Fault Detection of Aircraft Piston Engine Based on Exhaustive Database Search

Dubravko Miljković
Hrvatska elektroprivreda, Zagreb, Croatia
dubravko.miljkovic@hep.hr

Abstract - Modern aircraft piston engine monitor collects large amounts of data representing engine parameters. Data acquired during each flight is logged to a file. From the logs acquired during numerous previous flights a database of a healthy engine is formed containing feature vectors with selected engine parameters. Due to high speed processing of even modest microcontrollers, new feature vectors acquired during a flight can be compared in real time to all previous feature vectors stored in the database. Any significant discrepancy may indicate a problem in engine operation or engine fault. Such detection may alert a pilot and help prevent engine failure, partial loss of engine power or unplanned in-flight engine shutdown.

Keywords - aircraft; piston engine; engine monitor; fault detection; database

I. INTRODUCTION

Most general aviation aircrafts (popularly known as light aircrafts) are using aircraft piston engines. These engines provide a combination of power output and fuel consumption, suitable for such class of aircrafts [1]. However, these engines are many times less reliable than turboprop and jet engines that can be found on larger aircrafts, [2]. This fact is further aggravated because most general aviation aircrafts are single engine and engine failure or significant power loss may lead to an off airport landing that carries very high risks, particularly over inhospitable terrain (forest, water, urban areas). One way of increasing operational reliability of piston engines is application of engine monitors, also called engine analyzers and engine management system, [3]. Example of aircraft piston engine mounted on a piston engine aircraft is shown in Fig. 1. Complete or partial loss of engine power is quite a serious situation even with multi engine aircraft; because such loss severely reduces climb rate and produces asymmetric thrust that can be challenging to handle, particularly immediately after lift-off, before the aircraft reaches an altitude and speed. The idea behind the engine problem detection method described later in this paper is not to detect a particular type of engine problem, but to, in a simple way, detect engine parameters and their combinations (patterns) that are not usually encountered in a database formed from logs of previous flights. Such detection of aberrant parameters and patterns is to be used just to warn a pilot that something unusual is happening with the engine, and that adequate decisions can be made regarding the progress of flight. Accurate diagnostics is still by far in the domain of experienced maintenance personnel and is performed after the landing.

II. ENGINE MONITOR

Engine monitor tracks critical engine parameters, alerting to unexpected changes using its advanced exceedance monitoring system, [4-7]. Intelligent warning messages are displayed prominently, allowing pilots to immediately recognize and interpret a critical situation. Beside presentation and logging of engine operation, most engine monitors may be also used as an aid in leaning the engine. Every new generation of engine monitors brings some improvements in number of parameters they monitor and in format how they are presented on the display (use of vertical and horizontal bars, arcs and numerical data, in recent time almost always in color). A display of one modern engine monitor, [4] is shown in Fig. 2.
Many engine problems can be identified by a specific pattern that consists of CHT and EGT vertical bars. Such patterns are described in engine diagnosis charts that can be found in manuals that accompany engine monitor. Engine parameters are logged to the internal memory or memory card with the typical rate of one record every six seconds (this period is programmable in most engine monitors). Most common engine parameters that are monitored by modern engine monitors are listed in Table I, [8].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGT</td>
<td>Exhaust Gas Temperature</td>
</tr>
<tr>
<td>CHT</td>
<td>Cylinder Head temperature</td>
</tr>
<tr>
<td>OIL TEMP</td>
<td>Oil Temperature</td>
</tr>
<tr>
<td>OIL PRES</td>
<td>Oil Pressure 1</td>
</tr>
<tr>
<td>TIT1</td>
<td>Turbine Inlet Temperature 1</td>
</tr>
<tr>
<td>TIT2</td>
<td>Turbine Inlet Temperature 2</td>
</tr>
<tr>
<td>OAT</td>
<td>Outside Air Temperature</td>
</tr>
<tr>
<td>CDT</td>
<td>Compressor Discharge Temperature 1</td>
</tr>
<tr>
<td>IAT</td>
<td>Intercooler Air Temperature 1</td>
</tr>
<tr>
<td>CRB</td>
<td>Carburetor Air Temperature 1</td>
</tr>
<tr>
<td>CDT - IAT</td>
<td>Intercooler cooling</td>
</tr>
<tr>
<td>RPM</td>
<td>Rotations Per Minute</td>
</tr>
<tr>
<td>MAP</td>
<td>Manifold Pressure</td>
</tr>
<tr>
<td>%HP</td>
<td>% Horse Power 4</td>
</tr>
<tr>
<td>CLD</td>
<td>CHT cooling rate 2</td>
</tr>
<tr>
<td>DIF</td>
<td>EGT span 3</td>
</tr>
<tr>
<td>FF</td>
<td>Fuel Flow 1</td>
</tr>
</tbody>
</table>

