} D^{alt,mach,pc})^{-1}$, for all engines in $C_{alt,mach,pc}$. Switching between the linear estimators, we can build a global linear estimator over the operating space. It is expected that the resulting estimator performs well in the regions of the operating space with a high ‘‘observability’’ and low nonlinearity. On the other hand, if the ‘‘observability’’ is low and the nonlinearity significant, the estimation performance should be poor. The global linear estimator is used to demonstrate the effectiveness of estimation feasibility analysis, by showing that in regions with high estimation feasibility index where the linear estimator performs poorly, the

nonlinear estimator introduced in Section 2.1 produces a very accurate estimate, because it takes advantage of the good “observability” properties of the engine that otherwise, due to nonlinearity of the mapping \mathcal{F} , cannot be exploited.

4.4 Case Study: Stall Margin and Engine State Estimator Redesign

EFA might be useful in identifying operating space regions in which linear estimation cannot exploit the available observability. In these regions a nonlinear estimator should improve estimation performance significantly. Here, we validate this idea by applying estimation feasibility analysis to compressor stall margin (using the data based approach) and an engine state (using the simulation model based approach) for all the speed, pressure, and temperature sensors that may be available in a typical engine. Any other set of sensors would therefore be a subset of this one.

We start by dividing the operating space **altitude**, **mach number**, **power code** into $12 \times 12 \times 12$ cubes, collecting input-output engine data (150-300 points for each region) that can be used to perform data based EFA and to build a global linear estimator as discussed in Section 4.3. As for the model based approach, for each cubic region, we approximate $N = 30$ Jacobians and we calculate the “observability” index in (4). The results can be plotted using 3 dimensional “slice plots” in which each axis represents the corresponding operating condition, and the plot color represents the degree of “observability” calculated by means of EFA. The same representation is used to show the estimation error of the global linear estimator as a function of the operating condition. Hence, estimation error and EFA results can be compared and similarities or discrepancies can be easily found, as shown in Figure 1, where estimation performance and EFA results are compared for compressor stall margin and an engine state. Notice that dark gray is assigned to high estimation error and low “observability,” while white indicates good estimation performance and high “observability.” As for the compressor stall margin, the figure indicates a sharp difference between EFA and linear estimator in region $C_{8,10,6}$, where the correlation measure is high but the estimation performance is poor. This discrepancy leads us to believe that the global linear estimator can be redesigned by training, in those regions, a nonlinear estimator that can hopefully exploit the observability properties shown by estimation feasibility analysis. Using $W(e)$ as error performance index in testing (see Section 2.2 for a definition of $W(e)$), a neural network trained over this region achieves $W(e) = 5.45 \cdot 10^{-2}$, whereas the corresponding performance index for the linear estimator is $W(e) = 2.38 \cdot 10^{-1}$. Hence, a nonlinear estimator improves the estimation performance by a factor of four.

The model based analysis for the engine state shows a

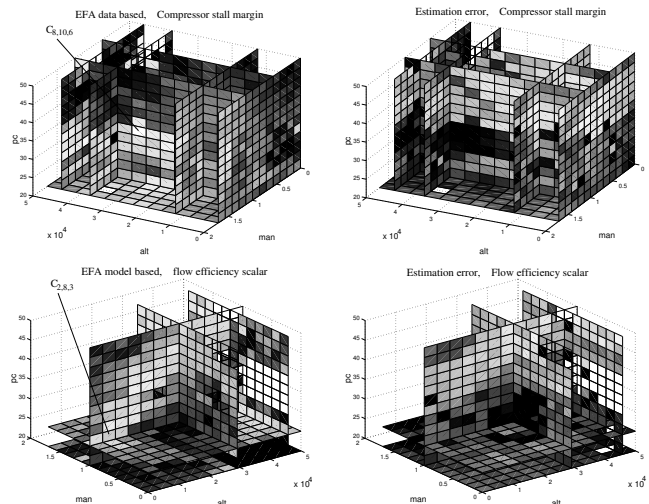


Figure 1: Comparison between EFA and estimation performance.

discrepancy in region $C_{2,8,3}$. Here, however, the difference between actual estimation error and “observability” index is less remarkable. Using the mean as an error performance index, a neural network trained over $C_{2,8,3}$ achieves $\text{mean}(e) = 1.68 \cdot 10^{-4}$ which, compared to $\text{mean}(e) = 1.27 \cdot 10^{-3}$ obtained by the global linear estimator, shows an improvement in the estimation by a factor of seven. These two examples show the advantage of nonlinear over linear estimation, confirming the information provided by EFA.

References

- [1] H. Emmons, C. Pearson, and H. Grant, “Compressor surge and stall propagation,” *Transactions of the ASME*, vol. 77, pp. 455–469, 1955.
- [2] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*. PWS Publishing Company, 1996.
- [3] G. Cybenko, “Approximation by superposition of a sigmoidal function,” *Mathematics of Control, Signals, and Systems*, no. 2, pp. 303–314, 1989.
- [4] S. Adibhatla and Z. Gastineau, “Tracking filter selection and control mode selection for model based control,” in *30th AIAA/AMSE/SAE/ASEE Joint Propulsion Conference*, (Indianapolis, IN), June 27-29 1994.
- [5] H. Stark and J. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*. Prentice Hall, 1994.