Fuzzy Sensor Validation and Fusion for Gas Turbine Power Plants

Kai Goebel
GE Global Research
Information & Decision Technologies

Alice Agogino
Department of Mechanical Engineering
University of California at Berkeley

Abstract
This paper introduces a fuzzy sensor validation and fusion methodology and applies it to sensor data from a gas turbine power plant. The FUSVAF (Fuzzy Sensor Validation and Fusion) algorithm tackles two problems: (1) It filters out noise from measurements provided as input to a gas turbine controller to maximize power plant performance. (2) It compensates for sensor failure to allow the power plant to operate despite temporary or permanent failure of one or more sensors. Using a combination of direct and functional redundancy the fusion algorithm determines confidence values for each sensor reading from nonlinear validation curves, which progressively discount readings with increasing distance from the predicted value. The predicted value in the FUSVAF algorithm is obtained through application of a Fuzzy Exponential Weighted Moving Average (FEWMA) time series predictor with adaptive coefficients. Simulations and a sensitivity analysis are used to validate the fuzzy rules and show the robustness of the chosen parameters. Experiments on operating data from a gas turbine in an electric power generating plant show how an increase in redundancy improves the robustness of the FUSVAF algorithm and leads to smoother controller input values.

Fuzzy Sensor Validation and Fusion for Gas Turbine Power Plants

Kai Goebel
GE Global Research
Information & Decision Technologies

Alice Agogino
Department of Mechanical Engineering
University of California at Berkeley

1 Introduction

The motivation to apply validation and fusion techniques to power plants is to keep the system running at peak performance, improve operating safety, and avoid unwarranted shutdowns [1]. The latter are usually accompanied by costly routine maintenance procedures and lag time before the power plant can be fired up again. To avoid unnecessary shutdowns a system must be able to distinguish between sensor failure and system malfunction. This distinction must be made at a level above the machine control level, such as by a supervisory controller [2]. Furthermore, if readings are corrupted by noise and degraded sensor performance, the system must filter out noise and effects of sensor failures [3]. The FUSVAF (Fuzzy Sensor Validation and Fusion) algorithm uses readings obtained from direct redundant sensors and through functional redundancy [4]. These values are then fused to render a corrected value which can be used for control purposes. In the gas turbine power plant application described herein, the controller attempts to run the gas turbine at a peak temperature subject to the limitations imposed by the material properties of the combustion chamber. It is important to keep the temperature in the combustion chamber at the maximum allowable point. However, this temperature can not be measured directly and must be calculated by other indirect measurements. Furthermore, these indirect measurements are subject to a number of internal and external disturbances such as changes in ambient temperature and pressure as well as aerodynamic disturbances, internal pressure fluctuations, erosion, corrosion, vibration impingement, flutter, combustion instabilities, material failure, etc. [5,6,7]. Thus sensor fusion and validation can play an important role in improving the integrity of the sensor data [8] and efficiency of the operation of the gas turbine.

2 Methodology

This section describes the FUSVAF sensor validation and fusion architecture and the theoretical underpinnings of the FUSVAF algorithm.

2.1 Fuzzy Sensor Validation and Fusion Architecture

The FUSVAF algorithm for fuzzy sensor validation and fusion makes use of a Fuzzy Exponential Weighted Moving Average (FEWMA) time series predictor, dynamic validation curves [9] which are determined by sensor characteristics, and a fusion scheme which uses confidence values for the measurements, the predicted value, the measurements, and the system state [10]. Inputs to the FUSVAF algorithm are the raw sensors measurements. Output is the corrected value which can be used for the machine level controller as well as for supervisory control tasks. Additional information regarding the performance of individual sensors or actuators can be provided for use in a diagnostic module (e.g., [11]). Fig. 1 shows the architecture of the FUSVAF algorithm.

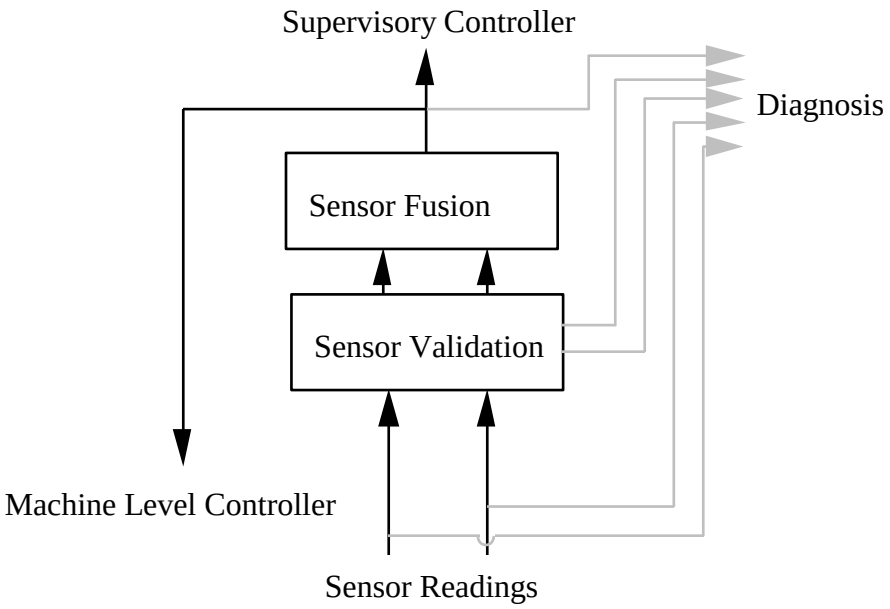


Fig. 1: Architecture for Sensor Validation and Sensor Fusion

2.2 The FUSVAF Algorithm

The validation and fusion algorithm works in the following manner: incoming sensor readings are validated within a validation gate using the fused value from the previous time step. A newly fused value is then used to assess the state of the system expressed by α. It is also used for prediction which in turn is necessary to perform the validation for the next time step. The fused value is also used for the machine level controller and supervisory control tasks. The algorithm is displayed in Fig. 2 where z^{-1} denotes a unit delay.

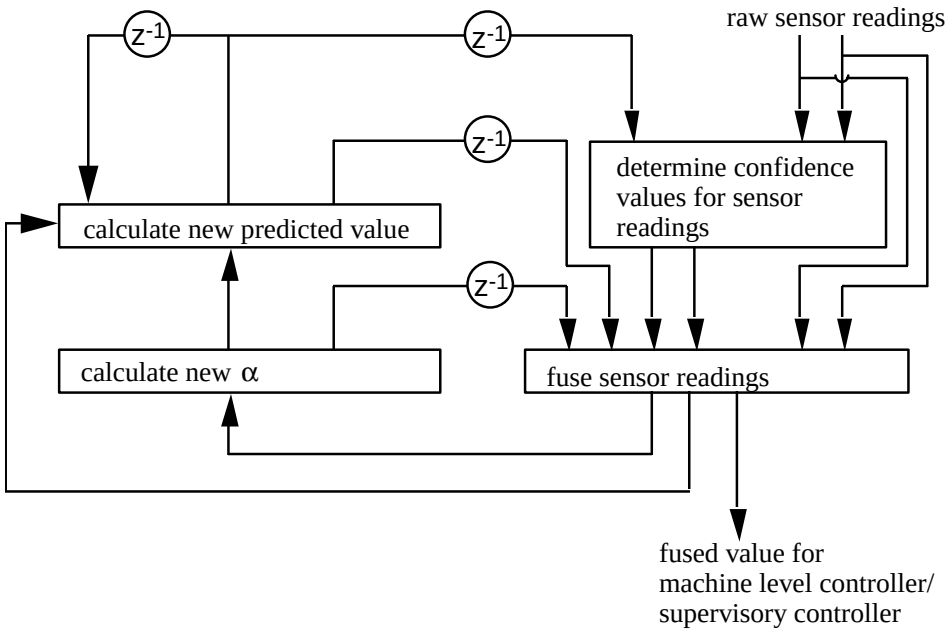


Fig. 2: FUSVAF Algorithm for fuzzy sensor validation and fusion

The confidence values that are assigned to all sensor measurements depend on the specific sensor characteristics, the predicted value, and the physical limitations of the sensor value. The assignment takes place in a validation gate that is bounded by the range of physically feasible values of the system in the new time step. If sensor readings show a change beyond that feasible limit, the readings are flagged as erroneous and are assigned a confidence value of '0' [12]. Within the region, a maximum value of '1' will be assigned to readings that coincide with the predicted value. The curve between the maximum and the two minima is dependent on the sensor model [13]. Generally, this is a non-symmetric curve which is wid around the maximum value if the sensor is known to have little variance and narrow if the sensor exhibits noisy behavior. The curves change dynamically with the operating conditions, capturing the change in behavior of the sensor over its operating span. Other effects, such as environmental conditions, resulting in changing sensor performance can be integrated into the validation curves as well. An example for a functional relation which captures the notion of changing confidence with increasing range is

$$a_i(z) = \frac{a_{i_0}}{z} \tag{1}$$

where

z is the measurement

a_i determines the width of the validation curve for which examples are listed below.