optional, "fastest cooling cylinder," difference between the hottest and coolest EGT, *calculated

Basic engine diagnostics involve comparing engine parameters with the predefined limits (limit checking). Default limits (could be changed) for some parameters measured by JPI engine monitors are shown in Table II.

### III. REGIME SWITCHING

Engine operation during a period of flight goes through various regimes that correspond to flight regimes (idle, run-up, taxi, take-off, climb, cruise, descent and landing). Engine parameters in one regime may be quite different then in another regime. It may be difficult to compare engine parameters belonging to different regimes. For sake of comparison, it may be beneficial to separate engine operations into several regimes, [8,11]. There are few ways of separating different engine regimes, some are quite easy to determine and some are more precisely related to flight regimes and require GPS data, [11]. Among the variables that can be considered for regime separations, simple solutions are engine RPM and calculated % HP. The use of RPM is the easiest way for separating engine operation into regimes. The better way is the use of calculated percent of rated horsepower (% HP) as a variable for regime switching, [6]. It is dependent on accurate measuring of additional parameters. Calculated % HP is supplied by monitor considering Fuel Flow, RPM, OAT and MAP, but also environment temperature and altitude variations. Values of calculated % HP as a function of RPM and MAP (manifold pressure) are shown in Fig. 5.

\[
r = \left[ \frac{\% HP}{20} \right] + 1
\]

Separation of engine operation in regimes as a function of % HP in 20 % intervals is given by (1) and listed in Table III.
TABLE III. ENGINE REGIMES AS A FUNCTION OF CALCULATED % HP

<table>
<thead>
<tr>
<th>% HP</th>
<th>Engine regime r</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-19</td>
<td>1</td>
</tr>
<tr>
<td>20-39</td>
<td>2</td>
</tr>
<tr>
<td>40-59</td>
<td>3</td>
</tr>
<tr>
<td>60-79</td>
<td>4</td>
</tr>
<tr>
<td>80-99</td>
<td>5</td>
</tr>
<tr>
<td>100-119</td>
<td>6</td>
</tr>
</tbody>
</table>

Due to calibration errors, calculated % HP sometimes exceeds 100%, hence regime 6 is provided. Operating regimes during a typical flight are shown in Fig. 6. High power is applied during the preflight check and takeoff.

After analyzing JPI logs supplied as samples with the EZtrends software, [10] related to Flt#70 and Flt#71 (total 5685 records, 9 h 28 min) from the six-cylinder, 209 kW (280 hp) Continental TSIO-550-G engine and belonging to Mooney 20TN Acclaim aircraft, the following distribution across regimes is achieved, as shown in Table IV.

TABLE IV. NUMBER OF RECORDS PER REGIME

<table>
<thead>
<tr>
<th>Regime</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Records</td>
<td>221</td>
<td>119</td>
<td>281</td>
<td>4948</td>
<td>114</td>
<td>0</td>
</tr>
</tbody>
</table>

All engine parameters don’t react to change in calculated % HP with equal speed. Exhaust gasses, EGT and TIT, react quickly while CHT and OILT due to cylinder and engine metal mass react more slowly.

IV. FAULT DETECTION USING DATABASE SEARCH

Basic method for detection of engine problems is shown in Fig. 7. The database used in this fault detection method consists of feature vectors obtained from numerous engine logs acquired during previous flights (all flights are united in one large log) where engine problems have not been encountered. The log contains 600 records per one flight hour (depending on the selected rate). Engine parameters are in real-time compared to all previously stored feature vectors belonging to the same regime.

