# Estimator Design in Jet Engine Applications*

Manfredi Maggiore*
Raúl Ordóñez*
Kevin Passino*
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]. Rotating stall is described in Figure 1, where a row of axial compressor blades is shown: a non-uniformity in the inlet flow causes an increase in the angle of attack of blade 1, creating a stall, i.e., a flow-blockage between blade 1 and 2. This blockage causes the inlet flow to be diverted and directed away from blade 1 and towards blades 2 and 3. Now, the angle of attack of the inlet flow on blade 2 is increased, generating a new stall. This process continues and propagates the stall to the blades of the entire blade row, causing a significant loss of thrust, undesired vibrations in the blading [2, 3], and reduced pressure rise of the compressor. To avoid such phenomena, several unmeasurable "stall margin" variables are generally introduced to

*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.

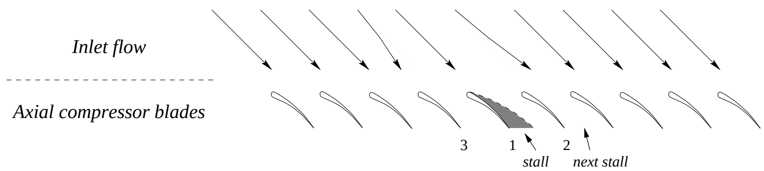


Figure 1: Propagation of stall cells in rotor blades.

characterize how close the system is to a stall condition. 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 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. Effects of the engine-to-engine manufacturing differences and engine deterioration can be accounted for in the simulation. Due to the engine simulator complexity there is no simple explicit mathematical model (e.g., an analytical nonlinear model) that is available for use in applying standard parameter estimation methods. The simulator can, however, be used to evaluate estimation schemes, and it can generate engine input-output data that can be used in estimator construction.

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. Here, we will use a standard neural network approach, with data generated by the engine simulator, to solve this problem. Our focus is not on how to pick the best engine training data, training method, or neural network structure. That is, our focus is not on the parameter estimation method, but on issues encountered when designing such estimators. Our neural network estimator will simply serve to help us illustrate our technique for estimator input selection and to show how our method of estimation feasibility analysis can be useful in estimator design.

Why is estimator input selection a difficult problem? Why not simply use all the available sensed variables as the inputs to an estimator? First, if we could achieve a good estimation performance using a subset of the full set of sensors, the number of on-board sensors could be decreased thereby reducing the production costs. Second, using too many inputs to the estimators would unnecessarily increase their complexity and hence could create problems with implementation. Third, there are typically dependencies between various sensed values so that adding more estimator inputs does not necessarily add more information to help the estimator solve its problem. In fact, as we will explain in Section 2, the additional inputs can degrade estimation performance. It is clear that it is useful to prune the size of the set of inputs, and the normal approach to do this is to use intuition to pick a smaller set of what one considers to be the most useful inputs. This is in fact how earlier work on this problem proceeded. In this paper we will show, however, that a better choice is possible using a systematic correlation analysis approach. It is important to note that estimator input selection methods have already been used in the system identification literature (e.g., [4, 5]) to validate the regressor vector of identification models. These ideas, however, do not apply to our problem, since we work in a static framework where all the engine variables are assumed to be in steady state. Furthermore, we do not look for a regressor vector that is dependent on the choice of the identification model. Rather, we need to find the set of inputs of an unknown function independent of the choice of the approximator structure. Moreover, methods found in the system identification literature often deal with "input dimension reduction" (see, for example, [6] for a description of OLS, and [7] for a description of PLS, which can be used in a similar fashion), which is achieved by applying a transformation on the inputs in a suitable way, so that unnecessary variables in the transformed space can be easily discarded. These methods, therefore, by performing input selection in the transformed space, do not eliminate the need of all the original input variables, and hence cannot be applied to the solution of our problem.

Why is it important to study estimation feasibility before designing an estimator? First, it is important to know if, for a given set of estimator inputs, a variable can be estimated, or if additional inputs are needed. Especially important is the case where an estimation feasibility analysis indicates that for the given sensor set it is not feasible to estimate some engine parameter, since in this case there may be the need to invest in an additional sensor (here, of course, input selection would indicate that an unmeasurable variable is essential to the estimation of the engine parameter, confirming the need of an additional sensor to measure it). In addition, estimator feasibility analysis may show regions of the engine operating space where a variable is particularly easy (or difficult) to estimate. If the analysis shows a region where a parameter is easy to estimate then it may be possible to use a local linear estimation scheme there. On the other hand, if in a certain region the estimation feasibility analysis indicates that estimation will be difficult, then nonlinear estimators may be needed in that region to get adequate estimation performance. In this paper we study two estimation feasibility approaches and give an example to show how estimation feasibility analysis can identify a region where a parameter is difficult to estimate by a linear estimator but its estimation feasibility turns out to be high, and, indeed, a nonlinear estimator can significantly improve performance. From this discussion, it should be clear that input selection and estimation feasibility analysis are closely related, even though we split them into two problems.

Furthermore, it should be clear that estimation feasibility is related to the standard concept of observability which is well studied for certain classes of finite dimensional systems (linear [8], feedback linearizable [9, 10]); unfortunately no existing analytical observability test can be applied to complex dynamical systems like jet engines (unless, of course, a simplified mathematical model is used to represent it, introducing the unavoidable risk of misrepresenting certain dynamics and nonlinearities in the real system and hence leading to erroneous conclusions about observability). For practical purposes, one needs to have some kind of "observability index" that can be calculated directly from the engine data or a very accurate simulation model such as the one we use for the XTE46.

The paper is organized as follows: Section 2 briefly discusses our estimator construction method and its application to the XTE46 estimation problems when only intuitive ideas are used for estimator input selection. A correlation analysis approach to input selection is introduced in Section 3, applied to estimator design, and compared to the results of Section 2. In Section 4 two procedures to perform estimation feasibility analysis are introduced, together with a global linear estimator which is used as a base-line comparison to show the effectiveness of these techniques in nonlinear estimator design. Finally, some conclusions are drawn in Section 5.

# 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 unavailable 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*[1] between measurable variables and unknown parameters.