In this formulation, a sensor with a relatively low variance and thus higher confidence level should have a large value of the parameter a_{i_0}. One possible parametric validation curve v(z) for a particular situation could be a piecewise bell curve of the form

$$v(z) = e^{-\left(\frac{\hat{x}-z}{a_w}\right)^2} \tag{2}$$

where parameters a_w have to be chosen separately for the left and right curves. The curves then need to be scaled from 0 and 1 between the validation border and the predicted value $\hat{x}$. The resulting assignment for confidence values is

$$\sigma = \begin{cases} 0 & z < v_{left} \\ \dfrac{e^{-\left(\frac{\hat{x}-z}{a_{left}}\right)^2} - e^{-\left(\frac{\hat{x}-v_{left}}{a_{left}}\right)^2}}{1 - e^{-\left(\frac{\hat{x}-v_{left}}{a_{left}}\right)^2}} & v_{left} < z \le \hat{x} \\ \dfrac{e^{-\left(\frac{\hat{x}-z}{a_{right}}\right)^2} - e^{-\left(\frac{\hat{x}-v_{right}}{a_{right}}\right)^2}}{1 - e^{-\left(\frac{\hat{x}-v_{right}}{a_{right}}\right)^2}} & \hat{x} < z \le v_{right} \\ 0 & z > v_{right} \end{cases} \tag{3}$$

where

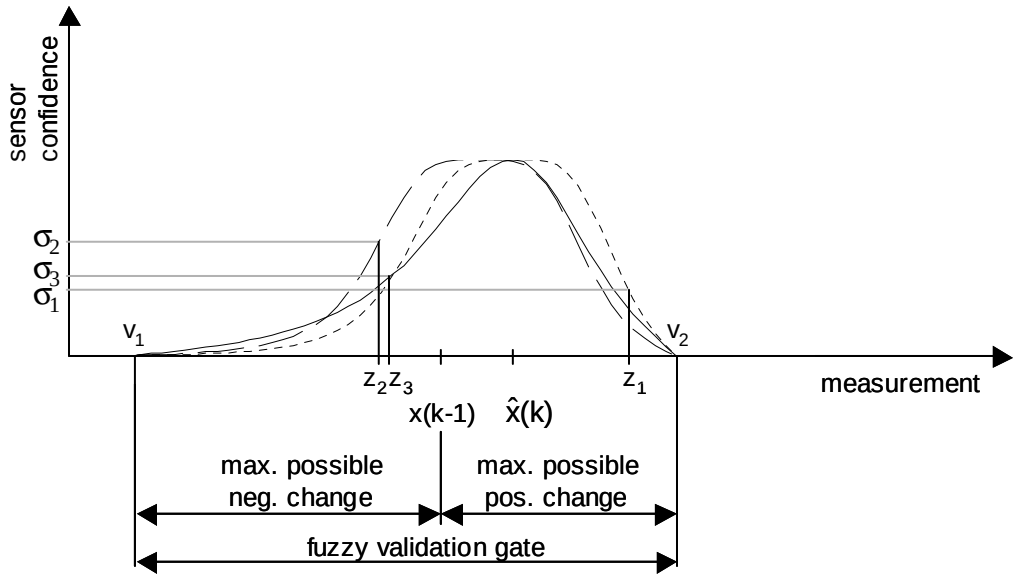
$\sigma(z_i)$ is the confidence value for a particular sensor with measurement z_i

v_{right} and v_{left} are the right and left validation gate borders, respectively

a_{right} and a_{left} are the parameters for the left and right validation curve

z is the sensor reading
$\hat{x}$ is the predicted value.

A validation gate is shown in Fig. 3. The three curves represent validation curves for three different sensors. z_1, z_2 and z_3 are readings of these three sensors at the same time instance *k*.



z_i sensor measurements
σ_i sensor confidence values
$\hat{x}(k)$ predicted value
x(k-1) old value at previous time step
v_i validation gate boundaries

Fig. 3: Validation gate for the assignment of confidence values

The fusion is performed through a weighted average of confidence values and measurements plus a term which includes the predicted value weighted by α (which represents the system state). The sum of the confidence values times the measurements is constructed to reward measurements closest to the predicted value, depending on the shape of the validation curve. Measurements further away are discounted. The confidence value drops to 0 at the border of the validation gate so that measurements beyond the gate borders are not taken into account. The validation gate boundaries v_1 and v_2 are calculated dynamically at each time step to reflect the maximum possible change the system may exhibit during one time step. These maximum changes will also vary in situations of abnormal behavior such as emergency conditions. The operative equation of the FUSVAF algorithm is

$$\hat{x}_f = \frac{\sum_{i=1}^{n} z_i \sigma(z_i)}{\sum_{i=1}^{n} \sigma(z_i)} \qquad (4)$$

where

$\hat{x}_f$: fused value

z_i: measurements
σ: confidence values

Note that if all sensors lie on one side of the predicted value, the fused value will also be pulled to the same side. This ensures that evidence from the sensors is closely followed yet discounted as the value moves further from the predicted value.

To prevent the system from becoming unstable when no valid readings are obtained, a scaling factor ω can be introduced into equation (4) which includes, to a small fraction, the predicted value. This permits the calculation of fused values even when all sensors have temporarily invalid readings. Furthermore, an additional parameter is included with information about the state of the system through the use of the adaptive parameter α. If the system is in steady state, α is large and past history is more heavily weighted. When the system is in a transient state, however, α is small and the predicted value has less weight, thus reducing the lag induced by the term. The calculation of α will be explained with the FEWMA algorithm later. The scaling factor ω has to be tuned to the system at hand but it is typically large to prevent the predicted value from dominating the sensor readings. The purpose of this term is only to deal with rare situations where no reading falls inside the validation region. This expanded equation is

$$\hat{x}_{\mathrm{f}} = \frac{\sum_{i=1}^{n} z_i \sigma(z_i) + \frac{\alpha \hat{x}}{\omega}}{\sum_{i=1}^{n} \sigma(z_i) + \frac{\alpha}{\omega}} \tag{5}$$

where

α: adaptive parameter representing the system state
ω: constant scaling factor

The motivation for using a time series predictor with an adaptive parameter in the design of time series prediction stems from the desirability of tuning the trade offs between responsiveness, smoothness, stability, and lag of the predictor for the needs of specific applications. The standard exponential weighted moving average predictor has the form

$$\hat{x}(k+1) = \alpha \hat{x}(k) + (1-\alpha) z(k) \tag{6}$$

If the parameter α is set to a fixed value the degree to which new information from sensor readings is used to update the system state is also fixed. There is a danger that the predictor will lag behind the true state too much for large constant values of α or that the predictor will follow the data so closely that noise is not filtered out when α is too small. Generally there are two states of the system for which a different design for α would be desirable. If the system is in a more or less steady state, then noise filtering should be the primary task. If, on the other hand, the system is in some sort of transient state, it is more important to track the system change [14]. Therefore,

α should be adaptive according to the system change. In particular, α should be large when the system is in a steady state and it should be small when the system is in a transient state. These linguistic statements can be interpreted as fuzzy rules. Assuming that noise is smaller than the change of the system, then, if the changes are small, the system is likely to be in a steady state. Otherwise, the system is in some dynamic (or transient) state. Thus, the following fuzzy rules can be formed:

- *IF change of readings "small" THEN α "large"*
- *IF change of readings "medium" THEN α "medium"*
- *IF change of readings "large" THEN α "small".*

For the design of the membership functions "small", "medium", and "large" we used triangular shaped functions with maximum overlap such that only two parameters have to be specified: one for the fuzzification and one for the defuzzification [14]. The functions are shown in Fig. 4.