If the closest vector in the database is still too distant (above specific threshold) than the test vector containing recent engine parameters is considered to belong to an engine that has developed some kind of a problem.

A. Engine Parameters Selected for Comparison

Parameters that are of most interest in interpretation and analysis of engine logs belong to one of following three groups:

- Cylinder Head Temperatures CHTs, Temperatures of cylinder heads for each individual cylinder
- Exhaust Gas Temperatures EGTs, Temperatures of exhaust gasses for each individual cylinder
- Parameters common to all cylinders like FF – Fuel Flow, OILT – Oil Temperature and OILP – Oil Pressure.

There are, naturally, many other parameters available in modern engine monitor, but they are generally of less importance and are not used for identifying engine problems in this method.

Above mentioned parameters are static, in term, that they don’t reflect time dynamics of engine parameters. As such, these parameters cannot be used (without their time history) to detect problems, like EGT oscillations that precede exhaust valve problems, [12]. However, they can still be used for detection of most engine problems.

Selected engine parameters are used as elements of the feature vector \( q \), (2):

\[
q = [\text{CHT}_1, \text{CHT}_2, \ldots, \text{CHT}_n, \text{EGT}_1, \text{EGT}_2, \ldots, \text{EGT}_n, \text{FF}, \text{OILP}, \text{OILT}] \quad (2)
\]

B. Comparison Metrics

A variant of Euclidean distance is used in comparisons. The Euclidean distance, also called a Euclidean metric is the distance between two points in Euclidean space. Euclidean distance \( d_E(p, q) \) between vectors \( p \) and \( q \) is given by (3), where \( p_i \) and \( q_i \) are vector components:

\[
d_E(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (3)
\]

However, vector components contain EGT, CHT, FF, OILP and OILT values that belong to different ranges, have different variances and their differences will differ hugely. To balance the contributions of different engine parameters it is necessary to perform transformation called standardization, (4), that rescales data to zero mean and unit variance, where \( x'_i \) is normalized vector component.

\[
x'_i = \frac{x_i - \bar{x}_i}{\sigma_i} \quad (4)
\]

The average value \( \bar{x}_i \) of each vector component \( x_i \) is determined for each engine operating regime, (5). The values of the particular vector component are averaged using all stored feature vectors for that engine regime.

\[
\bar{x}_i = \frac{1}{N} \sum_{l=0}^{N-1} x_i \quad (5)
\]

Standard deviation \( \sigma_i \) is determined according to (6).
For each regime following normalization factors are to be calculated:

- average EGT, standard deviation EGT
- average CHT, standard deviation CHT
- average FF, standard deviation FF
- average OILP, standard deviation OILP
- average OILT, standard deviation OILT

Following normalization factors have been determined using previously mentioned JPI logs, Flt#70 and Flt#71, as shown in Table V.

