AIAA SciTech
5-9 January 2015, Kissimmee, Florida
53rd AIAA Aerospace Sciences Meeting

Estimating Landing and Take-off Cycle Parameters Using MATLAB

Jacob Varney[1], Pedram Motevalli[2], Demetrios Katsaduros[3], and Matthew Prall[4]
Purdue University, West Lafayette, IN, 47906, United States

Mary E. Johnson, Ph.D.[5]
Purdue University, West Lafayette, IN, 47906, United States

Landing and Takeoff Cycle (LTO) data is used to estimate emissions in the Emissions and Dispersion Modeling System (EDMS) software. General aviation LTO data available in EDMS may not be an accurate representation of the flight data at a specific airport. Therefore, flight data is collected from aircraft at the specific airport and inputted into spreadsheets. The analysis of these spreadsheets is time consuming and tedious. In addition, multiple sets of flight data must be analyzed to determine appropriate parameters for a LTO cycle. A simpler process is needed to quickly parse multiple sets of flight data to identify fuel flow rates and time in modes for each phase of the LTO cycle. A MATLAB program was developed to improve the efficiency of this process and to standardize phase identification. Sorting the data into LTO phases consists of finding the point at which one phase of the LTO cycle ends and another phase begins. Selection of that point is based on engine parameters at a given time or the change of engine parameters over a given length time. Parameters can be difficult to identify in some data sets due to anomalies in the flight logs. Data sets with anomalies that are incompatible with the program may be removed from the spreadsheets. Outputted data will be formatted to enable transfer to EDMS for estimating engine emissions. This paper explains the development and testing of the scripts used in the MATLAB program, the engine parameters used in each phase of the LTO cycle, and the use of EDMS in emissions estimates.

I. Introduction

Data recording devices have long been installed on commercial aircraft. The information gathered from data recording devices can be used in studies to analyze correlations between certain flight and engine parameters. Such devices may be installed on single engine fixed wing aircraft. The Cirrus SR20 used in test flights at Purdue University, for example, utilizes the Garmin 1000 system. Flight data is recorded on a SIM card, which is either downloaded manually or through the use of a linked connection to the device. Each record within the downloaded file represents the conditions of the aircraft at a given time. These records are saved at approximately one record per second, from start up to shut down of the plane. For flights lasting over thirty minutes at a recording rate of one record per second, a data set may contain at least 1800 records. This large amount of data is multiplied by the large number of flights performed by a training fleet made up of sixteen aircraft, resulting in a huge mass of data that can be analyzed to gather flight information. The software developed by this process is necessary to analyze the large number of spreadsheets containing flight data.

In the specific application of determining phases of flight for emissions data (Katsaduros, Prall, and Johnson, 2014), it is necessary to determine the phase transitions and corresponding average values for certain parameters in each phase. This process can easily be done by an individual with knowledge of the standards set forth by the International Civil Aviation Organization (ICAO) for the Landing and Takeoff Cycle (LTO). However, that process is tedious to do manually. The analysis of each data set would require researchers to review hundreds of

[1] Student, Department of Mathematics, Purdue University, Sophomore
[2] Student, Department of Aviation Technology, Purdue University, Junior
[3] Student, School of Electrical Engineering, Purdue University, Junior
[4] Student, School of Mechanical Engineering, Purdue University, Junior
[5] Associate Professor, Department of Aviation Technology, Purdue University

spreadsheets and look for significant entries, likely needing to produce graphic outputs for phase transitions that are difficult to determine. Furthermore, the researcher must analyze the values for each phase and output these to another spreadsheet to be used for further analysis. In order to reduce the effort required by the researcher and to standardize results, the software was developed in MATLAB in accordance with the model in Figure 1. This model describes the sequence by which the program was developed. This sequence is appropriate given the result of each phase is required to perform later processes: input data must be formatted to have phases determined and phases must be determined to test for flights with erred phase outputs. The focus of these analyses will be averages for specific parameters required by the Emissions and Dispersion Modeling System (EDMS).