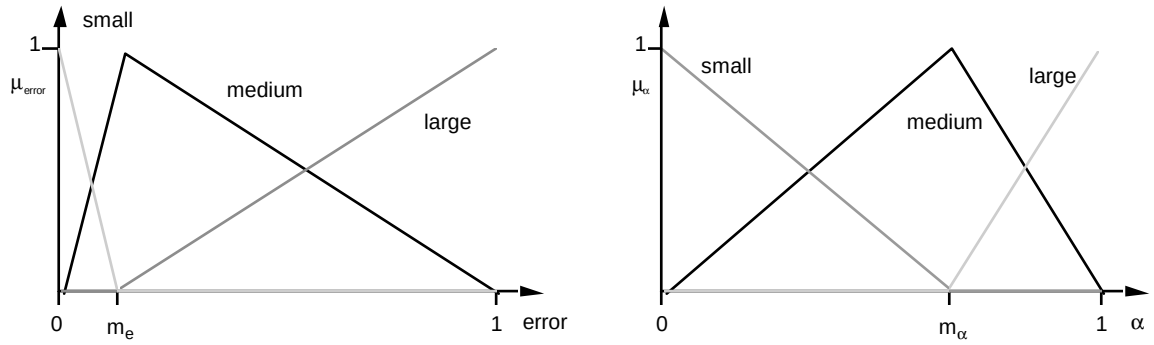


Fig. 4: Membership functions for fuzzy rules [14]

The parameter m_e determines the vertex of the triangular membership function, μ_{error}, "medium error". This parameter also sets the point where membership functions "small error" and "large error" change their shape, i.e., where the membership function for "small error" reaches 0 and changes course to a straight line with value 0. The membership function for "large error" changes from 0 to an increasing line. Similarly, the parameter m_α defines the shape of the membership functions, μ_α, for small, medium, and large α.

2.3 Simulations

To evaluate the performance of the FEWMA and FUSVAF, we simulated systems with different degrees of noise. We used the membership functions as described above and used a center of average defuzzification method. It must be noted that the granularity of the results, in particular the sensitivity to system changes and smoothing capabilities will depend on the number and shape of the membership functions as well as the defuzzification method shown. However, conceptually, this simple setup will suffice to demonstrate the operation of the FEWMA. Fig. 5 shows a system with no noise which starts in steady state, then undergoes a sinusoidal change, and ends again in steady state. The FEWMA adapts more rapidly to the changes than a predictor with large fixed α (α =0.99; SSE=1273.70) and it has a smaller Sum of Squares for Error (SSE=410.02). A predictor with small fixed α (α =0.01) has the smallest overall SSE

(SSE=80.78). Fig. 5b also shows the adaptation of α which decreases for the transitional period, then goes back to a larger value (cutoff at α=0.99).

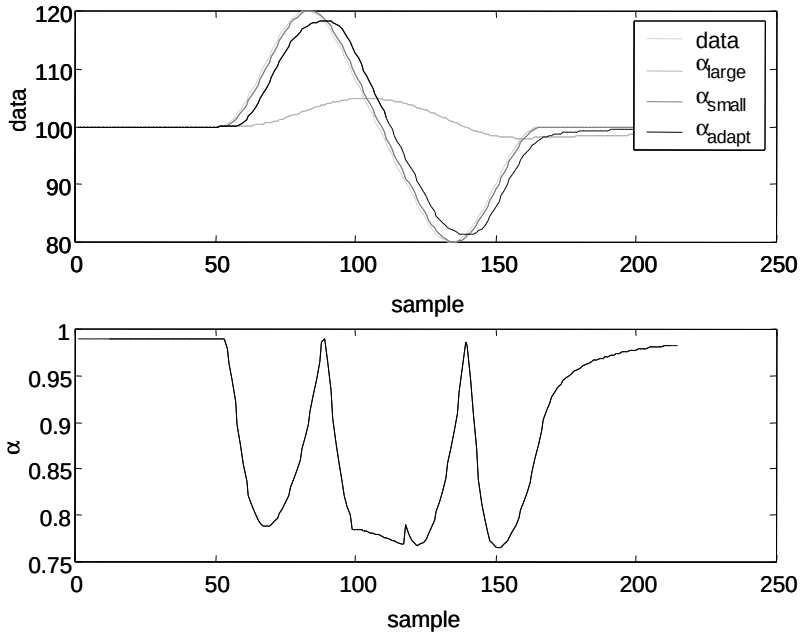


Fig. 5: Simulation for FEWMA with no noise

Fig. 6 shows a system with Gaussian noise with 1 standard deviation. Note that α reacts to the noise resulting in variation of the values for α. A more refined choice of membership functions and defuzzification method could be used to counteract this phenomenon. However, the general pattern of Fig. 5b can still be observed. Also to be noted is the standard deviation of the output which at steady state is almost an order of magnitude less (std_{adapt_α}=0.40, SSE=389.73) than the predictor with small α (std_{small_α}=1.21; SSE=197.06). The predictor with large α had an error of SSE = 1,284.8 and a standard deviation of std = 0.06.