**Table V. Averages and Standard Deviations for Engine Parameters in Various Engine Operating Regimes**

<table>
<thead>
<tr>
<th>Reg.</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Par.</td>
<td>Av. σ</td>
<td>Av. σ</td>
<td>Av. σ</td>
<td>Av. σ</td>
<td>Av. σ</td>
</tr>
<tr>
<td>EGT1</td>
<td>1103</td>
<td>79.1</td>
<td>1297</td>
<td>53.6</td>
<td>1374</td>
</tr>
<tr>
<td>EGT2</td>
<td>1060</td>
<td>84.7</td>
<td>1292</td>
<td>76.9</td>
<td>1412</td>
</tr>
<tr>
<td>EGT3</td>
<td>1086</td>
<td>79.2</td>
<td>1293</td>
<td>68.3</td>
<td>1396</td>
</tr>
<tr>
<td>EGT4</td>
<td>1066</td>
<td>78.3</td>
<td>1294</td>
<td>68.6</td>
<td>1384</td>
</tr>
<tr>
<td>EGT5</td>
<td>1176</td>
<td>90.8</td>
<td>1343</td>
<td>54.5</td>
<td>1379</td>
</tr>
<tr>
<td>EGT6</td>
<td>1096</td>
<td>79.0</td>
<td>1304</td>
<td>62.0</td>
<td>1372</td>
</tr>
<tr>
<td>CHT1</td>
<td>238</td>
<td>41.2</td>
<td>263</td>
<td>17.3</td>
<td>287</td>
</tr>
<tr>
<td>CHT2</td>
<td>224</td>
<td>42.8</td>
<td>257</td>
<td>16.7</td>
<td>291</td>
</tr>
<tr>
<td>CHT3</td>
<td>223</td>
<td>36.6</td>
<td>249</td>
<td>15.7</td>
<td>275</td>
</tr>
<tr>
<td>CHT4</td>
<td>226</td>
<td>37.2</td>
<td>253</td>
<td>15.6</td>
<td>280</td>
</tr>
<tr>
<td>CHT5</td>
<td>248</td>
<td>36.4</td>
<td>270</td>
<td>19.4</td>
<td>295</td>
</tr>
<tr>
<td>CHT6</td>
<td>227</td>
<td>37.7</td>
<td>255</td>
<td>16.8</td>
<td>286</td>
</tr>
<tr>
<td>FF</td>
<td>3.22</td>
<td>0.68</td>
<td>7.59</td>
<td>3.18</td>
<td>17.0</td>
</tr>
<tr>
<td>OILP</td>
<td>51.7</td>
<td>9.4</td>
<td>54.4</td>
<td>8.06</td>
<td>53.8</td>
</tr>
<tr>
<td>OILT</td>
<td>136</td>
<td>25.4</td>
<td>146</td>
<td>23.4</td>
<td>174</td>
</tr>
</tbody>
</table>

After some initial experiments, it was found that the squared difference in (3) doesn’t emphasize deviations enough (has not sufficient discriminatory power), and an exponent of four is used instead of two in new distance (7):

\[
d(p, q) = \sum_{i=1}^{n} (q_i - p_i)^4
\]

In real application, square root could be eliminated (with squared thresholds) to reduce computational burden.

C. Database

The database contains data for all engine operating regimes, as shown in Fig. 8. As can be seen, the database is divided into six sections corresponding to engine operating regimes. The database is filled with all normalized feature vectors (total 5685 feature vectors, 9 h 28 min) originating from JPI logs Flt#70 and Flt#71.

**Figure 8. Database containing data for various engine operating regimes**

**D. Finding Closest Feature Vector in the Database**

Test feature vector \( t \) is compared using comparison metrics (7) with all feature vectors \( q_r \) stored in the database that belong to the same engine operating regime \( r \) as the test feature vector and the minimum distance \( d_{min} \) is determined, where \( N_r \) is the number of feature vectors in the database that belong to engine regime \( r \), (8).

\[
d_{min} = \min_{i=1,N_r} (d(t, q_r))
\]

E. Why Not Clustering?

Use of clustering (K-Means, Isodata, Kohonen SOM) could be potentially beneficial, as it would substitute a large number of feature vectors with a much smaller number of cluster prototype. This would reduce the size of the database and reducing search time. One solution with just one prototype per regime obtained by averaging is described in [8]. However, clustering is not used in this method for two reasons. First, regime switching is used and it has already partitioned database into smaller subsections. Today, with modern microcontrollers it is not difficult to compare thousands of feature vectors in a real time. Second, the precision of the comparisons is better when original feature vectors are used instead of cluster prototypes.