Both these options have some drawbacks: a linear or nonlinear model may be suitable for application of a standard estimator design approach, but not accurate enough to express the physical properties of the engine over its operating space, rendering the estimate inherently unreliable. On the other hand, method (ii) may fail in extrapolating the relationship between sensor measures and unknown parameters, providing unreliable results as well. Notice that case (i) exploits dynamical estimation, whereas (ii) performs a steady-state static estimation.

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 [11], which will be described in detail in Section 2.1.2. 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,

$$y = \mathcal{F}(\mathcal{S}), \quad \mathcal{S} \in U_{\mathcal{S}} \subset \Re^n \tag{1}$$

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 $\Re^n$.

(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,

$$y^i = \mathcal{F}(\mathcal{S}^i), \quad i = 1, \ldots, M. \tag{2}$$

[1]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.

If assumptions (i)-(iii) are satisfied, $\mathcal{F}$ can be approximated by a tunable function (or *approximator*) $\hat{\mathcal{F}}(\mathcal{S}, \theta)$, were $\theta \in \Re^p$ 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 [12] that feedforward neural networks enjoy this property and, since $U_{\mathcal{S}}$ is assumed to be a compact set, we conclude that there exists $\theta^*$ 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 $\theta^*$ is, in general, a difficult task, particularly when the parametric dependence of the approximator is nonlinear. Next, we will describe a training technique to tune this vector of parameters in feedforward neural networks.

### 2.1.2 Multilayer Feedforward Neural Networks

The basic structure of a neural network, in the case of one hidden layer, is showed in Figure 2, and the input to the $j^{th}$ node in the $i^{th}$ layer is the weighted sum

$$s_{i,j} = w_0^{i,j} + \sum_{k,l} w_{k,l}^{i,j} o_{k,l}$$

where $w_0^{i,j}$ is the "bias" for node $i, j$, and $o_{k,l}$ is the output of the $l^{th}$ node in the $k^{th}$ layer ($o_{k,l} = g(s_{k,l})$ where $g(\cdot)$ is a sigmoid activation function, e.g., $g(x) = (1 + exp(-x))^{-1}$). Therefore, given a neural network with $n$ inputs, one hidden layer with $h$ neurons, and one output, the total number of parameters to tune is $(n+2)h+1$. Now, assume we stack these in a parameter vector that we denote with $\theta$.

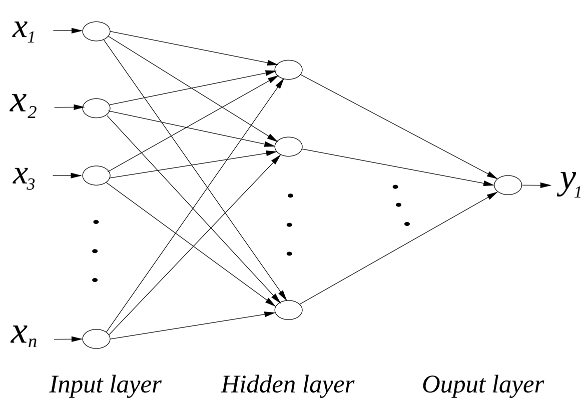


Figure 2: Diagram of a multilayer neural network.

Given a finite set of data pairs (also called a *training set*) $(\mathcal{S}^i, y^i)$, $i = 1, \ldots, M$, training an approximator involves tuning the parameters vector $\theta$ in order to achieve a desired approximation error over this set. The problem of the reliability of this estimate outside of the set will be discussed in Section 2.1.3. The standard training method for neural networks is "back-propagation" [11] and its various modifications. Basically, back-propagation is a gradient method applied to the minimization of a nonlinear least squares problem. While back-propagation has the advantage of being a robust, parallel, and distributed algorithm, it suffers from several drawbacks, which include the slowness of convergence in high-dimensional problems and the sensitivity to local minima. Many methods have been developed to overcome the slowness of backpropagation. Among them, the Levenberg-Marquardt optimization technique (which is a modification of the Gauss-Newton method providing, in many cases, better convergence properties) has gained increasing popularity as a neural networks training algorithm, and will be used to train our approximators by employing the Matlab neural networks toolbox.

### 2.1.3 Relationship Between Approximation and Estimation

In this section, we provide a brief explanation of the role of approximation in the estimation of engine parameters. As mentioned before, the training set used to train our approximators is generated using $M$ values of the engine variables in steady state. By assumption (i), there exists a *static* continuous mapping $\mathcal{F}$ which maps $\mathcal{S}$ to $y$, where $y$ can be thrust, stall margin, an unmeasurable state, or an actuator. For example, it is assumed that in steady state, thrust is a *static* function of pressures, temperatures, actuators values, and other variables, measured at different locations of the engine. In general, each variable is a function of a subset of the variables that characterize the behavior of the engine. If assumption (ii) is satisfied, the variables needed for the estimation of $y$ are known. In general, however, assumption (ii) is not satisfied, and the subset $\mathcal{S}$ is largely unknown. Physical intuition might suggest a reasonable set of variables, leading to the choice of a set $\hat{\mathcal{S}}$, but a discrepancy between $\mathcal{S}$ and $\hat{\mathcal{S}}$ is likely, leading to a *functional error*, which is due to the approximation of a function of a set $\mathcal{S}$ of variables by a function of a different set of variables $\hat{\mathcal{S}}$. Input selection is a procedure aimed at eliminating (or minimizing) the *functional error*, by choosing as inputs to the estimator the set of variables which is most correlated to $y$.

When using approximators to recover the unknown variables, we try to approximate the unknown static function which maps the subset of variables into the unknown parameter by another function, our approximator, which is tuned in order to approximate the input-output mapping represented in the training set. As we already pointed out in Section 2.1.2, this raises the issue of reliability of the estimate for points not in the training set, introducing the *generalization error*, the approximation error outside the training set. Generally, to reduce the generalization error the designer has to make a careful choice of the number of parameters of the approximator, which must be small compared to the number $M$ of data points since, when there are too many degrees of freedom as compared to the number of constraints, the problem becomes ill-posed.