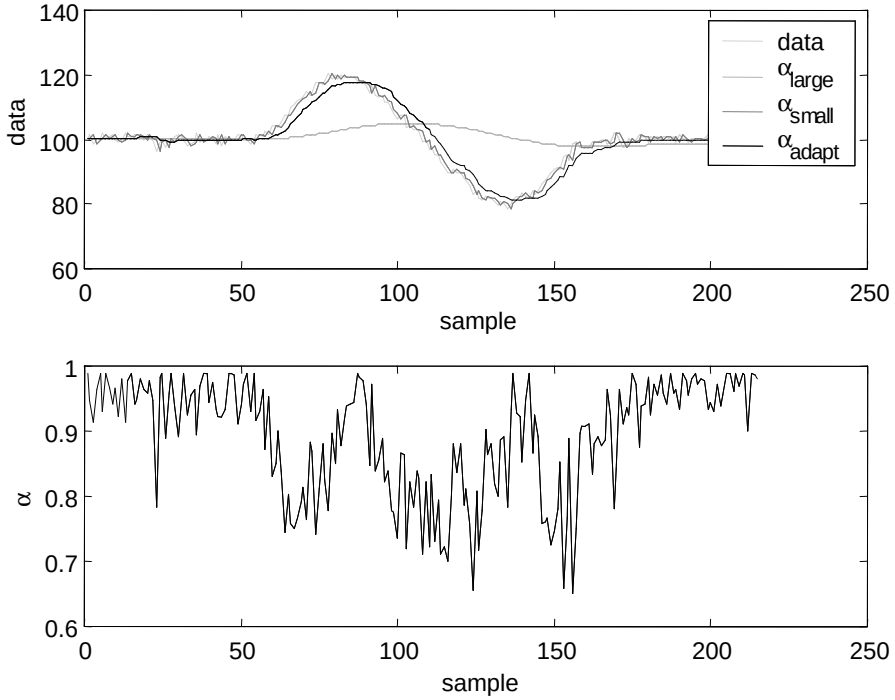


Fig. 6: Simulation for FEWMA of noisy system

To show the operation of the FUSVAF, we simulated two data streams, *reading*$_1$ and *reading*$_2$. The values of *reading*$_1$ were generated with varying noise between standard deviation 0 and 2.24 and *reading*$_2$ with noise of standard deviation 1 and random outliers of magnitude 20. The parameters for the FEWMA were chosen as before. In addition, we used two other fusion schemes, an averaging scheme with outlier protection logic eliminating outliers over 2 standard deviations, and a voting scheme which chooses the value closer to the previously one chosen. Both of these benchmarking tools were tuned to avoid unstable operation and maximum performance. Fig. 7 shows the two raw readings and the fused values. In particular, Fig. 7a shows the raw readings and Fig. 7b through Fig. 7d show the results of the averaging scheme, the voting scheme, and the FUSVAF respectively. It can be seen by visual inspection that the FUSVAF filters out noise and rejects outliers the best. Numerical results of both overall SSE and standard deviation (measured during the steady state period) confirm that the FUSVAF operates better than the other two tools. The averaging scheme had a steady state standard deviation of 2.51, the voting had a steady state standard deviation of 1.17, and the FUSVAF had a steady state standard deviation of 0.83. The SSE, which was computed over the entire range, was 11714 for the averaging scheme, 10,647 for the voting scheme, and 312 for the FUSVAF. These results are summarized in TABLE 1. Some residual noise can be observed in the FUSVAF which is a function of its weighted averaging operation. With the settings chosen, the FUSVAF defaulted to the FEWMA calculated value at samples 81 and 97, where both raw readings are beyond the

validation gate boundaries. That is, the logic of the additional term in equation (4) is used to protect the system from unstable operation.

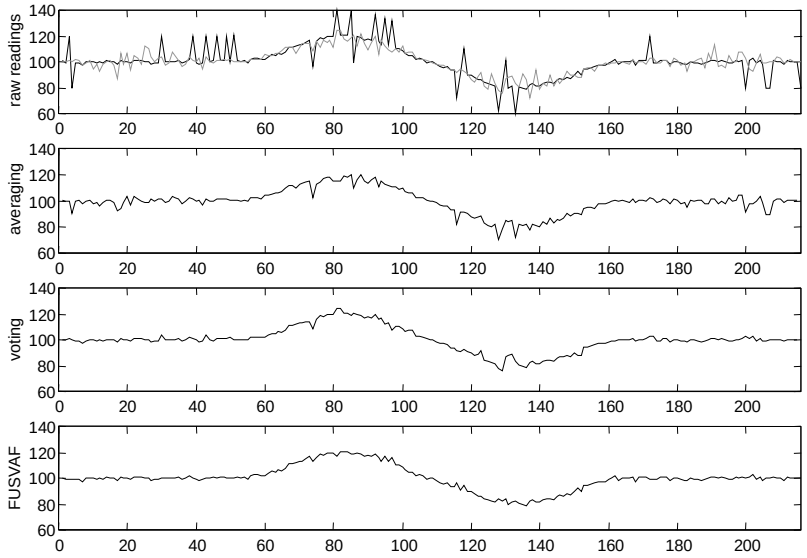


Fig. 7: Two simulated readings (a) and fused value obtained using averaging scheme (b), voting scheme (c), and FUSVAF (d)

TABLE 1

Comparison of Fusion Schemes

	averaging scheme with outlier protection	voting scheme	FUSVAF
SSE	11,714	10,647	312
standard deviation (steady state)	2.51	1.17	0.83

Parameter tuning is an issue in any real world application. Here, there are two parameters (a_1 and a_2) for each sensor, two parameters for the FEWMA (m_e and m_α, assuming the fuzzification and defuzzification method presented herein), and one parameter for the FUSVAF (ω, this parameter is optional, depending on whether equation (4) or equation (5) is employed), bringing the total number of parameters to seven. In the simulations, we used a symmetric validation curve, thus cutting the number of sensor parameters in half, resulting in five parameters total. The values were then established empirically without much effort. We performed a sensitivity analysis which demonstrates that the parameters are not sensitive to change in a wide region.

Parameters m_e and m_α should be tuned so as to optimize performance for a particular application by using operating data over a wide range of operating conditions and simulations over scenarios with sparse data. Our algorithm was tested on operational data where possible. In cases where sensor failure data were not available, we developed sensor models and derived simulated data

from these models. We initialized m_e and $m_\alpha$to 0.2 and used genetic algorithms [15] to fine tune the parameters. Since we did not note a marked performance improvement, we will not report details of this optimization process and instead show with a sensitivity analysis in the next section that the performance profile is relatively stable for the choice of parameters.

2.4 Sensitivity Analysis

We also performed sensitivity analyses over variations in Gaussian noise, outliers, bias, and clutter which were performed to better evaluate how the error changes with the initial value of the parameters. Fig. 8 shows the sensitivity analysis for m_e vs. m_α in the presence of Gaussian noise, clutter, random and steady bias, and outliers. Parameter $m_\alpha$is more receptive to change than m_e. In particular, m_α is found to be best at low values. If it is too large, the defuzzification method chosen produces values for α which are too large to properly respond to system changes. Although m_e is not very receptive to changes, the best values for m_e are found at higher values

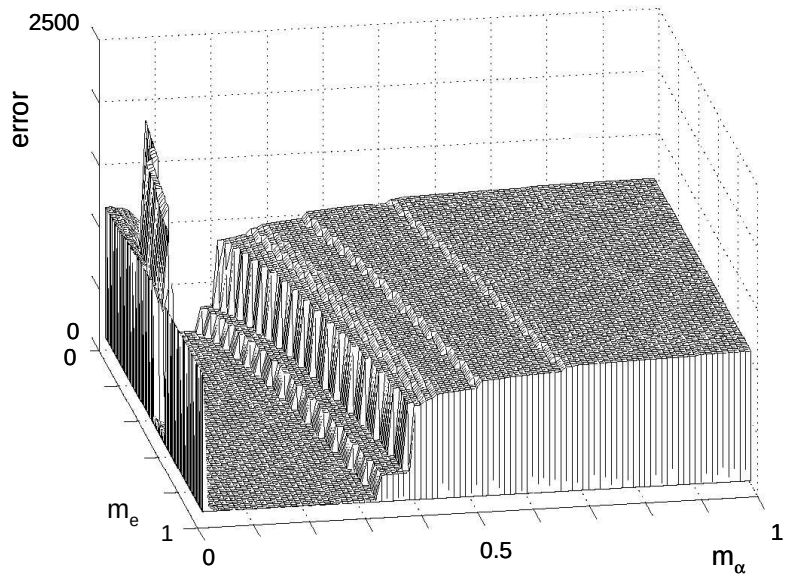


Fig. 8: Sensitivity analysis for m_e/m_α: system subject to outliers, bias (random and steady), clutter, and Gaussian noise

Fig. 9 shows the sensitivity analysis for a_1 and a_2. The algorithm is unstable only for very small values of a_1 and a_2. This means that any reasonably large value is a good starting value from which an optimal value can be found. Small values for a_1 and a_2 mean a small basis of the validation curve. If they are not big enough, the incoming readings will be assigned very small degrees of memberships, causing the old readings to be weighted too heavily. In other words, disregarding new information results in an unstable algorithm.

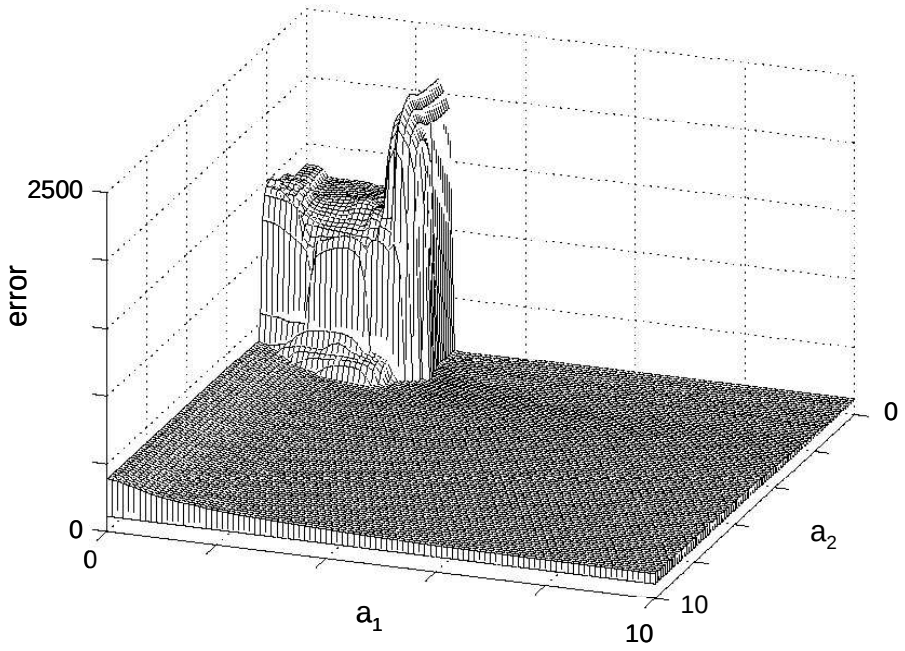


Fig. 9: Sensitivity analysis for a_1 and a_2

The sensitivity of the scaling factor ω of equation 5 (which determines to which degree the old value is integrated into the new estimate) is displayed in Fig. 10.

