INAE Lett (2017) 2:133–143
DOI 10.1007/s41403-017-0023-y

TECHNICAL NOTE

Denoising Signals Used in Gas Turbine Diagnostics with Ant Colony Optimized Weighted Recursive Median Filters

Chintan Raikar[1] · **Ranjan Ganguli**[2]

Received: 5 December 2016 / Accepted: 15 June 2017 / Published online: 29 June 2017

Abstract Accurate fault detection and isolation requires signal processing of measurement signals which are contaminated with noise. Typically, gas turbine faults are revealed by sharp trend shifts in the signals and these trend shifts should be preserved during signal processing. Linear filters can smooth out the sharp trend shifts while removing noise. However, nonlinear filters such as the weighted recursive median (WRM) filters show good noise reduction while preserving key signal features if their integer weights are determined optimally. We propose the ant colony optimization (ACO) method coupled with local search to calculate the integer weights of WRM filters. It is found that the filter weight optimization problem is mathematically equivalent to the quadratic assignment problem which can be solved by ACO. Optimal parameters for the ACO are found using numerical experiments. The WRM filter is demonstrated for abrupt and gradual faults in gas turbines and is found to yield noise reduction of 52–64% for simulated noisy signals considered in this paper.

Keywords Gas turbine diagnostics · Condition monitoring · Ant colony optimization · Signal processing · Nonlinear filters · Fault detection and isolation

✉ Ranjan Ganguli
ganguli@aero.iisc.ernet.in

[1] Department of Aerospace Engineering, IIT Mumbai, Mumbai 400076, India

[2] Department of Aerospace Engineering, IISc, Bangalore 560012, India

Introduction

Gas turbine diagnostics typically uses measured data in conjunction with identification, optimization or soft computing algorithms to detect and isolate engine module and sensor faults (Volponi et al. 2003; Lu et al. 2001; Sampath and Singh 2006). There are two broad types of engine faults: single faults and gradual faults. Single faults are typically preceded by a sharp change in the signal. Gradual faults lead to a slow change in the signal which can be approximated with a linear variation. Typical signals used for gas turbine diagnostics are exhaust gas temperature (EGT), fuel flow (WF), high rotor speed (N1) and low rotor speed (N2). These four basic sensors are present in almost all jet engines. The signals considered for gas turbine diagnostics are called "measurement deltas" which are deviations between sensor measurements of a "damaged" engine compared to a "good" engine. For an ideal undamaged engine, the measurement deltas are zero. The measurement deltas obtained from operational engines are typically non-zero and are also contaminated with noise. Fault detection and isolation (FDI) algorithms are used to detect and isolate the engine fault. Here "detection" is the process of identifying if a fault is present or not. Errors in detection can lead to false alarms. Also, "isolation" is the process of identifying the type of fault. Typically, fingerprint charts are used for fault isolation and these relate the measurement deltas produced to a given change in the engine state. For example, Table 1 presents the fingerprint chart for a 2% deterioration in the efficiency of the engine modules (Uday and Ganguli 2010). Fingerprint charts represent a linearized model evaluated at a selected engine operating point. Such tables are obtained from thermodynamics and are available with engine manufacturers.

Table 1 Fingerprints for selected gas turbine faults for $\eta = -2\%$

Module faults/measurement deltas	ΔEGT (C)	$\Delta WF\%$	$\Delta N2\%$	$\Delta N1\%$
High pressure compressor (HPC)	13.60	1.60	−0.11	0.10
High pressure turbine (HPT)	21.77	2.58	−1.13	0.15
Low pressure compressor (LPC)	9.09	1.32	0.57	0.28
Low pressure turbine (LPT)	2.38	−1.92	1.27	−1.96
Fan	−7.72	−1.40	−0.59	1.35