F. Suspicious Feature Vector Detection

Each new vector of engine parameters is compared to all previously stored vectors in the database belonging to the same engine operating regime. As the database is not too large (several thousands of feature vectors) this comparison can be accomplished in a real time. Distance between new feature vector and all stored feature vectors for a particular regime is determined. The statistical distribution of intraset distances (within the same regime) is shown in Fig. 9. Values for \( P_{95} \) and \( P_{99} \) percentiles for various engine operating regimes are shown in Table VI. The corresponding percentiles in form of Box-Jenkins plot are shown in Fig. 10. Decision thresholds for each engine regime are determined based on the statistical distribution.
of distances among feature vectors belonging to that regime. One way of determining the threshold is to consider acceptable numbers of occurrences during one flight. If the engine is operating correctly, this number should be very small or zero. However, due to lack of sufficient data necessary for determination of very high percentiles, value of 99 percentiles, may be chosen for appropriate threshold. Occasional and isolated threshold exceedances may be tolerable, particularly when they will be further reduced by temporal filtering. Different threshold exceedances may be tolerable, particularly when they will be appropriate threshold. Occasional and isolated threshold exceedances in a period of time. In this way error must repeat few times within a specified period, Fig. 11, otherwise is not considered and error. An alert is issued if the number of threshold exceedances in a period of time \( N_e \) is greater than \( N_T \), (10):

\[
N_e > N_T \tag{10}
\]

(e.g., collected within the last minute, sliding window)

So, if the new test vector of engine parameters is only a glitch, it would be filtered out by temporal filtering. However, if it is something more persistent, than temporal filtering will not filter it out and alert will be issued. Accurate adjustment of statistical and temporal thresholds is a tradeoff between sensitivity and false detections.

V. PROPOSAL FOR THE INCORPORATION OF LEARNING

Previously described basic method could be further extended by introduction of automatic learning. Once engine feature vector has been identified as a normal, it can be added to the existing database, Fig. 12. However, how can we know that these feature vectors belong to a healthy engine? One way would be to wait a period of time after this feature vector \( \mathbf{s} \) is collected and stored in the temporary buffer before it is added to the database. How long should one wait? There may be a slow drift of engine parameters toward the parameters that appear in a problematic engine.

Following requirements may be set on new feature vectors to update existing database:

- Feature vectors to update the database must be of the smaller distance to the closest stored vector than a threshold \( P_{99,r} \), or using \( P_{65,r} \) instead (with no \( K \)), (11).

\[
d_{\text{min}} > P_{99,r} \tag{9}
\]

- Feature vectors are held in a temporary buffer for a specified period (e.g., duration of 20 flights). If a fault develops in 20 hour period. So, feature vectors \( \mathbf{s}(t) \) are put in a “quarantine” for 20 hours. If a fault develops in 20 hour period, stored normal feature vectors are discarded. Otherwise, they are used for update, \( \mathbf{s}_n(t) \). With the data acquisition rate of one record every six seconds there will be 12000 records in 20 hours, (12):

\[
\mathbf{s}_n(t) = \mathbf{s}(t - 12000) \tag{12}
\]

After that, they can be used to update database in one of following ways:

- Adding a new feature vector \( \mathbf{s}_n(t) \) till space is available (database size and increases comparison time). Database size has also its limits.
- Replacing older feature vectors with new ones \( \mathbf{s}_n(t) \) using FIFO (First-In First-Out) principle.
- Updating closest existing feature vector with new one using exponential smoothing and a very small \( \alpha \). This would result in fine tuning database feature vectors.

Exponential smoothing is a method for smoothing time series data using the exponential window function, (13),(14). Smoothing is performed on all elements \( \bar{s}_i \) of a vector \( \mathbf{s} \).

\[
\bar{s}_i(0) = s_i(0) \tag{13}
\]

\[
\bar{s}_i(t) = \alpha s_i(t) + (1 - \alpha)\bar{s}_i(t - 1), \ t > 0 \tag{14}
\]

where \( \alpha \) is the smoothing factor, (15):

\[
0 < \alpha < 1 \tag{15}
\]

VI. VALIDATION OF THE METHOD

Only basic method (for detection without learning) was tested. Testing the method with learning would require too much engine monitor data that are not easy to obtain. Due to highly reliable engines, and hence the lack of engine monitor logs that include specific engine problems, artificial faults have been used. Three artificial faults have been generated from log records belonging to a healthy
engine, by modifying engine parameters according to parameter deviations described in JPI manual containing catalogued engine failure patterns as they appear on the display of an engine monitor, [7] and as described in [11,14]. The regime used is $5, P_{91,5} = 61,2639$ (Table VI).