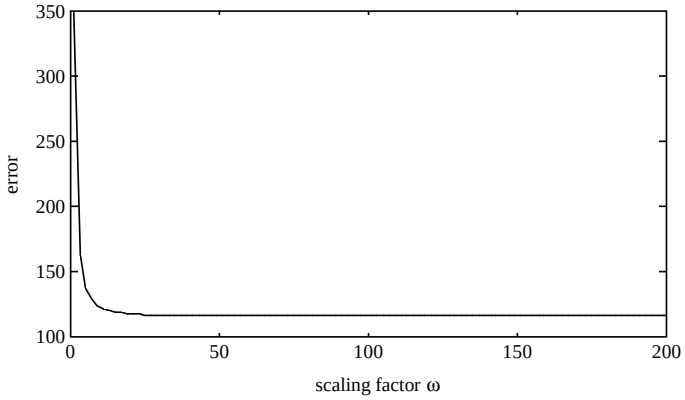


Fig. 10: Sensitivity analysis of the scaling factor ω

For small values of the scaling factor ω, large errors are produced. As ω increases the error sharply drops to a minimum and then slowly increases. Not visible on this curve is the flattening of the increase for large values of ω. For initial values, a large scaling factor is recommended. If too small values are used, new incoming information is discarded, resulting in a lag of the fused value because the old information is weighted too heavily in the new value. Further reduction of

ω can result in periods of unstable algorithm behavior in which case there are periods of constant output because no new value comes close to the old ones and the validation gate discards new information.

3 Application to Gas Turbines

A gas turbine consists of components inlet system, compressor, combustor, turbine, and exhaust system as shown in the diagram in Fig. 11.

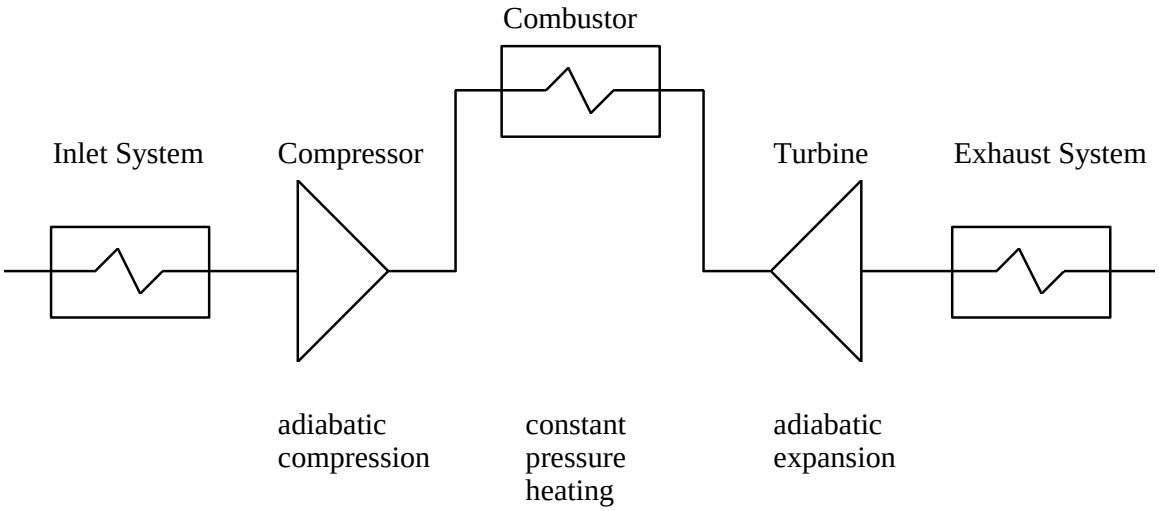


Fig. 11: The gas-turbine process

The basic process of a gas turbine is the Brayton cycle. The ideal Brayton cycle involves isentropic compression in a compressor (no heat exchange), constant pressure heating (burning) in the combustion chamber, and isentropic expansion through a turbine. Fig. 12 shows the enthalpy-entropy diagram of a Brayton cycle. While the ideal Brayton cycle consists of two vertical isentropics and two constant pressure lines in real life there are efficiency losses within the compressor and the turbine which are represented by the sloped lines. Moreover, there are pressure losses in the inlet, the combustion, and the exhaust which means that the lines from 2 - 3 and 4 - 1 are not true constant pressure lines. Furthermore, the mass flow rate around the cycle diagram is not constant either because air is introduced and extracted for cooling purposes which affect the total work input and output of various processes in the cycle. Therefore, the enthalpy-entropy diagram and its accompanying equations are only an approximation of the real quantities. The net work W_{net} of the cycle is the difference between the work obtained in the expansion process W_t and the work of the compression W_c. The efficiency of the whole process is calculated as the total output divided by the input. Gas turbines achieve efficiencies in the mid 30% range.

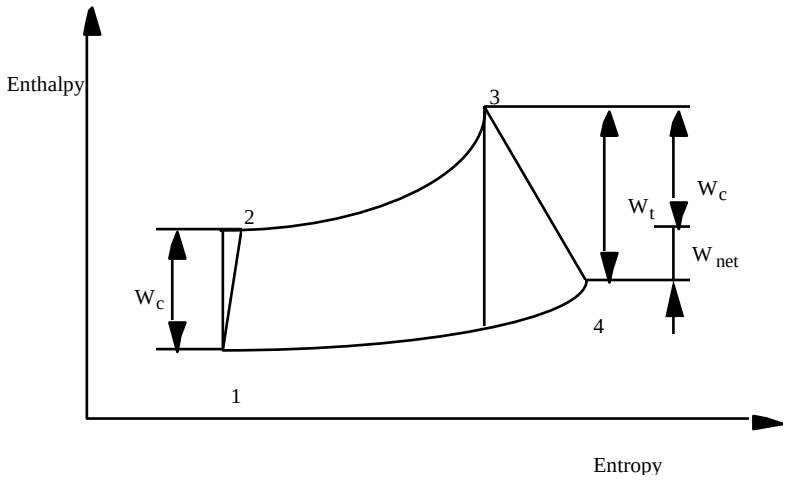


Fig. 12: Enthalpy-entropy diagram of the Brayton cycle

Several methods exist to further improve the efficiency of the process. Heat from the exhaust system can be reclaimed through heat exchanges, which in turn can be used to heat the working fluid during expansion. In addition, the compression phase is undertaken in several steps with intercooling in between.