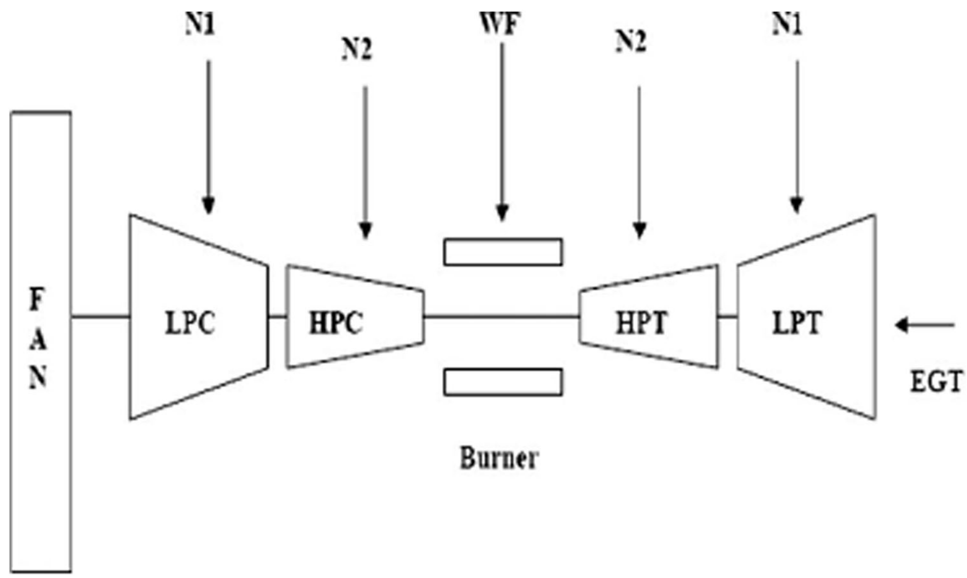

Fig. 1 Schematic representation of jet engine and its four key measurements

Figure 1 shows the engine modules and basic measurements for a typical turbofan engine. At the most basic level, fault isolation would indicate if the fault is present in the fan, high pressure compressor, low pressure compressor, high pressure turbine or low pressure turbine module. These are coupled faults within the major modules of the engine. Other system faults such as handling and ECS bleed leaks and failures, variable stator vane malfunctions, TCC malfunctions as well as certain instrumentation faults can also be considered as single faults (Volponi et al. 2003). We will only consider module faults to create the ideal signal for damaged engine in this study. Such faults can emanate from different physical processes but the signature is shown through the sensor measurement deltas. Once the fault is isolated to the module level, the maintenance engineer can then focus only on these modules for repair work.

The accuracy of FDI algorithms improve if noise is removed from the gas path measurement signals while preserving features indicating a single fault such as sharp trend shifts (Ganguli 2002). Depold and Gass showed that typical linear filters such as the moving average filter and exponential average filter can work as good smoothers for gas turbine signals (DePold and Gass 1999). The moving average filter is a simple finite impulse response (FIR) filter with equal weights and the exponential average is an infinite impulse response (IIR) filter. While linear filters can remove noise, they smooth out the sharp trend shifts which can indicate a single fault event. Therefore, nonlinear filters such as the median filter have been proposed for noise removal from gas turbine signals (Ganguli 2002). Other computational architectures for noise removal from jet engine signals include the auto-associative neural network (Lu and Hsu 2002), radial basis neural networks (Verma et al. 2006), myriad filter (Surender and Ganguli 2004) and recursive median filter (RM) (Ganguli and Dan 2004). The RM filter is an efficient alternative to the median filter and converges rapidly to the root signal when compared to the simple median (SM) filter. The SM filter can take many passes before converging to the root signal. However, the RM filter can lead to a phenomenon called "streaking" which involves creation of artificial step like artifacts in the signal. This problem can be removed by introducing weights resulting in the weighted recursive median (WRM) filter.

The WRM filters have integer weights and the optimal calculation of these weights for a given application is an important problem in filter design. The design space of the weights of WRM filters is multimodal (shows the presence of several local minima) and an exhaustive search of the

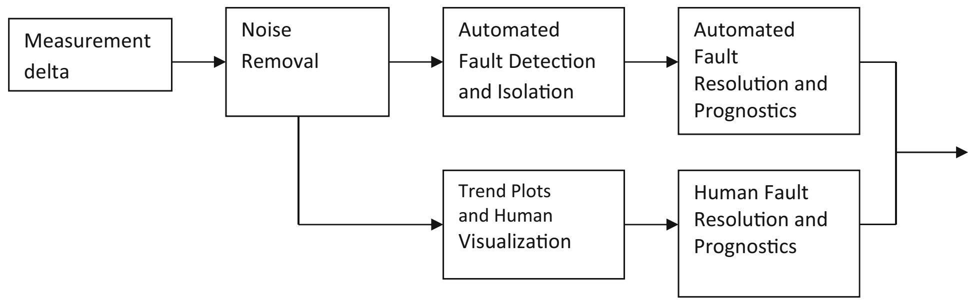

Fig. 2 Schematic representation of gas turbine diagnostics system

design space can be used to find the weights (Uday and Ganguli 2010). However, this exhaustive search method is very computationally intensive and there is a need for more efficient algorithms for solving this filter weight optimization problem. In this paper, ACO is used to design a WRM filter for use as a data smoothing preprocessor in gas turbine diagnostics. A schematic of this procedure is shown in Fig. 2. We focus on the "noise reduction" aspect of gas turbine diagnostics shown in Fig. 2.

Some researchers have addressed the problem of median filter weight optimization. Algorithms for calculating the integer weights of weighted median filters were proposed (Yang et al. 1995). Both recursive and non-recursive filters were considered but the study focused on center weights. A numerical approach for the optimization of recursive median filters was presented in Arce and Paredes (2000). Uday and Ganguli (2010) searched over the low integer space (1, 2 and 3) to find the optimal weights. They found that higher integer weights led to duplication in the filter and the low integer space was sufficient for the given problem.

Filter design spaces can often be multimodal which means that there can be more than one minimum point. Therefore, gradient based numerical optimization can settle into a local minimum point. To address this issue, the use of global optimization methods in filter design has grown substantially. Particle swarm optimization was used to solve the parameter estimation problem of nonlinear dynamic rational filters (Lin et al. 2008). Genetic algorithms have also been used for optimizing stack filters using a root mean square error approach (Zhao and Zhang 2005). Ant colony optimization was used for the design of IIR filters (Karaboga et al. 2004). Since the error surface of IIR filters is generally multimodal, global optimization methods such as ACO are well suited for their design. ACO is a relatively new approach for solving combinatorial optimization problems. The main characteristics of ACO are positive feedback, distributed computation, and the use of a constructive greedy heuristic (Dorigo et al. 1996). Note that a heuristic method is an approach to solving problems that employs a practical method which is not guaranteed to be optimal or perfect, but sufficient for the immediate goals. Heuristics are often rules of thumb or educated guesses. Since ACO is a heuristic method, is gives satisfactory solutions but these solutions may not prove to be optimal and the convergence of such methods cannot be guaranteed.