Finally, even when using the ideal set of inputs and training our approximator on an infinite size training set, there exists an *approximation error*, which is due to the finite number of parameters of the approximator used to approximate the unknown function $\mathcal{F}$. Approximation error might be reduced by increasing the number of parameters of the approximator with the risk, however, of increasing the generalization error (this will be illustrated with an example below).

We can summarize the previous considerations with the expression for the total estimation error

$$e(\mathcal{S}) = e_{functional}(\mathcal{S}) + e_{generalization}(\mathcal{S}) + e_{approx}(\mathcal{S})$$

This formalization explains two phenomena encountered during our simulations:

- An increase of the number of parameters above a certain threshold does not improve the estimation performance. This behavior has to do with $e_{functional}$ which is independent of the number of parameters and of the approximator structure itself. Therefore, when the number of parameters is high enough, $e_{approx}$ is generally negligible (if the network is trained properly) with respect to $e_{functional}$, and an increase of the number of parameters does not normally significantly affect the performance. Moreover, when using too many parameters, the estimation performance may be even degraded, since $e_{generalization}$ may increase giving rise to *overfitting*, as shown in Figure 3, where 20 samples of the function $y = sin(x)$ are approximated by means of a neural network with 40 or 25 neurons in the hidden layer. The approximator with 25 neurons (76 tunable parameters) performs, outside of the training points, significantly better than the one with 40 neurons (121 tunable parameters).

- Inclusion of any measurable variables in the set of estimator inputs does not necessarily improve the estimation performance. If the included variable does not belong to the ideal set of inputs, it

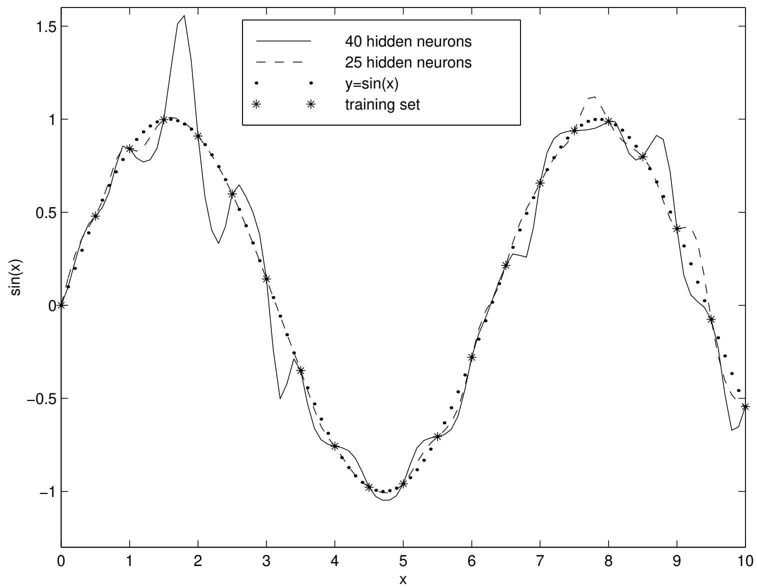


Figure 3: An example of overfitting when too many parameters are used in the approximator.

may be the case that $e_{approx}(\mathcal{S})$ increases, as observed in the course of our simulations. Therefore, it is not necessarily true that the use of every possible measurable variable improves estimation performance.

These considerations lead to the conclusion that there exists a trade-off between estimation accuracy and computational complexity, since increasing the number of parameters of the approximators or the number of inputs to them does not necessarily lead to an improvement of the estimation accuracy. Rather, as observed during our simulations, better results may be obtained with less parameters and a smaller set of inputs.

## 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)`[2]. This triple, together with the set of unmeasurable states[3], specifies exhaustively the simulation parameters of the engine when the environmental temperature is constant.

[2] 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.

[3] 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.

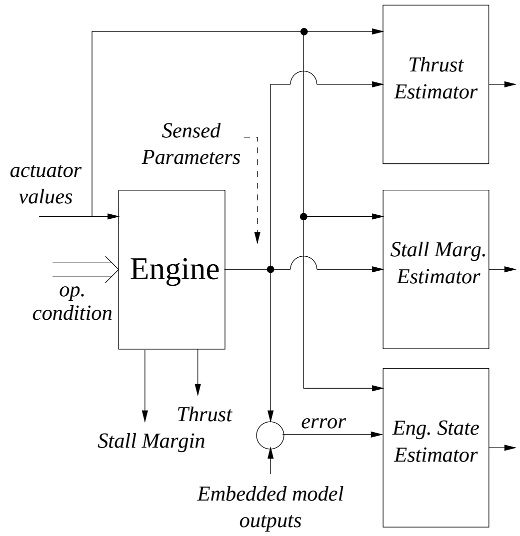


Figure 4: Block diagram of the simulator/estimator setup.

Figure 4 shows a block diagram of our simulator setup: 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 [13] 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, 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."

We choose the training set size $M$ to be big enough to capture the characteristics of the engine in the region of the operating space we are considering, and to minimize the effects of $e_{generalization}(\mathcal{S})$ and $e_{approx}(\mathcal{S})$, described in Section 2.1.3. These effects can be evaluated by testing the approximator on *test data*, which is different from that used for training. Following these guidelines, we chose $M = 1000$, and $M_{test} = 2000$ as the testing set size.

The inputs for the three estimators, shown in Table 1, 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.

The estimation results are shown in Table 2, 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: $100 \cdot$ (thrust - estimated thrust)/ thrust. The symbol $\sigma$ denotes the error standard deviation, and the weighted mean error $W(e)$, not used for the engine state, is defined as

$$W(e) = \frac{1}{M} \sum_{i=1}^{M} \left| e_i \left( \frac{PC_i}{50} \right) \right|$$

Table 1: Variables selected as inputs to the estimators.