The goal is to operate the turbine at maximum allowable temperature T_3 which cannot be measured directly. Instead, the control algorithm uses measurements of other variables in the power plant to control the relevant parameters. The compressor discharge pressure p_2 is often used in control algorithms for gas turbine driven power plants. p_2 can be measured directly. However, the pressure sensor is subject to noise which has a direct influence on the performance of the control algorithm and ultimately on the load generated. The noise is partly of Gaussian nature but can also be non-Gaussian. The FUSVAF deals particularly well, when compared to other algorithms, with the latter type. To give a better understanding of the types of noise encountered, the following figures (Fig. 13 - Fig. 16) show examples of observed noise. Data were also simulated using sensor models which mimic the behavior found in the data of the real gas turbine power plant [16]. Fig. 13 shows typical readings of a compressor discharge pressure sensor modeled on data obtained from an extant gas turbine power plant of the 40MW class. Data shown were observed over a period of 18 hours. The sampling time was about 5 sec. The data change slowly and have superimposed Gaussian noise. The slow changes can be attributed to changes of ambient temperature and pressure throughout the day which affect also the density of the inlet air. When observed over a 24 hour period, p_2 goes through one slow cycle. There is another even slower cycle with frequency 1 cycle/year which reflects the change of the mean ambient temperature over the seasons. This means the mean compressor discharge pressure is different when observed at different days in the year. Similarly, weather conditions may have an influence in the readings as well. The control algorithm of the power plant reacts to these changes to keep the power plant at maximum performance. To achieve this goal, noise has to be filtered out and replaced with the most accurate estimate of the temperature possible. Measurements from sensors measuring different variables can be used to compute the control input value and can be fused to achieve smoother and more robust control. Since all sensors are

subject to noise, this noise must be eliminated for each sensor. The sensors are listed in TABLE 2.

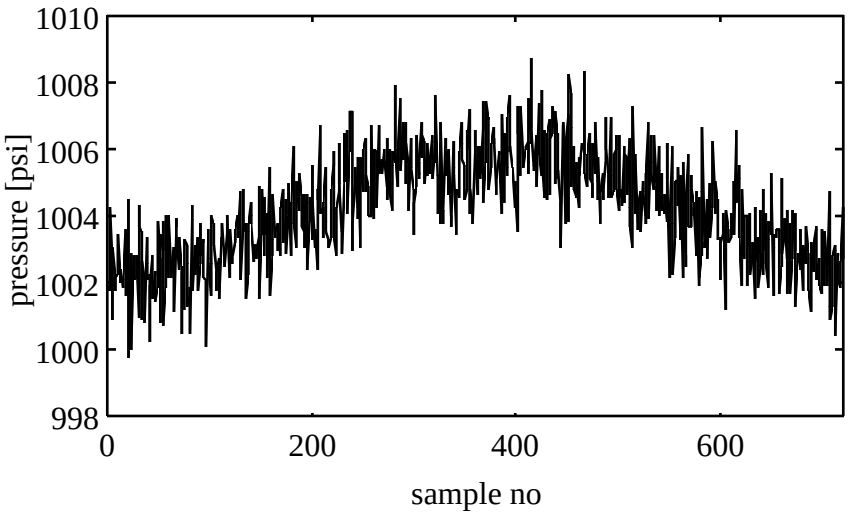


Fig. 13: Discharge pressure sensor p_2 over 18 hours

TABLE 2

Sensors used for fusion

Quantity measured	Symbol
Exhaust Temperature	T_4
Compressor Inlet Flange Temperature	T_{11}, T_{12}
Compressor Discharge Temperature	T_{21}, T_{22}
Compressor Discharge Pressure	p_2
Load generated	W
Fuel Flow	$\dot{m}$
Inlet Duct Manifold Pressure	p_i
Ambient Barometric Pressure	p_a

The inlet duct manifold pressure and the ambient barometric pressure are together used to give the inlet pressure $p_1 = p_i - p_a$.

The following figures display the operation of the various sensors used over a period of 18 hours. The sampling time was 5 sec. Fig. 14 shows the measured compressor inlet flange temperatures T_{11}, T_{12}. Note the quantization effect due to round off in the data acquisition that makes the temperature T_{11} and T_{12} appear to jump between values. Here, mere averaging algorithms do not perform very well.

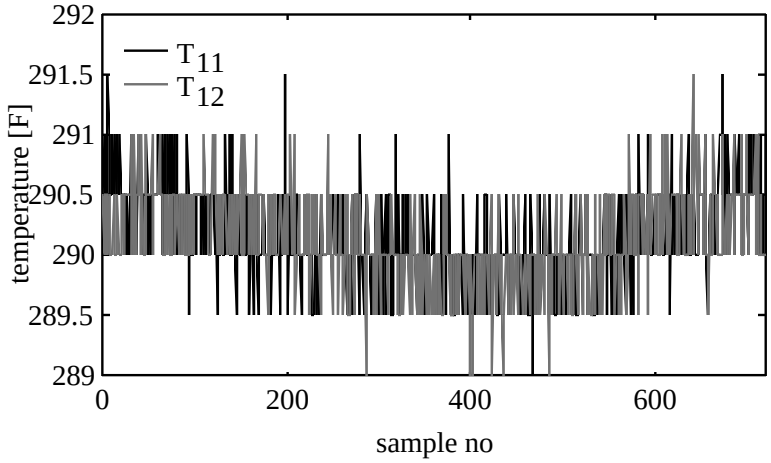


Fig. 14: T_{11} and T_{12}

Fig. 15 shows the load. It swings with a frequency of 1 cycle/day but has a superimposed smaller frequency of about 18 cycles/hour which can be modeled as a sinusoidal bias. It is interesting to note that all sensors measuring physical quantities after the turbine outlet exhibit the higher order swings. This may have to do with control action of the controller used, resonance in the manifold [6,7] or other phenomena.

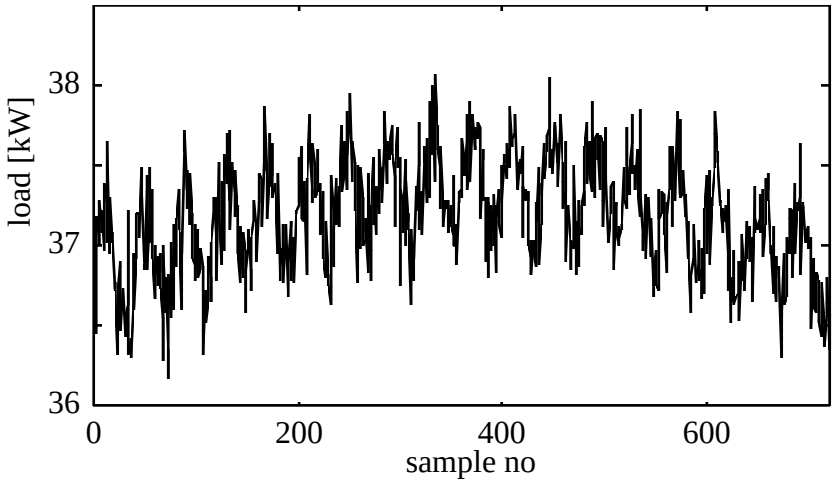


Fig. 15: Load generated

Fig. 16 shows the compressor inlet duct manifold pressure from which the compressor inlet pressure is calculated. The compressor inlet duct manifold pressure is an almost constant value over the period observed. However, outliers of relatively small magnitude can be seen occurring in a random fashion over the time interval of interest.