In this paper, we address the problem of finding the integer weights of WRM filters using ACO. The algorithm is demonstrated for signals simulating jet engine single (abrupt) and gradual faults.

Median Filters

The SM (simple median) filter with length of window of $N = 2n + 1$ can be represented as (Brownrigg 1984),

$$y_k = median(x_{k-n}, x_{k-n+1}, \ldots, x_k, \ldots, x_{k+n-1}, x_{k+n}). \quad (1)$$

Here x_k and y_k are the kth sample of the input and output sequences, respectively, and n represents integers ensuring that the window length N is odd for easy calculation of the median. The SM filter needs a large number of iterations to converge to a desired output. A five-point SM filter can be written as $y_k = median(x_{k-2}, x_{k-1}, x_k, x_{k+1}, x_{k+2})$ since $N = 5 \Rightarrow n = 2$. The five-point SM filter has a window length of five and a two point time lag as it needs measurements at the time points $k + 1$ and $k + 2$ to predict the output at k. Since most current jet engines have many data points available during the flight, a two point time lag for signal processing is acceptable.

A recursive median (RM) filter for window length $N = 2n + 1$ can be represented as,

$$y_k = median(y_{k-n}, y_{k-n+1}, \ldots, x_k, \ldots, x_{k+n-1}, x_{k+n}). \quad (2)$$

RM filters require very few steps for convergence as further application of the filter on a signal does not bring any changes. A five-point RM filter can be written as $y_k = median(y_{k-2}, y_{k-1}, x_k, x_{k+1}, x_{k+2})$ where the use of previously filtered output values y_{k-1} and y_{k-2} point to the *recursive* nature of this filter. Again, this filter has a two point time delay.

The WRM filter is a modified version of RM filter, where *integer* weights are assigned to each data point in the filter window. The output of a weighted recursive median filter with window length $N = 2n + 1$ is given by,

$$y_k = median(w_{-n} \circ y_{k-n}, w_{-n+1} \circ y_{k-n+1}, \ldots, w_0 \circ x_k, \ldots, w_{n-1} \circ x_{n-1}, w_n \circ x_{k+n}). \quad (3)$$

Here $^\circ$ stands for duplication and w are the integer weights. Duplication means that a particular sample x_k is repeated w_k times before taking the median of the array. For example, $(4 \circ x_1)$ is the same as $(x_1\, x_1\, x_1\, x_1)$, i.e. the value x_1 is duplicated four times. As, an example, consider a five-point WRM filter $y_k = median(2 \circ y_{k-2}, y_{k-1}, 3 \circ x_k, x_{k+1}, 2 \circ x_{k+2})$ which is actually identical to $y_k = median(y_{k-2}, y_{k-2}, y_{k-1}, x_k, x_k, x_k, x_{k+1}, x_{k+2}, x_{k+2})$. Again, this filter will have a two point time delay. The filter weights are indicated by the set $(w_{-n}, w_{-n+1}, \ldots, w_0, \ldots, w_{n-1}, w_n)$ where there are $N = 2n + 1$ weights. For a five-point filter the weight set is $(w_{-2}, w_{-1}, w_0, w_1, w_2)$.

Gas Path Measurement Signals

A turbofan jet engine in general consists of five modules: fan (FAN), low pressure compressor (LPC), high pressure compressor (HPC), high pressure turbine (HPT) and low pressure turbine (LPT), as shown in Fig. 1. Air coming into the engine is compressed in the FAN, LPC, and HPC modules; combusted in the burner; and then expanded through the HPT and LPT modules to produce power. The sensors N1, N2, WF, and EGT represent low rotor speed, high rotor speed, fuel flow and exhaust gas temperature, respectively. These signals provide information about the condition of these modules and are used for engine condition monitoring. In this study, an ideal root signal ΔEGT with implanted HPC and/or HPT faults is used for testing the filters. Similarly, root signals for ΔN1, ΔN2, ΔWF can be derived. Here the delta "Δ" refers to the deviation from the baseline "good" engine.

Test signals for faults in jet engines are used to demonstrate the algorithm (Uday and Ganguli 2010). For a new undamaged engine, the measurement delta is zero. For a typical engine which goes into service, the measurement deltas slowly increase with time due to *deterioration* as the number of flights increase. While deterioration increases gradually as flight hours and cycles accumulate, *single faults* lead to sudden, abrupt or step changes in the signal. For this study, a step change in measurement deltas of 2% or more is regarded as a large enough trend to be interpreted as a single fault event (Volponi et al. 2003). Gaussian noise is added to the simulated measurement deltas using the typical standard deviations for ΔEGT (deviation in exhaust gas temperature from a good baseline engine) as 4.23 °C. These values are obtained by a study of typical engine measurement deltas (Volponi et al. 2003). Measurement deltas are created using: $z = z^0 + \theta$ where θ is the noise and z^0 is the baseline measurement delta without any fault (ideal signal). Thus, z is the simulated noisy signal. Therefore, a filter φ is required to eliminate noise from data and return a filtered signal $\hat{z}$ for accurate condition monitoring:$\hat{z} = \varphi(z) = \varphi(z^0 + \theta)$.

We consider three different types of signals for designing the WRM filter using ACO. Though these signals are outlined for gas turbine diagnostics, they are applicable for any general FDI problem as the abrupt fault and long term deterioration are characteristics of all signals used for condition monitoring.