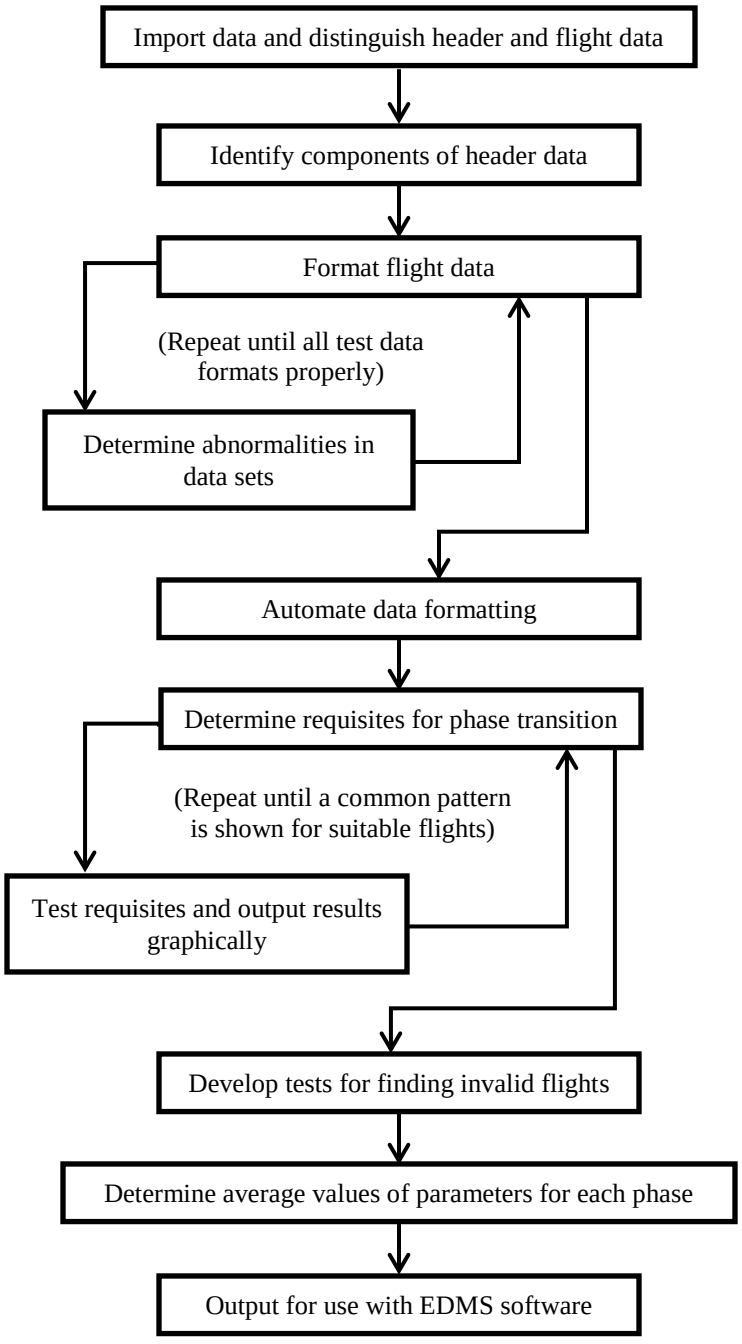

Figure 1. Model of process by which a set of spread sheets will be analyzed.

II. Materials

The intended inputs for this software are comma separated value (CSV) files - otherwise referred to as spreadsheets - containing SIM card data from the aircraft. The first row is eight fields wide whereas the remaining rows are sixty-six fields wide. These CSV files are formatted with the initial three lines as header data and the remaining lines as flight data as shown in Table 1.