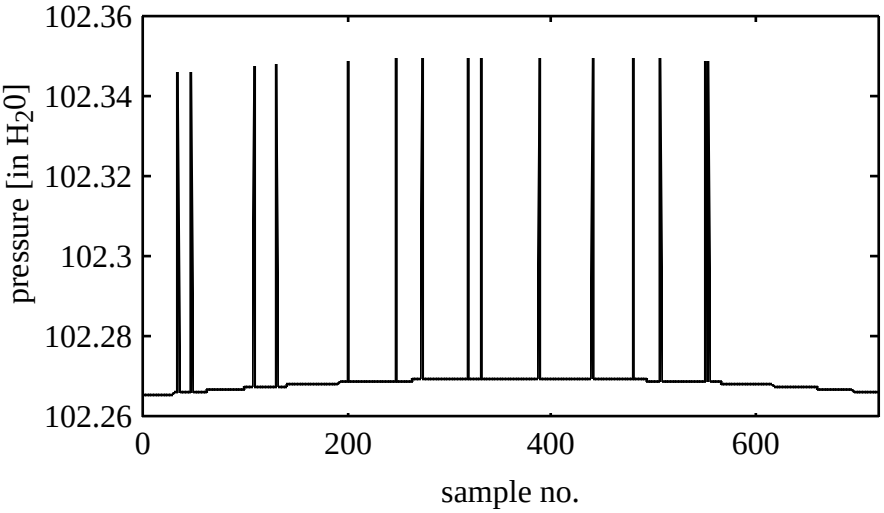


Fig. 16: Compressor inlet duct manifold pressure p_i

3.1 Validation and Fusion

It is shown in the following section how sensor validation and fusion can be used to achieve better measurements and hence better performance of the controller. Furthermore, it is shown that the algorithm is robust against sensor failure. The cost of a failed sensor could range from degradation of performance to an unnecessary shutdown of the plant. The efficiencies of the compressor and the turbine were assumed to be η_c=0.86 and η_t=0.9275, respectively. The validation and fusion algorithm fuses the two values from the two inlet flange temperature sensors T_{11}, T_{12} to get one inlet flange temperature value. It then does the same with the two compressor discharge temperature sensors to get one fused compressor discharge sensor. Then the compressor discharge pressure is calculated as

$$p_2 = p_1\left(\frac{\eta_c\left(T_{2_f} - T_{1_f}\right) + T_{1_f}}{T_{1_f}}\right)^{\frac{\kappa}{\kappa-1}} \tag{7}$$

where

T_{1_f} is the fused inlet flange temperature

T_{2_f} is the fused compressor discharge temperature

To reduce the lag and smooth the rugged readings and, while retaining robust functionality, the scheme displayed in Fig. 17 was used. It consists of smoothing all sensor readings through the use of the fuzzy time series predictor, followed by fusing the values for temperature sensors T_{11} and T_{12} to form T_{1f} as well by fusing T_{21} and T_{22} to form T_{2f}. The value of p_{22} is calculated as in the previous example and p_{23} is calculated as

$$p_2 = p_1\left(\frac{T_{2i}}{T_{1_f}}\right)^{\frac{\kappa}{\kappa-1}} \tag{8}$$

where

$$T_{2i} = T_1\left(\frac{T_3}{T_{4i}}\right) \tag{9}$$

$$T_3 = T_4 + \frac{W}{\dot{m}c_p} \tag{10}$$

$$T_{4i} = T_3 - \left(\frac{T_3 - T_4}{\eta_t}\right) \tag{11}$$

Finally, the three values for p_2 are fused to give p_{2f}.

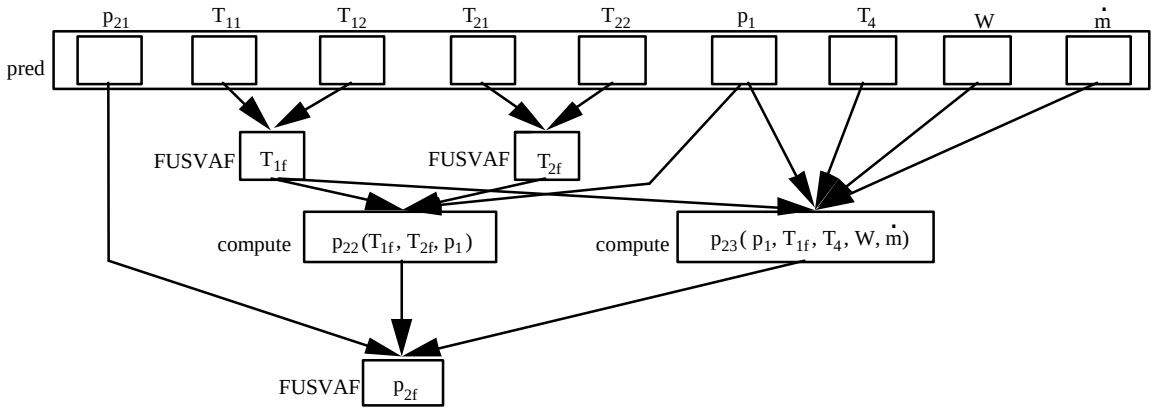


Fig. 17: Scheme for validation and fusion

Fig. 18 shows the output of the scheme used under normal operating conditions. Apart from the measured value of p_2, the computed value for p_{21} and p_{22} are also displayed along with the fused value p_{2f} and the smoothed fused value p_{2fm}. The latter has the most desirable features. It is smooth yet responsive to the change of the readings.

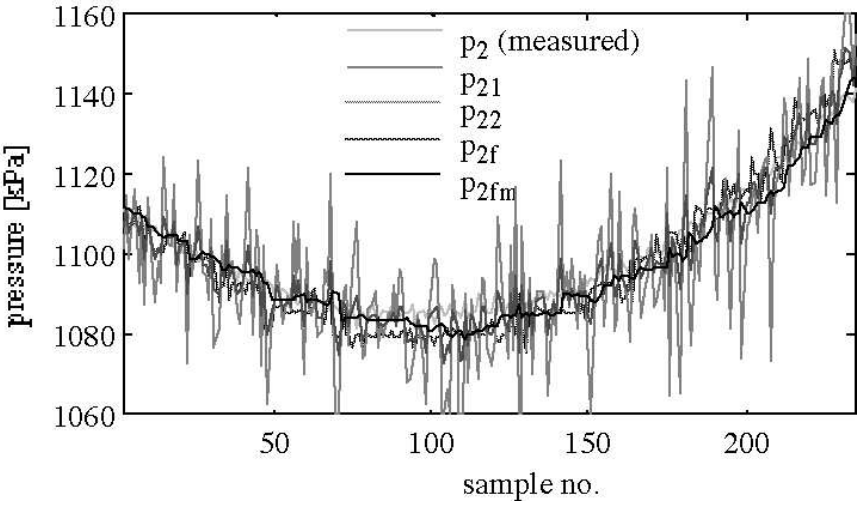


Fig. 18: Validation and multi-fusion under normal operation

In a next step, Monte Carlo simulations were used to investigate how the system behaves under sensor failure. For illustrative purposes we show in Fig. 19 a summary of the observations with multiple forced sensor failures. Failures were superimposed every 25 samples as summarized in TABLE 3. T_{11} fails from samples 51 to 200, T_{21} fails from samples 76 to 225, p_2 fails from samples 101 to 175, and T_4 fails between samples 126 and 150. The mass flow sensor fails as well in the range from 76 to 150 samples. The fusion algorithm disregards the calculation based on the faulty readings and uses the calculations based on T_{12}, T_{22}, and p_1 only. The smoothed value for the estimation of p_2 still provides a good input to the power plant controller.

TABLE 3

Summary of failures imposed on power plant data for five simultaneous sensor failures

Sensor failed	Range of failure
T_{11}	51-200
T_{21}	76-225
p_2	101-175
T_4	126-150
$\dot{m}$	76-150

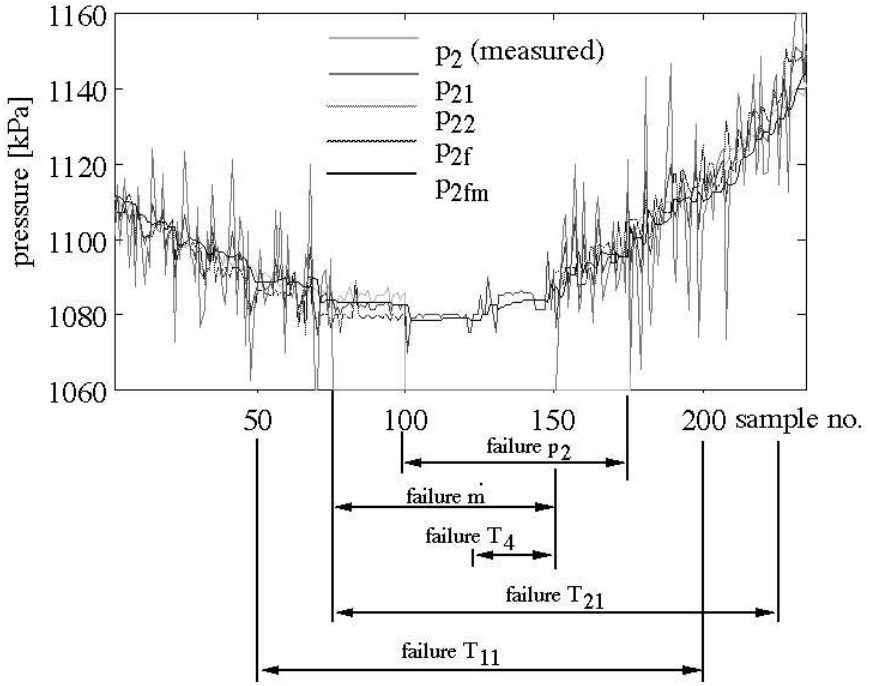


Fig. 19: Output of validation and fusion scheme subject to failure of five sensors