1. Step signal (abrupt fault or single fault)
2. Ramp signal (gradual fault)
3. Combination signal (comprising of both abrupt and gradual fault)

Each of the signals contains 200 data points which represents a time series of engine data available for signal processing. The data comes in at each epoch k (Fig. 3) and the filtered value is calculated using the N-point WRM filter. The filter of window length N processes the data as it comes in with a time lag of n (Eq. 3). We use a five-point filter in this paper. So the WRM filter works on a stream of 200 data points $(x_1, x_2, x_3, \ldots, x_{198}, x_{199}, x_{200})$ to yield filter output $(y_1, y_2, y_3, \ldots, y_{198}, y_{199}, y_{200})$ as given in Eq. 4

$$\begin{aligned}
&k = 1,\ y_1 = x_1\\
&k = 2,\ y_2 = x_2\\
&k = 3,\ y_3 = median(w_{-2} \circ y_1, w_{-1} \circ y_2,\\
&\quad w_0 \circ x_3, w_1 \circ x_4, w_2 \circ x_5)\\
&k = 4, y_4 = median(w_{-2} \circ y_2, w_{-1} \circ y_3,\\
&\quad w_0 \circ x_4, w_1 \circ x_5, w_2 \circ x_6)\\
&\vdots\\
&k = 100, y_{100} = median(w_{-2} \circ y_{98}, w_{-1} \circ y_{99},\\
&\quad w_0 \circ x_{100}, w_1 \circ x_{101}, w_2 \circ x_{102})\\
&\vdots\\
&k = 198, y_{198} = median(w_{-2} \circ y_{196}, w_{-1} \circ y_{197},\\
&\quad w_0 \circ x_{198}, w_1 \circ x_{199}, w_2 \circ x_{200})\\
&k = 199, y_{199} = x_{199}\\
&k = 200, y_{200} = x_{200}.
\end{aligned} \quad (4)$$

We see that at $k = 3$, to get y_3 requires x_4 and x_5. So the filter has a two point time delay. Also, for the last two points in the time series we use the input value of the data. However, in normal operation, the data points would continue to stream in as the aircraft continues to accrue flights. The data point to be processed for fault detection is thus available with a two-point time delay. This data point can be used by trend detection algorithms which are

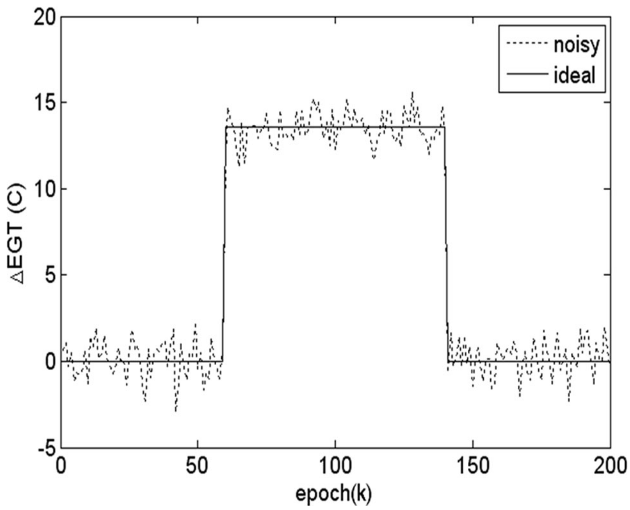

Fig. 3 Step signal representing a single fault and its repair

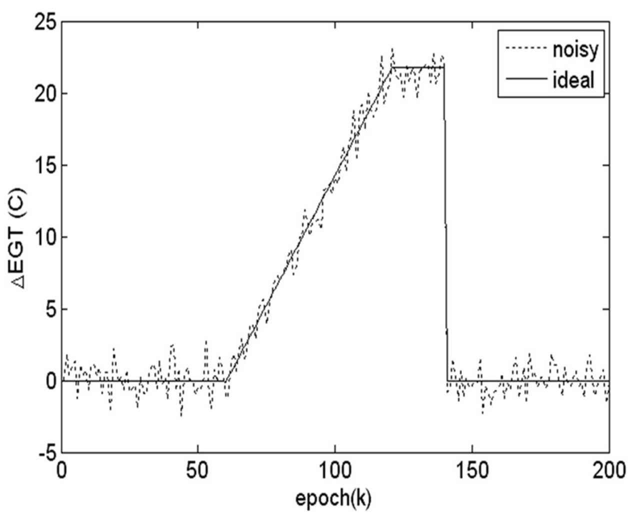

Fig. 4 Ramp signal representing gradual fault followed by its repair

typically based on derivatives (Ganguli and Dan 2004). So fault detection occurs with only a two-point time delay for the five-point filter.

The ideal signal in Fig. 3 represents a single fault that may be due to any damage. Data point $k = 60$ represents the onset of this fault. The damage caused is identified as a 2% fall in HPC efficiency and the HPC module is repaired at point $k = 140$. This signal is created based on the fingerprint chart given in Table 1. In Fig. 4, the development of the HPT fault is illustrated by use of the ramp signal. This fault differs from the HPC fault in that it does not occur suddenly as it develops due to engine deterioration. Again, the maximum value of EGT here corresponds to a 2% fall in HPT efficiency. Here, the growth is gradual and is approximated by a linear function from points $k = 40$–120. From $k = 120$ the HPT fault remains steady