| Thrust and Stall Margin |
|---|
| bypass duct static pressure |
| compressor inlet total pressure |
| combustor fuel flow |
| exhaust nozzle area |
| rear variable area bypass injector |
| fan inlet guide vanes |
| compressor inlet hub temperature |
| compressor inlet tip temperature |
| core engine pressure ratio |
| fan rotor speed |
| core rotor speed |
| 3 operating conditions |

| Engine State |
|---|
| bypass duct static pressure error |
| combustor inlet total pressure error |
| combustor fuel flow |
| exhaust nozzle area |
| rear variable area bypass injector |
| fan inlet guide vanes |
| combustor inlet total temperature error |
| compressor inlet tip temperature error |
| core engine pressure ratio error |
| fan rotor speed error |
| core rotor speed error |
| 3 operating conditions |
| engine pressure ratio error |
| HP turbine inlet temperature error |
| specific fuel consumption estimated error |
| thrust estimated error |
| temperature at combustor inlet error |
| combustor inlet static pressure |
| LP turbine blade temperature |

where $e_i$ and $PC_i$ denote the estimation error and power code values for the $i$-th data point. These

Table 2: Estimation results.

| Error Index | % Thrust | Stall margin | Flow eff. scalar |
|---|---|---|---|
| *MAX(e)* | 0.103 | 0.95 | $4.64 \cdot 10^{-3}$ |
| *MEAN(e)* | 0.020 | 0.27 | $6.43 \cdot 10^{-4}$ |
| *MEDIAN(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 | - |

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:

$$\rho_{xy} = \frac{E[(X - \bar{X})(Y - \bar{Y})]}{\sigma_X \sigma_Y} \tag{3}$$

where $\bar{X}, \bar{Y}$ denote the expected values of $X, Y$, respectively; $E(\cdot)$ denotes is expectation operator, and $\sigma_X, \sigma_Y$ are the standard deviations of $X$, $Y$, respectively. Note that $0 \leq |\rho_{xy}| \leq 1$. Given $M$ points, we will approximate $\rho_{XY}$ by means of the following unbiased estimator [14]

$$\hat{\rho}_{xy} = \frac{\frac{1}{M}\sum_{i=1}^{M}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\frac{1}{M}\sum_{i=1}^{M}(X_i - \bar{X})^2 \frac{1}{M}\sum_{i=1}^{M}(Y_i - \bar{Y})^2}} \tag{4}$$

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

1. **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: for each data point, engine states and operating conditions are chosen randomly within their range of validity (this implies that for each data point a random engine is specified), and all the engine variables are stored. The resulting data will be stored in an $n \times M$ matrix, where $n$ is the total number of engine variables. The $i$-th row of this matrix will contain $M$ steady state values of the $i$-th variable of the engine, each of them corresponding to a random choice of engine states and operating conditions.

2. **Form Correlation Coefficient Matrix:** Generate, by using (4), 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 $i = 1, \ldots, n$. The $(i, j)$-th element of $C$ will be the correlation coefficient between the $i$-th and $j$-th variables of the engine. The $k$-th row of $C$ will contain the correlation coefficients between variable $k$ and all the other variables of the engine. Some entries of this matrix corresponding to actuator values will be undefined because of the on-off nature of these variables.

3. **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[4] (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.

4. **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. For example, if it is found that thrust is highly correlated to combustor inlet static pressure and compressor inlet total pressure, and the analysis above shows that these two pressures are correlated with correlation coefficient 0.96, then they carry nearly the same statistical information. This observation leads to the following step.

5. **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. For example, in the case above, if the correlation coefficient between combustor inlet static pressure and thrust is higher than the one between compressor inlet total pressure and thrust, discard the latter measurement.

[4]One could decide to exclude unmeasurable variables from the data set. As pointed out in the introduction, though, it could be interesting to perform input selection including some unmeasurable variables which could be sensed by additional sensors. Hence, in general, we assume that the data set contains some unmeasurable variables.

6. **Construct Estimator:** Train the estimator with the set of inputs given by the above correlation analysis procedure, using the procedure outlined in Sections 2.1.2 and 2.2, and testing it on the test set. Store the error vector.
7. **Find Correlation Between Inputs and Estimation Errors:** Calculate and plot the correlation coefficients of this error vector with respect to the engine variables.
8. **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.
9. **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 $\mathcal{S}$.

In our experience with the XTE46 engine, the resulting set of variables tends to be a good candidate for being inputs to the estimator for the $l$-th variable. The procedure above can be iterated until the error is uncorrelated to all the measurable variables. The choice of the two thresholds $\underline{\delta}$ and $\bar{\delta}$ determines the size of the final set of inputs. In our experience, however, it has always been very easy to come up with a good choice, as will be evident in the examples that follow which we use to show the effectiveness of the above procedures for estimator input selection.

### 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 multilayer feedforward neural networks, and the results will be compared to the ones obtained in Section 2.2. Our data set is formed of 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 $l = 2$. A plot of the correlation coefficients versus the variable index number is found in Figure 5. Clearly the variable thrust ($l = 2$) has correlation coefficient one with itself.

Looking at this graph, a good choice of the first threshold $\underline{\delta}$ appears to be $\underline{\delta} = 0.7$, which, after the exclusion of the non-measurable variables, gives the following subset of variables:

| | | |
|---|---|---|
| `bypass duct stat.press.` | `compressor inlet tot.press.` | `compressor inlet tot.temp.` |
| `compressor inlet tip temp.` | `combustor inlet stat.press.` | `LP turbine blade temp.` |
| `LP turbine frame stat.press.` | `LP turbine exit temp.` | `temperature at combustor inlet` |
| `fan rotor speed` | `core rotor speed` | `combustor fuel flow` |
| `altitude` | `power code` | |

The next step involves studying the cross-correlation among the variables of this subset. We choose a threshold $\bar{\delta} = 0.95$, and we group together the variables with high cross-correlation. Proceeding in this way we obtain the 2 groups

1.

| | | |
|---|---|---|
| `bypass duct stat.press.` | `compressor inlet tot.press.` | `compressor inlet tot.temp.` |
| `compressor inlet tip temp.` | `combustor inlet stat.press.` | `LP turbine frame stat.press.` |
| `combustor fuel flow` | | |

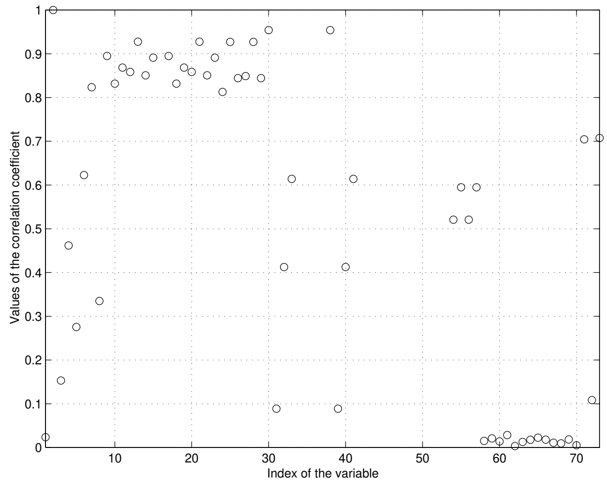


Figure 5: Correlation coefficients of all the variables with respect to thrust.