There are, of course, fault scenarios when too many sensors fail resulting in a lack of information for the algorithm to come up with any meaningful value for p_2. If, in addition to the five failures shown in the previous example, the inlet duct manifold pressure sensor fails as well in the range

from samples 126 to 225 there is not enough sensor information available from any sensor combination to calculate a good value for p_2. Rather, the algorithm keeps the last recorded value as the best estimate for the system state. The algorithm also issues a warning alerting the operator to a possible system failure which can trigger appropriate action. When some sensors come back on-line, they are again used to update the estimate for p_2. This validation and fusion scheme has the desirable feature of filtering out noise and bad sensor readings and provides a robust platform for the power plant controller.

There are other sensor failure combinations than the ones shown which also still allow proper determination of p_2, some involving more than five simultaneous failed sensors. However, there are also combinations of sensor failures which occur when fewer sensors fail. In particular, p_1 is used in both calculating p_{22} and p_{23}. Therefore, if p_2 and p_i or p_a fail at the same time over longer periods, the algorithm will not be able to give proper results. This highlights the importance of the use of independent sensors. From the suite of sensors at hand in this example, however, this bottleneck cannot be circumvented.

4 Summary and Conclusions

The use of the FUSVAF algorithm is a powerful means to provide smooth and robust sensor input to the gas turbine power plant. The fusion algorithm combines data obtained through direct and functional redundancy in algorithms using double and triple redundancy. A higher degree of redundancy leads to higher robustness against sensor failure, e.g., the algorithm used with three redundant values for p_2 is more robust than when only two redundant readings are used, which in turn is more robust than the controller relying on the direct measurement of p_2 alone. All three values for the control input discharge pressure p_2 are obtained through largely independent readings. Thus, if sensors fail on a particular subsystem, the readings from the other subsystems are able to maintain proper control input and unnecessary shutdowns can be avoided. In essence, the more sensor readings are integrated into the fusion algorithm, the smoother, reactive, and robust is the control input. Noise is filtered out before the individual sensor readings are used to compute the discharge pressure using the FEWMA predictor. In experiments with simulated sensor failure data on operating power plant data, we show how the FUSVAF and FEWMA algorithms improve the fused control input p_2 by filtering noise. We also show how superimposed sensor failure is rejected for multiple simultaneous sensor failures.

Our sensitivity analysis has shown the importance of selecting the initial parameters in the FUSVAF algorithm. A new application will require domain-specific fine tuning of these parameters, taking into account the operating conditions and sensor history. This can be accomplished through machine learning techniques using real data [16]. It is also important to establish sensor characteristics that allow proper modeling of the validation curves and also provide the means for diagnosis. Since a higher degree of redundancy results in more robust operation, it would be worthwhile to investigate the use of other sensors in the validation and fusion algorithms for the gas turbine application, especially if known functional relationships can be exploited [12]. It is crucial, however, to maintain maximum independence of the values computed – i.e., no two calculated values should rely on the same sensors – to avoid failure of several calculated values when one sensor malfunctions.

The FUSVAF has been successfully applied to other systems such as intelligent motor vehicles [10] on instrumented highways and monitoring of tool wear in milling machines [17]. It shows more robust performance than Bayesian sensor fusion systems when dealing with unknown systems, including non-Gaussian, sensor noise [18]. As an extension, the FUSVAF is not restricted to fusion on the data level. It can also be successfully employed at the feature level or decision level [17].

Acknowledgements
This research was in part supported by PATH (Partners in Advanced Transit Highways) and Caltrans through grants MOU-132, MOU-157, and MOU-231 as well as project MICRO (contract #93-004).

References

1. J.H.P. Millar 1993 *Nuclear Energy*, **32** (1), 57-63. Total Process Surveillance (TOPS).
2. A.M. Agogino, S. Srinivas, and K. Schneider 1988 *Mechanical Systems and Signal Processing* **2** (2), 165-185. Multiple Sensor Expert System for Diagnostic Reasoning, Monitoring, and Control of Mechanical Systems.
3. C. Jiaa and D. Dornfeld 1992 *Mechanical Systems and Signal Processing* **6**, (2), 97-120. Detection of Tool Wear Using Gradient Adaptive Lattice and Pattern Recognition Analysis.
4. X. Wu, J. Chen, W. Wang, Y. Zhou 2001 *Mechanical Systems and Signal Processing* **15** (5), 995-1006. Multi-Index Fusion-Based Fault Diagnosis Theories and Methods.
5. G.P. Singh, D.A. Steinke, J.L. Tsou 1993 *Proceedings of the American Power Conference* , **1836** (2), 1504-7. Heat Exchanger Workstation: A Comprehensive Software Tool for Heat Exchanger Operation and Maintenance.
6. Sir H.R. Cox, 1953 *Gas Turbine Principles and Practice*, London: George Newnes Limited.
7. H. Constant, 1953 *Gas Turbines and Their Problems*, London: Todd Publishing Group Limited.
8. C.V. PhaniShankar, S. Orth, J. Frolik, and M. Abdelrahman 2000 *Proceedings of 2000 American Control Conference*, 2912-2916. Fuzzy Rules for Automated Sensor Self-Validation and Confidence Measure.
9. P. Prajitno and N. Mort 2001 *Proceedings of the SPIE, Sensor Fusion: Architectures, Algorithms, and Applications V*, **4315**, 301-312. A Fuzzy Model-Based Multi-Sensor Data Fusion System.
10. K. Goebel and A. Agogino 2001 *Journal of Dynamic Systems, Measurement, and Control* , **123**, 145-146. Sensor Validation and Fusion for Automated Vehicle Control Using Fuzzy Techniques.
11. K. Goebel, A.M. Agogino 2000 *International Journal of Uncertainty, Fuzziness, and Knowledge Systems* **8** (4), 453-469. Fuzzy Belief Nets.
12. Y.J. Kim, W.H. Wood, and A.M. Agogino, 1992 *Proceedings of the 2nd International Forum on Expert Systems and Computer Simulations in Energy Engineering* **9-5**, 1-6. Sensor Systems for an On-Line Diagnostic Expert System in Fossil Power Plants.
13. J. Wang, S.Y. Chao, A.M. Agogino 1999 *Proceedings of the Advanced Controls Conference* **FM11**, 3760-3764. Sensor Noise Model Development of a Longitudinal Positioning System for Advanced Vehicle Control Systems.
14. P. Khedkar and S. Keshav 1992 *Proceedings of the IEEE International Conference on Fuzzy Systems*, **1438**, 281-288. Fuzzy Prediction of Time Series.
15. D.E. Goldberg 1989 *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading: Addison Wesley.
16. K.Goebel 1996 Management of Uncertainty for Sensor Validation, Sensor Fusion, and Diagnosis Using Soft Computing Techniques Ph.D. Thesis, University of California at Berkeley.
17. K. Goebel, V. Badami and A. Perera 1999 *Proceedings of the Second International Conference on Information Fusion*, **1**, 331-336. Diagnostic Information Fusion for Manufacturing Processes.
18. K. Goebel, S. Alag, A.M. Agogino 1997 *Proceedings of ISATA '97, 30th International Symposium on Automotive Technology & Automation* **1**, 711-719. Probabilistic and Fuzzy Methods for Sensor Validation and Fusion in Vehicle Guidance: A Comparison.