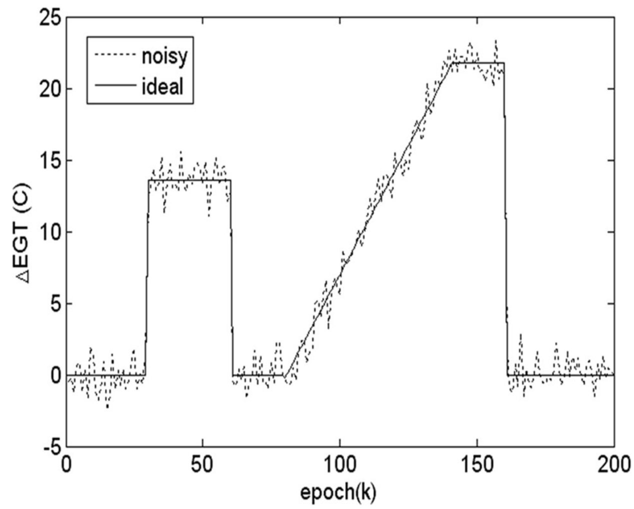

Fig. 5 Combination signal representing single fault and its repair followed by gradual fault and its repair

and is finally repaired at $k = 140$. The step and ramp signals represent the two types of faults considered individually. Now, Fig. 5 shows a combination signal, wherein, both types of faults may occur one after the other. This is a more practical case since any jet engine is susceptible to both these faults.

A signal to noise ratio of 1.5 is used for the numerical results. A five-point WRM filter is considered and this filter processes the 200 point measurement delta signal with a window of five points and a time delay of two point.

Objective Function

To get a quantitative idea of noise reduction, the mean absolute error (MAE) is considered for each signal ($\bar{N} = 200$) and $M = 1000$ random realizations are used to get an estimate of the error. These random realizations can be considered to be simulated signals of noisy data. These random signals are $z = z^0 + \theta$ and are obtained by adding a different noisy sample to the ideal measurement for each case. These noisy signals are generated via Matlab which is also used for all the results in this paper. We illustrate the ACO algorithm for a filter of window length equal to five. For finding the optimum weights of this filter, we have to minimize the objective function:

$$f(w) = f(w_{-2}, w_{-1}, w_0, w_1, w_2) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{\bar{N}}\sum_{j=1}^{\bar{N}}\left|\hat{z}_j - z_j^0\right|. \quad (5)$$

Since the weights $w = (w_{-2}, w_{-1}, w_0, w_1, w_2)$ are integer design variables, the problem is a combinatorial optimization problem. Also, in Eq. (5) $\hat{z}$ is the WRM filtered signal and z^0 is the ideal or root signal. By minimizing the function in Eq. (5), we want to identify the weights which minimize the difference between the ideal signal and the noisy signal over a large number of noisy points. Note that the objective function with the 1000 random samples is only created to find the optimal filter weights. Once the filter weights are found, the filter can be tested and used on new random samples.

Ant Colony Optimization

Ant colony optimization (ACO) is a biologically inspired stochastic method which is especially suited for combinatorial optimization problems (Dorigo and Stutzle 2005). Such problems have a finite number of solutions with discrete design variables. Ants are able to find the shortest paths needed to go from their nest to a food source by using *stigmergy*, which is a form of indirect communication conducted by modifications of the environment. Ants use sign based *stigmergy* where an individual ant leaves markers or messages. While these markers do not solve the problems themselves, they influence the behavior of other ants in a way that helps them in problem solution.

The inspiration for the ACO algorithm came from some experiments performed on Argentine ants which revealed the science behind their optimal path finding capabilities. An innovative experiment was performed using a double bridge between an ant nest and a food source. Here each bridge is of the same length as shown in Fig. 6. If the bridges are of the same length, after some time, the ants tend to take one of the bridges to the food source. If the experiment is repeated several times, it is found that the probability of selection of any one bridge is about 0.5. The biological explanation of the ant behavior is as follows. Once the ants leave the nest, they will move randomly until some find the bridge. Some ants will randomly start on bridge A while others will randomly start on bridge B. Now, ants deposit a chemical called *pheromone* when they walk along a path. They prefer to follow a path with higher pheromone deposits. Since, there is no pheromone initially on either bridge A or B, the probability of ants taking either bridge is equal at 0.5. Once the ants discover the food source, they will take some food and return back to the nest. This process will lead to ants traveling on both bridges until through random chance; a few more ants take, say, bridge A. After this point, the *pheromone trail* on bridge A will strengthen making it more attractive for the ants. Another important point to appreciate is that pheromone keeps evaporating and so the pheromone trail on bridge B will weaken. After some time, almost all ants will take bridge A.

Of course, taking only one bridge to the food source when two bridges of equal length are available is not a smart thing to do. However, ants are prisoners of their swarm intelligence which becomes very useful in the situation where two paths of different lengths are present, as shown for the two bridges in Fig. 7. In this case, the ants again set out initially in a random manner and take both the bridges with equal probability assuming that the lengths are too large for their field of vision. The ants which choose the shorter bridge B reach the food source first. When the ant wants to return to its nest, it comes to point 2 and finds that bridge B has a higher level of pheromone due to less evaporation. The probability of choosing bridge B therefore becomes higher. As this process continues, the positive feedback effect means that more pheromone is put on bridge B and less evaporation of pheromone takes place on bridge B. A positive feedback loop is thus created and after some time, most ants will take bridge B to the food source. We can see that ants are capable of a high degree of *self-organization* using the stigmergy principle. By modifying the environment via pheromones, they can collectively perform complex functions despite their poor vision. In fact, some species of ants are completely blind but are still able to find the shortest path.