#airframe_info	log_version="1.00"	airframe_name="Cirrus SR20"	unit_software_part_number="006-B0319-9B"	unit_software_version="11.11"	system_software_part_number="006-B0764-09"	system_id="24A5EF377"
#yyy-mm-dd	hh:mm:ss	hh:mm	ident	degrees	degrees	ft Baro
Lcl Date	Lcl Time	UTCOfst	AtvWpt	Latitude	Longitude	AltB
7/2/2012	09:43:25	-04:00				471.8

Table 1. 6X7 excerpt from CSV file. *The first three rows contain header data whereas the remaining rows are flight data.*

The header data contains information pertaining to the aircraft and flight parameter descriptions. Within the first row of the header data are the airframe name (airframe_name) and system I.D(system_id). These fields are required in order to identify the aircraft that the spreadsheet describes. The data in the next two rows are headers for each column of parameters. The first of these two rows contains the units for the parameters, while the second row contains the names of the parameters.

The flight data contains information pertaining to the flight parameters. The parameters referenced by this data include local time, location, details of movement, orientation, engine parameters, and environmental measures. The local time is given in one of two formats: “dd/mm/yyyy” or “yyyy-mm-dd”. Location is formatted in decimal notation - as opposed to a degree notation - and describes latitude, longitude, and altitude. Details of movement are numeric values describing aspects such as indicated airspeed or ground speed. Orientation refers to the degree at which the aircraft is rotated on a given axis. Engine parameters refer to the state of the engine, including fuel flow rate, temperature, and RPM. Environmental measures record details pertaining to the region outside the aircraft.

Within the flight data, inconsistencies in records may occur, including missing data, interrupted records, and timestamp irregularity. Missing data from the SIM card includes any fields that are missing. The missing data is not always regular in shape; data may be missing for an entire row in the data set or may be missing for a certain parameter. Rows following missing records may also contain null values for certain parameters. Interrupted records occur when the aircraft is shutoff and the record being written was unable to be completed. This interrupted record is the last row on the spreadsheet. Timestamp irregularity refers to the events in which timestamps are not evenly incremented. Timestamps usually increment by one second in every row, but often will either not increment or increment by two after a row.

III. Program Description

The resulting program accepts as input a CSV file as described within Materials and returns to the user the measurements necessary for input to EDMS software. The program is the integration of several functions to determine what is needed to output to the EDMS software. The design allows the user to repeat the experiment for other parameters and append data by modification. See Figure 2 for top-level function design.

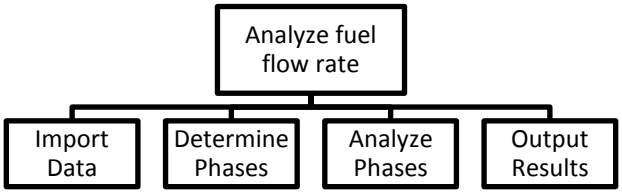

Figure 2. Top-level design for determining fuel flow rates for LTO cycles.

"Import Data" pertains to all operations in which CSV files are copied to matrices in MATLAB. In order to maintain use of these matrices, each is appended to a .MAT file - a MATLAB data file - from which the matrices are extracted and used for analysis in later functions. This process works as described below with reference to Figure 3. The dumped text is used to locate certain identifiers for each row, such as commas and new line characters. The rows and fields are then defined on the basis of the location of new line characters and commas respectively. Separation of the header data from the flight data places the three rows of header data into its own matrix for analysis. From this header data, the function seeks out the corresponding identifiers for the aircraft, which are system_id and airframe_name. Using the remaining data, the number of rows is determined based on the number of new line characters within the array. A matrix of doubles will be allocated for the flight data, where the number of rows will reflect the number of rows in the data set, excluding interrupted records, and the number of columns will reflect the columns in the data set. With the exception of "Lcl Date" (Local Date) and "Lcl Time" (Local Time), all data will be inputted as either 0, if string irrelevant to studies; an integer representing a key number, for strings of interest within the scope of study; or a corresponding double, if numerical. "Lcl Date" and "Lcl Time" are combined using MATLAB operations to form a serial date number, from which one can calculate elapsed time and return timestamps in any preferred format using other MATLAB operations. The function will begin writing the data for each column to the matrix. Values that are undefined prior to engine startup are defined as zero. All missing data is defined as zero, excluding timestamp, which will be defined as the timestamp prior to the missing data. The airframe will be checked against all airframes in the current key. If an airframe of this name has not yet been tested, it will be written to the key. The key number for this airframe will be assigned to the corresponding column in the matrix. This process will be repeated for every file in the given folder and saved to "tables.mat".