2. [ `LP turbine blade temp.  LP turbine exit temp.  temp. at combustor inlet`
`fan rotor speed`

and three isolated variables which are not highly correlated with any other variable: `core rotor speed, altitude, power code.` Next, we choose the variables within each group that are most highly correlated to thrust, and we get the following set of inputs for the estimator: `combustor fuel flow, fan rotor speed, core rotor speed, altitude, power code.`

Using our set of inputs[5] to train a neural network with five neurons on a training set of 1000 data, and testing it on a test set of 2000 points, we get the results in Table 3 when testing the estimation performance.

Table 3: Estimation results for thrust: 5 variables chosen as inputs.

| *% MAX* | *% MEAN* | *% MEDIAN* | $\% \, 2\sigma$ | $\% \, W(e)$ |
|---|---|---|---|---|
| 0.063 | 0.087 | 0.056 | 0.181 | 0.082 |

Comparing these results to the ones shown in Table 2, it appears quite evident that in order to have a significant computational reduction (only 5 inputs and 36 parameters in the approximator, versus 14 inputs and 209 parameters used previously), we apparently pay the price of a less accurate estimation.

Following the correlation analysis procedure, however, we easily overcome this problem: calculating the testing error after training the neural network, and estimating the correlation coefficient between this error and all the other variables, we get the results of Figure 6, which shows the correlation coefficient between the estimation error and the engine variables.

Here, it is observed that the error is significantly correlated with the measurable variables

[5] Notice that this is a subset of the variables chosen via intuition in Table 1.

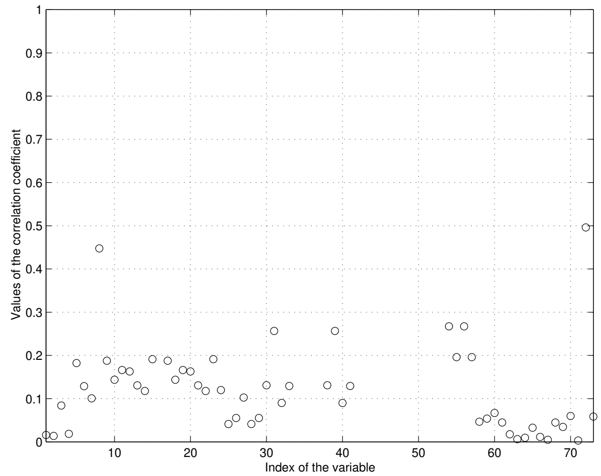


Figure 6: Correlation coefficients of all the variables with respect to the estimation error for thrust.

(we do not consider variables that we cannot measure) with indexes 39, 56, 72[6], corresponding to

`exhaust nozzle area   fan inlet tot.temp.   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 following set $\mathcal{S}$

| | | | |
|---|---|---|---|
| `combustor fuel flow` | `fan rotor speed` | `core rotor speed` | `altitude` |
| `power code` | `mach number` | `exhaust nozzle area` | `fan inlet tot.temp.` |

and the estimation performance results in Table 4, which are better than the ones shown in Table 2, with a significant reduction in the number of inputs (8 versus 14) and of the parameters (51 versus 209).

Table 4: Estimation results for thrust: 8 variables chosen as inputs using correlation analysis.

| % *MAX* | % *MEAN* | % *MEDIAN* | % $2\sigma$ | % $W(e)$ |
|---|---|---|---|---|
| 0.088 | 0.017 | 0.014 | 0.027 | 0.016 |

Now let us apply correlation analysis to the estimation of compressor stall margin, which is the variable with index $l = 4$. A plot of the correlation coefficients versus the variable index number is found in Figure 7.

[6]From Figure 6, it is noticed that the variables to take into consideration are the ones with indices 8, 31, 36, 54, 56, 72, but variable 8 is non measurable (therefore we cannot include it in our set of estimator inputs), and variables 39 and 56 are the measured values of variables 31 and 54, respectively. We then pick the sensed values of these two variables which, since our system is assumed to be noiseless, are identical to their nominal values.

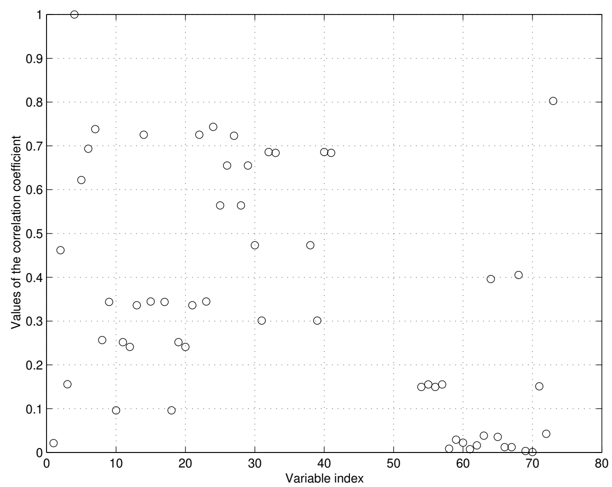


Figure 7: Correlation coefficients of all the variables with respect to stall margin.

Repeating the correlation analysis as before, we get $\mathcal{S}$ to be the following set of seven inputs:

```
compressor inlet tot.temp.   fan inlet tot.press.   combustor inlet stat.press.
combustor fuel flow          exhaust nozzle area    rear VABI
power code
```

with estimation results shown in Table 5. The results are almost indistinguishable from the ones

Table 5: Estimation results for stall margin: 7 variables chosen as inputs after correlation analysis.

| $MAX$ | $MEAN$ | $MEDIAN$ | $2\sigma$ | $W(e)$ |
|---|---|---|---|---|
| 1.05 | 0.29 | 0.25 | 0.41 | 0.27 |

of Table 2, 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.