A. Loss of EGT for One Cylinder

First engine problem used in a test is the loss of EGT for one cylinder engine and engine running rough. This is probably caused by stuck valve. The corresponding catalogued diagnostic pattern (example, different cylinder may be affected) is illustrated in Fig. 13 and deviation described by (16), [14].

Figure 13. Catalogued engine fault pattern for fault in case A

$EGT_{\text{min}} < 600$ \hspace{1cm} (16)

Artificial test vector is $t_i$ (17) and distances (18) and (19).

$t_i = [1332 \ 1388 \ 1359 \ 546 \ 1345 \ 1316 \ 310 \ 317 \ 306 \ 313 \ 317 \ 316 \ 29.4 \ 53184] (17)$

$d_{\text{min}} = 222.19200$ \hspace{1cm} (18)

$d_{\text{min}} > P_{91,5}$ \hspace{1cm} (19)

B. Decrease of EGT for One Cylinder at Low RPM

The second test problem is caused by a decrease of EGT for one cylinder. This problem appears when the intake valve is not opening fully. The corresponding catalogued diagnostic pattern (example, different cylinder may be affected) is illustrated in Fig. 14 and deviation described by (20), [14].

Figure 14. Catalogued engine fault pattern for fault in case B

$\text{DIFF} > 500$ and $\text{RPM} < 1500$ \hspace{1cm} (20)

Artificial test vector is $t_i$ (21) and distances (22) and (23).

$t_i = [1332 \ 1388 \ 1359 \ 546 \ 1345 \ 1316 \ 310 \ 317 \ 306 \ 313 \ 317 \ 316 \ 29.4 \ 53184]$ (21)

$d_{\text{min}} = 80.60468$ \hspace{1cm} (22)

$d_{\text{min}} > P_{91,5}$ \hspace{1cm} (23)

C. EGT and CHT not Uniform

In third test problem EGT and CHT temperatures are not uniform across all cylinders in the engine (large deviations are present). This is caused by dirty fuel injectors or fouled plugs. The corresponding catalogued diagnostic pattern (example, almost endless variations are possible within constrains) is illustrated in Fig. 15 and deviation described by (24), [14].

Figure 15. Catalogued engine fault pattern for fault in case C

$\text{EGT}_i, \text{CHT}_i < 3 \ OR \ \text{EGT}_i, \text{CHT}_i > 6$ for any $i, i=1, \ldots, N$ \hspace{1cm} (24)

Artificial test vector is $t_i$ (25) and distances (26) and (27).

$t_i = [1332 \ 1388 \ 1358 \ 546 \ 1345 \ 1316 \ 310 \ 317 \ 306 \ 313 \ 317 \ 316 \ 29.4 \ 53184]$ (25)

$d_{\text{min}} = 64.40012$ \hspace{1cm} (26)

$d_{\text{min}} > P_{91,5}$ \hspace{1cm} (27)

From the previous examples, problematic test vector has exceeded the threshold value and as such examples are classified as problematic in all three cases. Due to the large number, of vector components, i.e. 15, (2), the method is sensitive only to larger variations. To increase sensitivity exponent of four in the squared difference is used in the distance (7) instead of two in Euclidean distance (3).

VII. CONCLUSION

Loss of engine power in an aircraft has always been a serious threat. To diminish the frequency of such occurrences application of digital engine monitors is recommended. An approach for detection of engine problems based on an exhaustive database search of previously stored feature vectors of selected engine parameter is described. This approach doesn’t pretend to identify the cause of engine problems (this would require detailed and carefully designed expert system), but to warn pilot on unusual engine parameters that were not present in previous flights, and raise suspicion that engine reliability may be at stake. Combination of statistical and temporal threshold is used to reduce spurious alerts and hence generate more robust alerts. The described method was tested on three engine problems represented by artificially generated faults. An extension of this method that includes automatic learning and modifying database is proposed.

REFERENCES

[1] M. Bush, Mike Busch on Engines, Savv Aviator, Wayoming, 2018