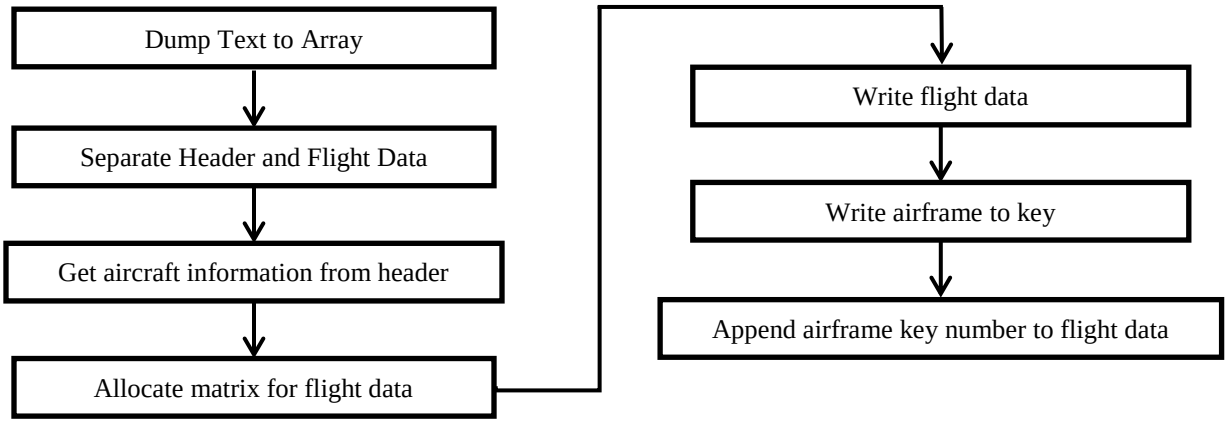

Figure 3. Insert Data Process.

"Determine Phases" corresponds to all operations in which the flight data are searched for indications of phase transition. The input is a .MAT file containing the matrices determined by the operations under "Import Data", and the output is the same .MAT file for which the matrices have phase numbers in the corresponding columns. Valid phase numbers will be displayed as the corresponding phase numbers, whereas phases that could not be determined will be negative one. See Table 2 for specific phase numbers. The phases include both those in the LTO cycle as well as phases between those in the LTO cycle. The process works as described below. From Startup

to Cruise, the phases are analyzed chronologically. In each phase, an indicator is sought that will denote a transition to the next phase. From Cruise to Taxi-in, phases are analyzed in reverse chronological order. In each phase, starting with taxi-in, an indicator is sought that would have indicated a transition to this phase. The rationale for reverse chronological order is to avoid processing "touch and goes", which are not considered within the scope of this project, but do exist which would cause discrepancies in analyzing each data set. For all phases, there is allowed an elapsed time requirement. This would require that an indicator be present for multiple consecutive rows, with a number of exceptions defined for the phase. Table 3 details the requirements for phase transitions, as well as the required elapsed time and number of exceptions. Along with these requirements are backtracking operations, which will write over previous phase numbers for rows that were tested for requisites over the required time period and positively indicated a phase transition. If the phase analysis is unable to determine each phase, the flight is ruled invalid and negative one is assigned as the phase number for all rows. Following this process, the flight data set and its phases are then tested to ensure phases are reasonable. The tests check for irregularities in phases that would denote that the data set was either unsuitable for the scope of the research or unable to be correctly processed by the phase determining algorithm. This process will be repeated for every matrix contained in the .MAT file passed as input and each will be written to a .MAT file of the same name, effectively overwriting the .MAT file passed to the function.

Phase	Phase Number
Startup	0
Idle	1
Taxi-out	2
Takeoff	3
Climb	4
Cruise	5
Descent	6
Patterned Flight	7
Taxi-in	8
Cannot Determine	-1

Table 2. Corresponding Phase Numbers.