### 3.3 Results: Discussion

The proposed procedure relies on the calculation of the correlation coefficient, defined in (4). Intuitively, the reason why the correlation analysis approach is successful in solving the estimator input selection problem is that the correlation coefficient provides an indication of the quantity of information shared by two random variables. This statement is rigorously true for random variables that are related to each other by means of a linear function. In the nonlinear case, however, conditions may be found for which this statement is false[7]. Thus, if the system is nonlinear, two variables with a low correlation coefficient are not necessarily unrelated, leading to the possible erroneous elimination of input variables in the first few steps of correlation analysis. The last three steps, however, are designed to avoid this problem, since input variables that have not been included in the first instance because of the wrong indication provided by a low correlation coefficient, should have a higher correlation coefficient with the estimation error. Of course, the analysis for the input-to-error correlation may suffer from the same nonlinearity problem, but the chances to include all the significant variables are significantly higher; in fact, the effectiveness of this technique has been confirmed by its successful application to other engines, providing a systematic tool for the solution of the input selection problem.

# 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. Our final objective is to develop a method to predict the estimation performance for a specific parameter and a particular sensor set, as a function of the operating condition, i.e., the triple (`altitude, mach number, power code`). 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[8].

- **Operating Space Partitioning:** Partition the operating space into local regions. The size of these regions should be small enough so that correlation analysis would capture the "observability" properties of the engine in a neighborhood of the operating point located at the center of a specific region. 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.

[7] Let $X$ be a Gaussian random variable with zero mean and unit variance, and $Y = X^2$. Then $\rho_{xy} = 0$, but $X$ and $Y$ are clearly strongly related to each other.

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

- **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 $E^{alt,mach,pc} = \left[D^{alt,mach,pc}, Y^{alt,mach,pc}\right]$, 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 (4) 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. This guarantees that each column of $c^{alt,mach,pc}$ is a vector representing the correlation between the corresponding unknown parameter and the measurable outputs of the engine associated to sensor set $\mathcal{S}$. Therefore, $c^{alt,mach,pc}$ contains $n_y$ correlation vectors, each of size $n$.

- **Correlation Measure:** Choose a function $\mu : \Re^n \to \Re$ 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 $\mu$ are 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

As opposed to the data based approach, which assumes the availability of a sufficient amount of engine data, 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.

### 4.2.1 Proposed Method

- **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:

$$o^k_{alt,mach,pc} = \frac{\left\|\bar{\mathcal{J}}^k\right\|_1}{\sum_{l=1}^{\bar{n}_y}\left\|\bar{\mathcal{J}}^l\right\|_1} \cdot \mathrm{cond}\left(\bar{J}_{alt,mach,pc}\right)^{-1}, \; k = 1, \dots, \bar{n}_y \tag{5}$$

$$\left\|\bar{\mathcal{J}}^k\right\|_1 \triangleq \sum_{i=1}^{n}\left|\bar{\mathcal{J}}^k\right|_i \tag{6}$$

$$\mathrm{cond}(\bar{J}_{alt,mach,pc}) = \sqrt{\frac{\lambda_{max}(\bar{J}^T\bar{J})}{\lambda_{min}(\bar{J}^T\bar{J})}} \tag{7}$$

where $\mathcal{J}^k, \; k = 1, \dots, \bar{n}_y$, is the $k$-th column of $\bar{J}_{alt,mach,pc}$ and $\left|\mathcal{J}^k\right|_i, \; i = 1, \dots, n$, is the absolute value of the $i$-th element in the above column. Notice that $\mathrm{cond}(\bar{J}_{alt,mach,pc})$, 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 $o^k_{alt,mach,pc}, k = 1, \dots, \bar{n}_y$ for each non-measurable parameter.

### 4.2.2 Motivations for the Choice of the "Observability" Index

The choice of the observability index $o^k$ defined above is motivated by the following mathematical considerations. Without loss of generality, let us drop the indices and consider a matrix $J = \frac{\partial \mathcal{S}}{\partial \mathcal{Y}}$, where $\mathcal{Y}$ and $\mathcal{S}$ are vectors containing non-measurable variables and measures belonging to the sensor set, respectively; $J$ can be estimated by means of finite differences from the simulation model. For small enough increments $\mathrm{d}\mathcal{S}$, the following identity holds:

$$\mathrm{d}\mathcal{S} = \left.\frac{\partial \mathcal{S}}{\partial \mathcal{Y}}\right|_{\mathcal{Y}=\mathcal{Y}_0} \cdot \mathrm{d}\mathcal{Y} = J \cdot \mathrm{d}\mathcal{Y} \tag{8}$$

in particular, this identity holds in a neighborhood of $\mathcal{Y}_0$. If $J$ $(n \times \bar{n}_y)$ had full column rank, and $n \geq \bar{n}_y$, then $J$ would be an "immersion" and the map $J : \mathbf{Y} \rightarrow \mathrm{span}(\mathbf{Y})$ would be injective[9]. Injectivity of this map would imply that $\mathrm{d}\mathcal{Y}$ can be recovered from $\mathrm{d}\mathcal{S}$, i.e., from a variation in the output measurements one would be able to estimate the correspondent variation in $\mathcal{Y}$. Since our objective is to study the "observability"[10] of each element of the vector $\mathcal{Y}$, we need to find a measure of the *degree of invertibility* of the linear map represented by $J$. Such a measure is naturally provided by the inverse of the condition number of the matrix $J$, i.e., the ratio

[9] $\mathbf{Y}$ denotes the space of the non-measurable variables, a subspace of $\Re^{\bar{n}_y}$

[10] In this analysis, we cannot refer to observability in the classical way since we are working with an engine in steady state. Therefore, by using "observability" of an engine variable, we refer to the possibility of estimating that variable using the available measurements from the sensors.

between its minimum and maximum singular values. Taking the inverse guarantees that the smaller the measure is, the less the map is invertible. In the worst case, when $\mathrm{cond}(J)^{-1}$ is zero, $J$ is not full rank, and the map is not invertible. In this situation, we say that $\mathcal{Y}$ is *not* "observable." On the other hand, if $\mathrm{cond}(J)^{-1}$ is high, then the matrix is well conditioned, which implies that it is far away from loosing rank and the inverse of the map can be calculated without numerical problems. In this situation, we say that $\mathcal{Y}$ is *quite* "observable." So far we showed that $\mathrm{cond}(J)^{-1}$ is a good candidate for the "observability" of $\mathcal{Y}$ as a whole, now we have to provide an estimation feasibility index for each component of this vector, independently.