We can see that the behavior of ants could be used for optimization algorithms which involve finding good paths through graphs Boryczka and Kozak (2015), Dorigo et al. (1996). In ACO, several generations of artificial ants search for good solutions. We use the word "ant" to refer to the "artificial ant" in the ACO algorithm in further discussions. Every ant of a generation builds up a solution in a step by step manner while going through several probabilistic decisions. In general,

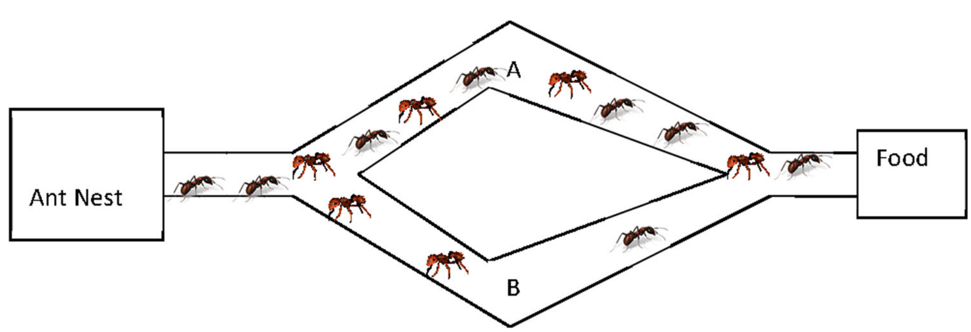

Fig. 6 Two-bridge experiment with ants for bridges of equal length

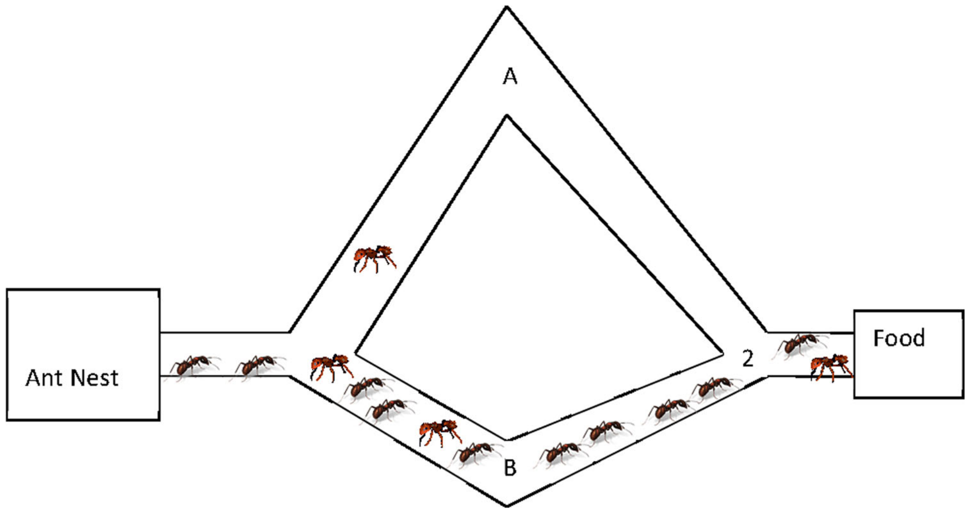

Fig. 7 Two bridge experiment with ants for bridges of unequal length

ants that find a good solution mark their paths through the decision space by putting some amount of pheromone on the edges of the path. The ants of the next generation are attracted by the pheromone trail left behind by the previous ants so they search the solution space near good solutions. In addition to the pheromone values, the ants are also guided by some problem specific heuristic (a rule of thumb specific to the problem being solved). The ACO combines a priori information about a promising solution with a posteriori information about previously obtained good solutions.

ACO has been used to solve combinatorial optimization problems (Gambardella et al. 1999; Dorigo and Stutzle 2005). As assignment problem is a combinatorial optimization problem were a set of items or objects is assigned to a set of resources or locations. Such assignments can be represented as a mapping from a set I to a set J. The objective function to be minimized is a function of the assignment done. Consider the pair (i, j) where i is an item and j is a resource. The pheromone trail τ_{ij} is the desirability of assigning i to j.

We can see that the WRM filter optimization is an assignment problem, wherein integer weights are assigned to a particular data point of a WRM filter with the objective of minimizing the mean absolute error over M samples. We seek to find the weight vector w which minimizes Eq. 5. Consider the five-point WRM filter with weights $w = (w_{-2}, w_{-1}, w_0, w_1, w_2)$. We want to assign weights from set of integers (1, 2, 3, 4) to minimize the error in Eq. 5.

Ant Colony Algorithm

This section describes the various components of ACO.

Set of initial solutions this set is made such that solutions are not repeated and no two solutions in this set can be converted into each other by swapping of elements.

Pheromone trail matrix an important feature of ACO is the pheromone trail management. Along with the objective function, pheromone trail values are used to construct new solutions from the existing ones. Pheromone trail values measure the desirability of having an element in the solution. These are maintained in a pheromone matrix T with elements τ_{ij}.

Filter Weight Optimization

This section discusses about the application of ACO to WRM filter optimization problem.

Initialization of solution initially, every ant k is allotted a randomly chosen solution w^k such that no two ants have the same initial solution. A total of m ants are used. Each initial ant solution is optimized using a local search procedure and the best solution is labelled as w^*.