"Analyze Phases" corresponds to all operations performed in order to generate the data required for output. Given a .MAT file containing sets with analyzed phases, these operations return average values for emissions and length times for each phase in each data set. In order to store these, a new structure is created that stores a matrix for each spreadsheet. This matrix will be composed of the average values for each phase where the row number will indicate the phase number and the column number will indicate a parameter.

"Output" corresponds to all operations that write data in the expected format for use with the EDMS software. The specific output will be a CSV file containing the required averages for processing the LTO cycles.

IV. Development

The development cycle closely follows the process described by Figure 1, with the intent to be designed as Figure 2 at the top level, while lower level functions may still be used for future project design.

In order to import the data and separate components, properties of the text file, including how rows and entries were separated and how many rows were in the header, were speculated from the text file and dumped text file in MATLAB. With the exception of the first and last row of each CSV file, each row contained the same number of commas, fields, thus each row was concluded by a new line. Using these, the indices were identified for the start of each new line. The first three lines, by speculation, were found to be header data. These three lines would be written to another array and subsequently removed from the original text dump, leaving the flight data.

The process for identifying key header data was also supported by speculation. Given that the fields in the header text containing the airframe name and system identification also always contained "airframe_name=" and "system_id=" respectively, using string comparison functions in MATLAB allowed for searching for these terms and copying the respective terms.

Transition	Required Parameter(s) for Transition to Phase ([AND] and [OR] represent corresponding logical connectors)	Elapsed Time Required for Transition	Maximum Number of Exceptions
Startup -> Idle	RPM > 0 rpm	1 second	0 seconds
Idle -> Taxi-out	RPM > 1200 rpm [AND] 20 > Ground Speed > 5 knots	5 seconds	0 seconds
Taxi-out-> Takeoff	RPM > 2000 rpm FF > 10 gph	10 seconds	0 seconds
Takeoff-> Climb	RPM > 2650 rpm [AND] Altitude relative to start of takeoff > 50 ft. [AND] FF > 10 gph	0 seconds	0 seconds
Climb-> Cruise	1900 < RPM < 2400 rpm [OR] Vertical Speed < 0 ft. per minute	20 seconds	0 seconds
Taxi-in-> Patterned Flight	Altitude Relative to end of records > 50 ft.	0 seconds	0 seconds
Patterned Flight-> Descent	Indicated Airspeed > 100 knots [AND] Vertical Speed <= 0 ft. per minute	50 seconds	5 seconds
Descent-> Cruise	RPM > 1700 rpm [AND] Vertical Speed >= 0 ft. per minute [AND] Altitude relative to end of descent > 100 ft.	13 seconds	0 seconds

Table 3. Required conditions to transition from phase to phase. *The order corresponds to the order in which each phase is determined.*

Several test cycles were required in order to properly format flight data for irregularities and efficiency. As stated in Materials, several spreadsheets contained missing data or interrupted data. While normal spreadsheets formatted properly, errors would arise in others, requiring that the abnormality be identified and issued in the formatting process. Formatting missing data required inserting either zero for the missing data or the copying the previous valid timestamp for missing timestamps. The rationale for making missing data 0 is that given the unknown intent of a pilot during flight, missing data cannot be inferred from previous or future data. The rationales for making the timestamps the same as the previous are the timestamp irregularities, which make it impossible to predict, and setting them to zero would interfere with future operations. Efficiency became a focus after data import was successful for all test sets. The main distinguishing factor in efficiency was storage type, being either cell or matrix. Cells are capable of holding strings and doubles in a manner that makes table generation easy, however, their usage is more expensive computationally than matrix usage. Matrices are capable of performing similar operations by being composed of only doubles, using keys in order to identify string values. Ultimately, given two implementations using matrices and cells, the computational speed for data import using cells was always slower than data import using matrices.

Once data can be successfully formatted for the test sets, automation of data formatting and a process to store the matrices is required. A folder containing CSV files would have to be analyzed in order to prevent the user from repetitive operations regarding selecting files for import. This automation only required that exception for cases that could be handled by the data import algorithm.