Let us express $J$ in terms of its columns: $J = [\mathcal{J}^1, \mathcal{J}^2, \ldots, \mathcal{J}^{\bar{n}_y}]$; then, (8) can be written as

$$\mathrm{d}\mathcal{S} = \mathcal{J}^1(\mathrm{d}\mathcal{Y})_1 + \ldots + \mathcal{J}^{\bar{n}_y}(\mathrm{d}\mathcal{Y})_{\bar{n}_y} \tag{9}$$

where $(\mathrm{d}\mathcal{Y})_i$ indicates the $i$-th component of the increment $\mathrm{d}\mathcal{Y}$. From (9) we see that any variation of the sensor measurements is the sum of the columns of $J$, each weighted by the variation of the corresponding component of $\mathcal{Y}$. Thus, if the absolute values of the elements of the column $\mathcal{J}^k$ are bigger than that of other columns, a variation of the scalar $(\mathcal{Y})_k$ will be more "observable" from the output. Following this idea, an indication of the relative importance of the $i$-th component of $\mathcal{Y}$ is given by

$$\frac{\left\|\mathcal{J}^i\right\|_1}{\sum_{k=1}^{\bar{n}_y}\left\|\mathcal{J}^k\right\|_1}$$

In conclusion, we have that the "observability" index of $(\mathcal{Y})_i$ is given by the overall observability index weighted by the relative importance fraction described above, which gives the index in (5).

**Remark 1:** All the above discussion holds in an open neighborhood of the vector $\mathcal{Y}_0$, around which the linearization is performed. To take into account the fact that $\mathcal{Y}_0$ varies in $\mathbf{Y}$, the observability index is applied to a matrix $\bar{J}$ which is the result of averaging over $N$ linearizations around $N$ different points. This, together with the fact that each region $C_{alt,mach,pc}$ has been chosen small enough to try to get "uniform" observability properties, should guarantee that $o^k_{alt,mach,pc}$ captures the inherent nonlinearity of the system.

**Remark 2:** The uniform rank theorem states that a necessary condition for the nonlinear mapping between $\mathcal{Y}$ and $\mathcal{S}$ to be invertible on the compact set $U_S$, and therefore $\mathcal{Y}$ be "observable," is that the Jacobians $J$ be full rank for every $\mathcal{Y}_0$ in $U_S$. Hence, strictly speaking, the observability index in (5) does not provide a sufficient condition to guarantee injectivity of this mapping. Nevertheless, $o^k_{alt,mach,pc}$ should provide a reasonable indication of the degree of invertibility.

### 4.3 Relationship Between Data Based and Model Based Approaches

Estimation feasibility analysis, by means of the two proposed techniques, is intended to be easily applied to any engine. The two approaches, although different in nature, present some similarity. In the data based approach the "observability index" is calculated by means of the correlation coefficient between each measurements and unknown parameters. Given two random variables $X$ and $Y$, the correlation coefficient between $X$ and $Y$ is calculated by means of (3). Furthermore, the optimal linear mean square estimator (LMSE) of $Y$ is given by $Y = RX + B$ (see [14] for the proof), where

$$R = K_{xy}K_{xx} \triangleq E\left[(X - \bar{X})(Y - \bar{Y})\right]E[(X - \bar{X})^2] = \rho_{xy}\frac{\sigma_Y}{\sigma_X} \tag{10}$$

and $B = \bar{Y} - R\bar{X}$. Therefore, using the correlation coefficient between $X$ and $Y$ as a measure of the estimation feasibility for $Y$ is somewhat equivalent to linearizing $Y$ with respect to $X$ and measuring the slope of the line. If $X$ and $Y$ are vectors, the idea is generalized straightforwardly: $K_{xy} = E[(X - \bar{X})(Y - \bar{Y})^T]$, and the first identity in equation (10) still holds. Now $R$ represents a matrix, and looking at variations of $X$ and $Y$, we have

$$\mathrm{d}Y = R\mathrm{d}X \tag{11}$$

which is of the same form as (8), where $R$ is substituted by a partial derivative, which is a linearization of $S$ around $\mathcal{Y} = \mathcal{Y}_0$. In the data based approach the "observability" index is, roughly, a function of the rows of $R$, whereas in the simulation model based technique the index depends on the condition number of the matrix. In conclusion, the two approaches, although conceptually very similar, differ in that the model based technique is applied after an averaging process carried on $N$ partials, and the estimation feasibility index derived in the two cases is different.

### 4.4 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

$$\mathcal{Y} \approx R\mathcal{S}, \qquad R = Y^{alt,mach,pc^T} D^{alt,mach,pc}\left(D^{alt,mach,pc^T} D^{alt,mach,pc}\right)^{-1} \text{ for all engines in } C_{alt,mach,pc} \tag{12}$$

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.

### 4.5 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 the following set of sensors of the XTE46 engine:

| | | |
|---|---|---|
| `bypass duct stat.press.` | `compressor inlet tot.press.` | `compressor inlet tot.temp.` |
| `compressor inlet tip temp.` | `combustor inlet stat.press.` | `LP turbine blade temp.` |
| `LP turbine frame stat.press.` | `LP turbine frame exit temp.` | `fan rotor speed` |
| `core rotor speed` | `LP exit tot.press.` | `compressor discharge temp.` |
| `bypass duct tot.press.` | `core bypass stage inlet stat.press.` | `CDFS-tip inlet press.` |
| `bypass duct pressure` | `inter-turbine temp.` | `inter-turbine press.` |

where CDFS stands for Core-Driven Fan Stage.