Pheromone trail initialization pheromone matrix component τ_{ij} measures the desirability of assigning weight w_i to a jth data point in the N point filter. In the weight assignment problem, T matrix size is $N \times Max$ where N is the number of points in the WRM filter and Max is the maximum positive integral value that a weight can be assigned. Here $Max = 4$ and $N = 5$ are considered. The pheromone matrix T is formed by setting all the pheromone trails τ_{ij} to the same initial value τ_0. The pheromone trails determine the quality of the solution obtained, hence τ_0 must take a value that depends on value of the solution. Hence we chose to set $\tau_0 = Q/f(w^*)$ where w^* is the best initial solution and Q is a constant to be found from numerical experiments.

Solution construction pheromone trail based modification is applied by every ant to its own solution w^k. It consists of the following steps. Any arbitrary filter data point r of the N-point filter is chosen and then a second point s is chosen such that $s \neq r$ and the weights w_r and w_s are swapped in the current solution w. The second index s is chosen in such a way that the value of $\tau_{rw_s} + \tau_{sw_r}$ is maximum. Thus, by exploiting the pheromone trail, a new solution $\hat{w}^k$ is obtained for each ant which gives the most desirable path with the highest pheromone value.

Local search modification local search involves exploring of neighborhood of current solution $\hat{w}^k$. It involves changing weights w_r while keeping the other weights constant to produce $\tilde{w}^k$. The improvement is recorded as $\Delta(\hat{w}^k, r, s)$ which is the difference in objective function $f(w)$ when weight w_r is changed with s, where s can be any integral weight excluding w_r. This procedure is repeated for all the data points of the filter. Using the objective function as a measure, we find the optimum solution $\tilde{w}^k$. If no improvement is found, then no change is made to the earlier solution $\hat{w}^k$ obtained by the ants.

Pheromone trail modification there are several different pheromone update formulas. For the current problem, we use the ant colony system (ACS) pheromone update formula. Each ant applies this update as follows:

$$\tau_{ij} = (1 - \alpha)\tau_{ij} + \alpha\tau_0. \quad (6)$$

Here, α is a parameter that controls pheromone evaporation and is called the pheromone evaporation rate $(0 < \alpha < 1)$.

Terminating condition the termination condition is reached when a predefined number of ant generations (*niter*) have completed their search in the solution space.

The different steps of algorithm are given as pseudocode in the "Appendix". The parameters Q, *niter*, m and α are found from numerical experiments.

Results and Discussion

The ACO was tested for different parameter setting of Q, number of iterations *niter*, number of ant's m, and pheromone evaporation rate α for 100 noisy ramp input signals. The number of ants m was varied from 2 to 10 while number of iterations *niter* was varied from 1 to 10. The evaporation rate was varied from 0.1 to 0.9 and a rate of 0.4 was found to be good.

It was found that the number of ants was the key driving force for a good quality solution. The solutions were found to improve with increasing number of ants as seen from Table 2 for one case with three iterations. The optimum number of ants was found to be 8–10. Therefore, we use ten ants for finding out best value of number of iteration.

The optimum number of iterations was found to be three on the basis of solution quality and simulation time as seen from Table 3. The final parameter settings are selected to be: $\alpha = 0.4$, $Q = 0.1$, $niter = 3$ and $m = 10$. Numerical experimentation showed that the parameters obtained are not dependent on the type of noisy signal used. The ACO algorithm is finally applied on the three different types of noisy test signals.

Table 2 Change in objective function with number of ants ($niter = 3$)

No. of ants	MAE value	Best weight
1	0.3209	[1 4 3 1 3]
2	0.2899	[3 3 2 2 4]
3	0.3206	[3 1 3 4 1]
4	0.2854	[3 1 2 1 3]
5	0.2854	[4 1 2 1 4]
6	0.2854	[3 1 2 1 3]
7	0.3206	[3 1 3 4 1]
8	0.2817	[4 1 2 2 3]
9	0.2854	[4 1 2 1 4]
10	0.2771	[4 1 2 2 3]

Table 3 Change in objective function with number of iterations ($m = 10$)

Iteration	MAE value	Weight
2	0.2795	[3 1 2 1 3]
3	0.2771	[4 1 2 2 3]
4	0.2771	[4 1 2 2 3]
5	0.2795	[3 1 2 1 3]

Table 4 Optimal WRM filter weights

Signal type	w_{-2}	w_{-1}	w_0	w_1	w_2
Step	4	2	3	1	4
Ramp	4	1	2	2	3
Combination	3	1	2	1	3

The optimal filter weights obtained using ACO are shown in Table 4. The performance of WRM filter is compared with the performance of SM and RM filters in Table 5. These comparisons are for a different set of 1000 noisy data points compared to those used for finding the filter optimal weights.

To quantify the advantage of using the optimal WRM filter for noisy data, we define a noise reduction measure as follows,

$$\rho = \frac{MAE^{(noisy)} - MAE^{(filtered)}}{MAE^{(noisy)}} \times 100. \quad (7)$$

Table 5 clearly illustrates the improvement shown by the WRM filters. The WRM filter with weights given in Table 4

Table 5 Mean absolute error for the filters

Signal type	SM filter	RM filter	WRM filter
Step signal	0.3638	0.3031	0.2426
Ramp signal	0.3856	0.3739	0.2773
Combination signal	0.3999	0.3930	0.3027

Table 6 Performance comparison of median filters

Signal type	ρ^{SM}	ρ^{RM}	ρ^{WRM}
Step signal	45.87	54.90	63.90
Ramp signal	42.49	44.23	55.13
Combination signal	40.04	41.07	51.94

provides a noise reduction of about 52–64%. In contrast, the RM filter yields noise reduction of 41–55% and the SM filter shows a noise reduction of only 40–46%. Note that the RM filter could be considered as a WRM filter with unit weights. The improvement between SM and RM filter results is due to recursion. The improvement between the RM and WRM results are due to optimal weights obtained using ACO. The values of these parameters in Table 6 clearly justify the improved performance of WRM filters over SM and RM filters as illustrated by the simulated signals from gas turbine diagnostics.