In order to determine phases for each data set, a standard set of requisites for phase transitions would be necessary. These transitions would rely on the standards for phase of flight as set by the ICAO, as well as testing in order to ensure the phase transitions were accurate (ICAO, n.d.). In order to improve the accuracy of phase transitions, the altitude of each data set was plotted against the elapsed time, as in Figure 4.

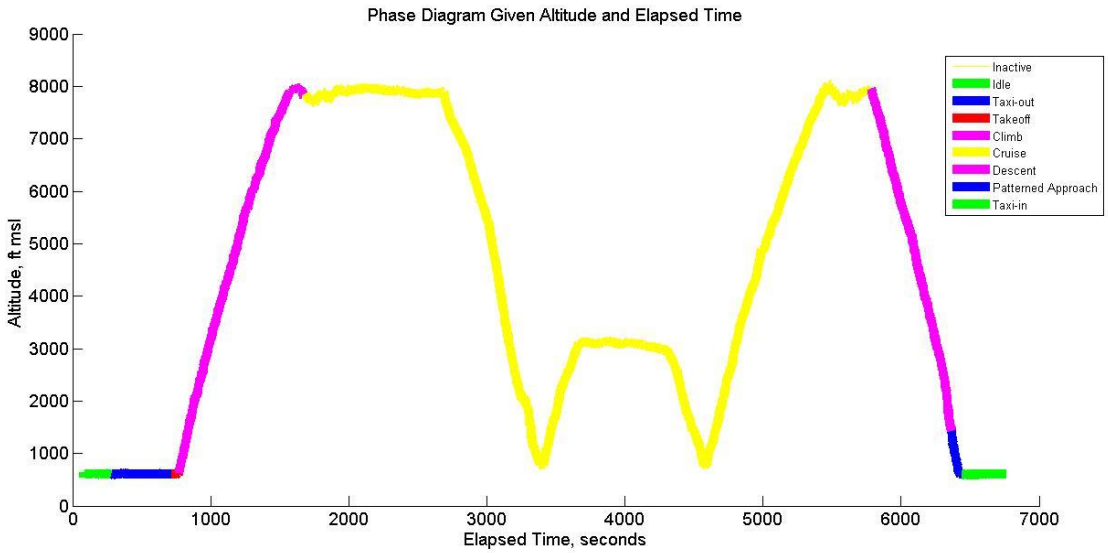

Figure 4. Phase testing diagram for a data set.

The plot is then separated by colors to display the differences between phases, and analyzed This analysis will determine if the flight both contained the required phases of flight in order to fit the LTO cycle and that the phases were labeled correctly given the appearance of the plot. In order to test if a flight contained all the requisites for the LTO cycle, a series of tests, listed in Table 3, were developed to analyze each phase as determined by the process. In order to improve the precision of the phase labeling, other parameters involved in the phase transition were tested aside from those defined by the ICAO. These other parameters would help distinguish changes in the phases using details about the aircraft's motion and relative location.

V. Application

EUROCONTROL's Emission Dispersion Modeling System (EDMS) (EDMS, 2013) is used to generate emissions estimations for a specified airport. EDMS utilizes airport-specific inputs to generate emission dispersion models and emission outputs of carbon monoxide, total hydrocarbons, non-methane hydrocarbons, volatile organic compounds, nitrous oxides, and particulate matter. Output from the processed flight data files is used as input into EDMS in order to create customized airport inputs. Parameters of interest are time in mode and fuel flow rates for each phase, or operational mode, of the landing and takeoff cycle.

Emissions of an aircraft type for one year can be estimated using the following formula (adapted from EPA, 1993):

$$E_k = n \sum_{i=0}^{j} (DUR_j * FFR_j * EI_{jk}) \quad (1)$$

Where:

E = Emission [kg yr^{-1}] of species (k)

n = number of LTOs for aircraft type per year

DUR_j = Duration of operational mode (j) [seconds];

FFR_j = Fuel flow rate for operational mode (j) [(kg fuel) sec^{-1}]

EI_{jk} = Emission inventory [g (kg fuel)$^{-1}$] of species (k) for aircraft type and operational mode (j)

Emission output is a function of fuel flow for operational mode, duration of operational mode, and emission inventories. Emission inventories are taken from the latest ICAO engine exhaust data bank and EUROCONTROL Base of Aircraft Data in emission calculations, and will remain fairly constant for a given aircraft at any given airport. Fuel flow rate and duration of operational mode will vary significantly by aircraft and airport, and are the parameters found with the MATLAB program in this project. These parameters are imputed manually into EDMS through the use of "User-Created Aircraft". The EDMS interface for User-Created Aircraft is shown in Figure 5.

