Readme

Revision 1: 9/20/2012, Dinkar Mylaraswamy, Honeywell.

Data Description

The data contains time trace of condition indicators (CI) and diagnostic monitors from an aircraft equipped with four engines. There are four files corresponding to the data from the four engines. These files are named as follows:

File name	Description
pv_trace_1.csv	Data from engine #1
pv_ trace _2.csv	Data from engine #2
pv_ trace _3.csv	Data from engine #3
pv_ trace _4.csv	Data from engine #4

Filename Convention: pv_trace_*k*.csv, where k corresponds to the *engineid*. Since we have 4 engines per aircraft, k = 1, 2, 3, 4.

Every file has the same format, a comma separated text file. It is envisioned that the user will write one import function to can read the information from all 4 files. The appendix of this document provides a sample code for importing the data to Matlab.

The CSV files contain processed data. In contrast, the original recordings from the Flight Data Acquisition and Management System (FDAMS) contain raw sensor values in ARINC 717 format. This raw sensor value is analyzed by a series of proprietary algorithms to generate the following two information streams:

1. **Diagnostic Monitors**. These are enumerated indicators. Possible values are 0, 1 and -1. A "zero" value provides exonerating evidence for a pre-defined set of failure modes. A "one" value provides indicting evidence for the same set of failure modes, indicating that one or more of these failure modes may be occurring in the engine. A "negative one" value indicates an unknown state arising from the inability of the algorithm to provide any meaningful value. Each monitor has a unique integer ID and an ambiguity set containing one or more failure modes. Each failure modes has a unique integer ID as well.

   In this data set, the following diagnostic monitors are provided.

Table 1: List of Diagnsotic Monitors included in the data.

Monitor ID	Mnemonic	Description

7401	noStart	The engine failed to start
7404	HighEGT	The temperature was too high during start
7402	slowStart	Engine took long time to start
7408	multStart	Multiple start attempts were made
7403	hungStart	Engine did not reach idling speed
7409	phOneDwell	Starter cutoff speed was too low
7418	starter	Torque from the starter was too low
7417	ignitor	Igniter could not provide spark
7419	highRollEGT	The temperature was too high during shutdown
7410	abruptRoll	The speed was too high during shutdown
7411	rollBearing	High vibration from the engine bearings
7412	medTempMargin	Temperature margin is trending yellow
7413	lowTempMargin	Temperature margin is trending red
7463	ovSpdShutdown	Over speed uncommanded shutdown
7464	ovTempShutdown	Over temperature uncommanded shutdown

2. **Condition Indicators**. These are parametric values derived from various engine sensors. Consequently, they can attain any real value and have engineering units. Typically the engine manufacturer specifies a nominal range for these condition indicators, such that when the CI remains within this range, the engine is considered as operating as per its intended design. However, one must view these bounds as mere recommendations and often represent a conservative threshold. Incipient problems can be occurring while these CI remain within the range defined.

   The underlying algorithm may be unable to calculate the CI value for a given flight. For example, if the flight did not contain a takeoff then none of the take-off related CI can be calculated. In this case, non availability of CI is indicated as a NaN (not a number).

   In this data set, the following condition indicators are provided.

Table 2: List of Condition Indicators (CI) included in the data.

ID	Mnemonic	Units	Description
740301	startTime	seconds	The time to start the engine. Typically between 20 and 50 seconds.
740302	idleSpeed	% (rpm)	Idling speed attained under no-load conditions (normalized). Typically between 40 and 55 % rpm
740303	peakEGTC	% (^{o}C)	The peak temperature at startup (normalized). Typically between 80-90 % C
740304	peakN1TKO	% (rpm)	The peak fan speed during takeoff (normalized). Typically between 80-90 % rpm
740305	tkoN1	% (rpm)	The average fan speed during takeoff The peak temperature at startup (normalized). Typically between 80-90 % rpm

740306	tkoEGTC	% (^{o}C)	The average temperature (normalized). Typically between 80-90 % C
740307	rollTime	seconds	The time to shutdown the engine. Typically between 20 and 40 seconds.
740308	resdTemp	% (^{o}C)	The average temperature during the engine shutdown (normalized). Typically between 30-45 % rpm

Data is listed in the CSV file using a comma delimiter. Each file contains several lines, each line contains the state of the diagnostic monitors that fired and the CI values from one flight. Hence if the file contains N lines, then it corresponds to the above mentioned data from N flights. Please note that all diagnostic monitors may not fire for every flight. Further, CI values may not be available for each flight. The format of each data line is as follows:

1. The first entry is the flight timestamp. It is not a real number, but just a monotonically increasing index. Use this index to synchronize the data from all four engines.
2. An integer that indicates the number of diagnostic monitors that fired for this flight. This number will always be Let denote this number.
3. There will be integer entries listed as pairs of As described earlier, the ID can be any of the integer value listed in Table 1, and the monitor state can either be a 0, 1, or -1.
4. An integer that indicates that the number of CI values that could be calculated for this flight. This number will always be . Let denote this number.
5. There will floating point numbers listed as pairs of . As described earlier, the ID can be any of the integer values listed in Table 2, and the numeric value is a floating point number.

The maximum number of entries on a single line in the CSV files shall be 47 as described by the following formula.

Importing the data

Matlab commands are indicated using Helvetica font. % indicates a comment line.

```
% Open the file for reading.
1. fp = fopen('pv_trace_3.csv', 'r');

% Scan a maximum of 47 values to retrieve the data from one flight.
2. flgIndex = textscan(fp, '%f', 1, 'delimiter', ','); % The first floating point number is the flight
   index or timestamp
```

3. nDiagMon = **textscan**(fp, '%d', 1, 'delimiter', ','); % The next integer is the number of diagnostic monitors that fired.
4. dM = **textscan**(fp, '%d %d', nDiagMon, , 'delimiter', ',');
   % Read the next *nDiagMon* pairs. dM{1} contains the ID and dm{2} contains their states.
5. nConInd = **textscan**(fp, '%d', 1, 'delimiter', ',');
   % The next integer is the number of diagnostic monitors that fired
6. cI = **textscan**(fp, '%d %f', nConInd, , 'delimiter', ',');
   % Read the next *nConInd* pairs. cI{1} contains the ID and cI{2} contains their numeric values.

   % The data from one flight is imported.
7. **if feof**(fp), **return**; **end**
   % Check if you have reached end of file. If yes, then exit.

   % If end-of-file is not reached. repeat steps 2—6 to get data from the next flight.