This set represents the totality of 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 4.4. As for the model based approach, for each cubic region, we approximate $N = 30$ Jacobians and we calculate the "observability" index in (5). 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 8, 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 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 exam-

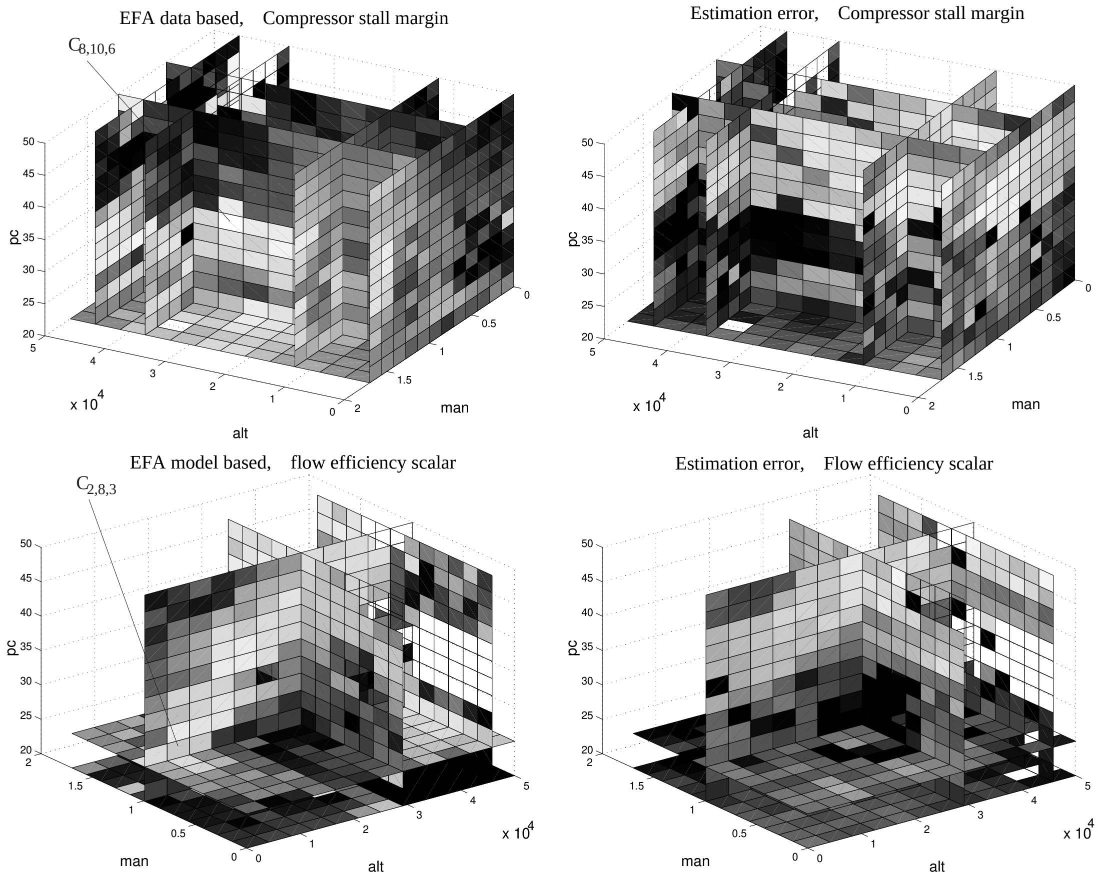


Figure 8: Comparison between EFA and estimation performance.

ples show the advantage of nonlinear over linear estimation, confirming the information provided by EFA.

# 5 Conclusions

This paper highlights the importance of input selection and estimation feasibility analysis as tools for estimator design in complex dynamical systems. In these concluding remarks we would like to highlight some possible drawbacks of the methods proposed here. A set of techniques has been proposed to solve this problem; however, the complexity of the systems we are dealing with, together with the lack of mathematical models, do not allow for an analytical study of the properties of these methods.

Correlation analysis, though easy to implement and successful in its application, cannot be guaranteed to pick the optimal set of inputs to the estimators. The same holds for data based and simulation model based approaches to estimation feasibility analysis, where the available information from the engine is used to construct indices aimed at providing an *indication* of observability.

Variations and future improvements to these methods may be developed: the objective of this paper is to formulate the problem and devise practical alternatives that a designer could employ for its solution.

Thrust and stall margin estimation results, together with the estimator redesign examples provided in Section 4.5, show the promising features of these techniques. Whether these results can be fully generalized to other applications is an open question.

Finally, these techniques are introduced in a *static* framework (i.e., we study the engine in steady state), and future research directions might include their extension to the dynamic case.

# 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] J. Horlock, *Axial flow compressors: Fluid mechanics and Thermodynamics.* London: Butterworths scientific publications, 1958.

[3] E. Greitzer and H. Griswold, "Compressor-diffuser interaction with circumferential flow distortion," *The journal of mechanical engineering science*, vol. 18, no. 1, pp. 25–43, 1976.

[4] S. Billings and Q. Zhu, "Non-linear model validation using correlation tests," *Int. J. of Control*, vol. 60, pp. 1107–1120, 1994.

[5] S. Billings and Q. Zhu, "Model validation tests for multivariable non-linear models including neural networks," *Int. J. of Control*, vol. 62, pp. 749–766, 1995.

[6] S. Chen, S. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. of Control*, vol. 50, pp. 1873–1896, 1989.

[7] A. Lorber, L. E. Wang, and B. R. Kowalski, "A theoretical foundation for the PLS algorithm," *Journal of Chemometrics*, vol. 1, pp. 19–31, 1987.

[8] C. Chen, *Linear System Theory and Design.* New York: Holt, Rinehart, and Winston, 1984.

[9] H. Nijmeijer and A. van der Schaft, *Nonlinear Dynamical Control Systems.* New York: Springer Verlag, July 1990.

[10] R. Marino and P. Tomei, *Nonlinear Control Design: Geometric, Adaptive, and Robust.* Prentice Hall, 1995.

[11] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design.* PWS Publishing Company, 1996.

[12] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, no. 2, pp. 303–314, 1989.

[13] 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.

[14] H. Stark and J. Woods, *Probability, Random Processes, and Estimation Theory for Engineers.* Prentice Hall, 1994.