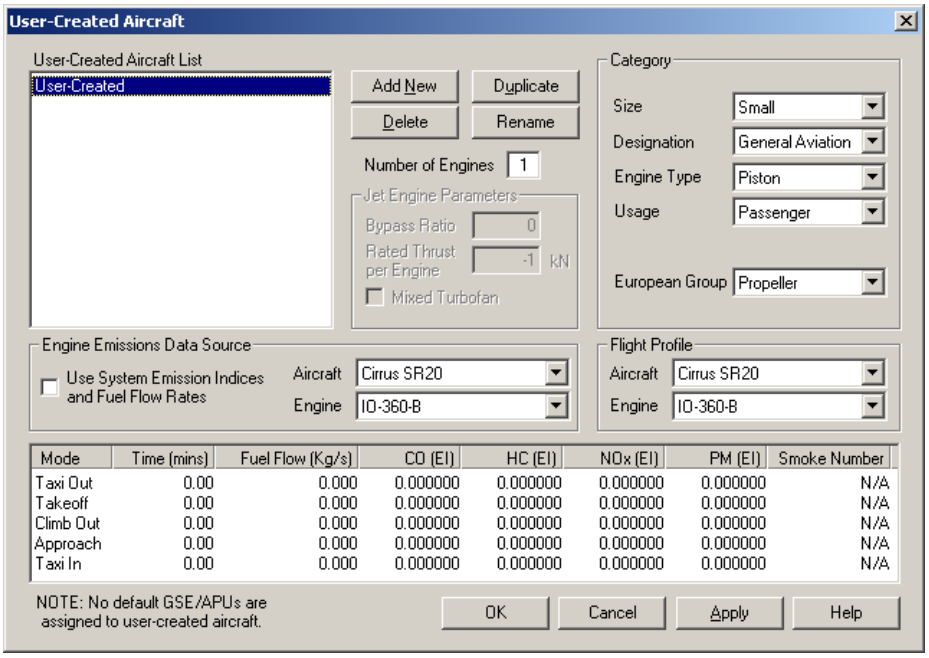

Figure 5. EDMS User-Created Aircraft interface (EDMS, 2013).

For each User-Created Aircraft, background information of the aircraft is required. Since flight data is taken from Cirrus SR20s, the User-Created Aircraft is defined as a small, general aviation, piston engine configured, passenger plane. Emission inventories are taken to be the same as inventories referenced from the database for the standard Cirrus SR20. Fuel flow rates and time in modes are inputted for each phase of flight. Once the User-Created Aircraft are fully detailed, the aircraft can be added to an emissions study and used to generate emission output.

References

EDMS, Emissions and Dispersion Modeling System, Software Package, Ver. 5.1.4.1, EUROCONTROL, Brussels, Belgium, 2013.

Environmental Protection Agency (1993). *Evaluation of Air Pollutant Emissions from Subsonic Commercial Jet Aircraft (R-99-013)*. Retrieved from http://www.epa.gov/OMS/regs/nonroad/aviation/r99013.pdf

ICAO (n.d.). *Local Air Quality*. Retrieved from http://www.icao.int/environmental-protection/Pages/local-air-quality.aspx

Katsaduros, D., Prall, M. and Johnson, M.E. (2014). Exploration of Emissions Modeling for a General Aviation Airport. *AIAA SciTech Proceedings*, 52nd Aerospace Sciences Meeting. National Harbor, MD. 13-17 January 2014.

MATLAB, Matrix Laboratory, Software Package, Ver. 2012a, Mathworks, Natick, MA, 2012.