The WRM filter obtained provides a noise reduction of about 52–64%. In particular, compared to the noisy signal, the noise reduction is 64, 55 and 52% for the step signal, ramp signal and the combination signal, respectively. It is clear that ACO represents an effective approach for the development of WRM filters.

We should mention that we have not considered sensor faults in this paper as other algorithms such as those based on Kalman filters can address such issues.

Conclusions

Removing noise from gas turbine measurement signals before subjecting them to fault detection and isolation algorithms is an important aspect of gas turbine diagnostics. Nonlinear filters such as the WRM filter are attractive for these problems as they do not smooth out the step changes in signals which typically indicate the onset of a single fault. However, filters such as the WRM filter proposed for gas turbine diagnostics need to be optimized for the specific application.

The problem of finding optimal integer weights of WRM filters using ACO is addressed. An analogy between the WRM filter weight optimization problem and the quadratic assignment problem is found. Test signals simulating abrupt and gradual faults are contaminated with noise and then used to find the WRM filter weights which minimize the noise. Numerical experiments are used to find the best parameters required for the ACO application. The WRM filters obtained in this study show noise reductions of 52–64% relative to the noisy signal compared to 41–55% for the RM filter, which uses unit weights. For the step, ramp and combination signals considered, noise reduction gains of about 9, 11 and 10% are obtained through filter optimization due to the use of weights obtained by ACO.

Appendix

Generate m ants with each ant being given different weight permutation w^k.

/*Initialization*/

Generate all possible different non-arranged permutations and randomly assign them to m ants.

Improve the weights $w^1, w^2, w^3, \cdots, w^m$ by the local search procedure. Let w^* be the best solution.

Initialize the pheromone matrix T

/* main loop */

For i = 1 to niter

/* solution construction */

For each permutation w^k(i) $(1 \le k \le m)$ do

Apply r pheromone trail swaps to $w^k(i)$ to obtain $\hat{w}^k(i)$

Apply the local search procedure to $\hat{w}^k(i)$ to obtain $\tilde{w}^k(i)$

End For

/* pheromone trail updating */

Update the pheromone trail matrix

End for if /* terminating condition */

References

Arce GR, Paredes JL (2000) Recursive weighted median filter admitting negative weights and their optimization. IEEE Trans Signal Process 483(3):768–779

Boryczka U, Kozak J (2015) Enhancing the effectiveness of ant colony decision tree algorithms by co-learning. Appl Soft Comput 30:166–178

Brownrigg DRK (1984) The weighted median filter. Commun ACM 27(8):807–818

DePold H, Gass FD (1999) The application of expert systems and neural networks to gas turbine prognostics and diagnostics. J Eng Gas Turbine Power 121(4):607–612

Dorigo M, Stutzle T (2005) Ant colony optimization. Prentice Hall of India Private Limited, New Delhi

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B Cybern 26(1):29–41

Gambardella LM, Taillard ED, Dorigo M (1999) Ant colonies for the quadratic assignment problem. J Oper Res Soc 50(2):167–176

Ganguli R (2002) Fuzzy logic intelligent system for gas turbine module and system fault isolation. J Propuls Power 18(2):440–447

Ganguli R, Dan B (2004) Trend shift detection in jet engine gas path measurements using cascaded recursive median filter with gradient and laplacian edge detector. ASME J Eng Gas Turbine Power 126:55–61

Karaboga N, Kalinli A, Karaboga D (2004) Designing digital IIR filters using ant colony optimization algorithm. Eng Appl Artif Intell 17(3):301–309

Lin YL, Chang WD, Hsieh JG (2008) A particle swarm approach to nonlinear rational filter modeling. Expert Syst Appl 34(2):1194–1199

Lu PJ, Hsu TC (2002) Application of autoassociative neural network on gas-path sensor data validation. J Propuls Power 18(4):879–888

Lu PJ, Zhang MC, Hsu TC, Zhang J (2001) An evaluation of engine fault diagnostics using artificial neural networks. J Eng Gas Turbine Power 123(2):340–346

Sampath S, Singh R (2006) An integrated fault diagnostic model using genetic algorithm and neural networks. J Eng Gas Turbines Power 128(1):49–56

Surender VP, Ganguli R (2004) Adaptive myriad filter for improved gas turbine condition monitoring using transient data. ASME J Eng Gas Turbines Power 127(2):329–339

Uday P, Ganguli R (2010) Jet engine health signal denoising using optimally weighted recursive median filters. J Eng Gas Turbine Power 132(4):41601–41608

Verma R, Roy N, Ganguli R (2006) Gas turbine diagnostics using a soft computing approach. Appl Math Comput 172(2):1342–1363

Volponi AJ, Depold H, Ganguli R, Daguang C (2003) The use of Kalman filter and neural network methodologies in gas turbine performance diagnostics: a comparative study. J Eng Gas Turbine Power 125(4):917–924

Yang R, Gabbouj M, Neuvo Y (1995) Fast algorithms for analyzing and designing weighted median filters. Signal Process 41(2):135–152

Zhao C, Zhang W (2005) Using genetic algorithm optimizing stack filters based on MMSE criterion. Image Vis Comput 23(10):853–860