# Appendix - Engine Deck Generation

Since the DLA 180cc engine which is a twin cylinder engine and its performance data, apart from the maximum power and RPM, is unavailable, it was decided to generate its performance data from the data available for its slightly larger twin engine counterpart, i.e., the TOA 288 Figure 1.1 and Figure 1.2 show the power and fuel consumption data available for the TOA 288 in its manual.

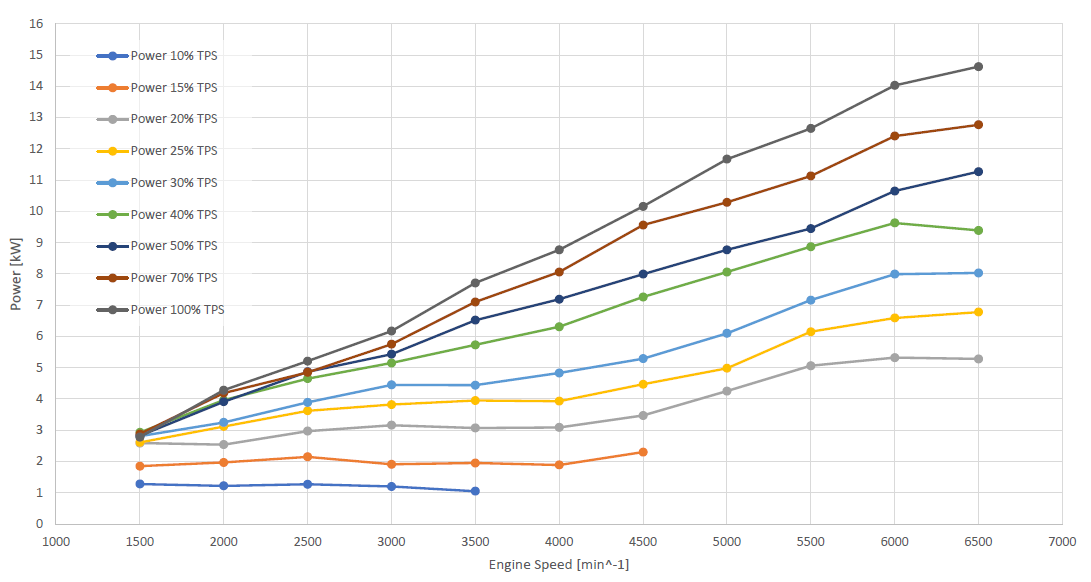


Figure . TOA 288 Shaft Power Data

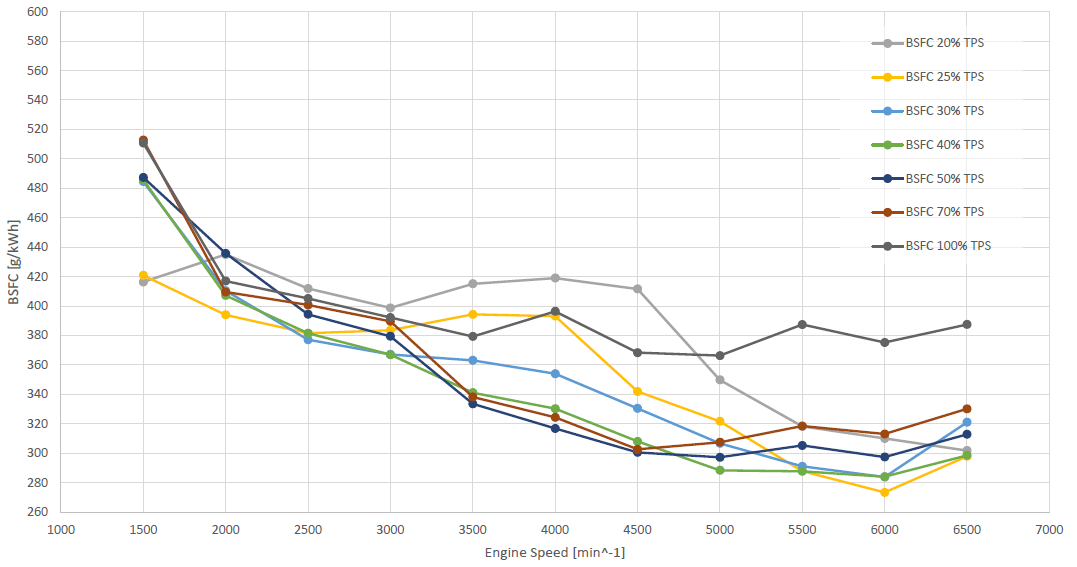


Figure . TOA 288 Brake Specific Fuel Consumption Data

The data in both Figure 1.1 was only given up to 6500 RPM, the tabular data for power was therefore extrapolated to 7000 RPM so as to match the maximum RPM of the DLA 180cc engine.

The extrapolated matrix data was then normalized, such that each column of the extrapolated matrix, which corresponded to different RPMs, was divided by its largest shaft power value i.e., the values in the 100 percent throttle row; a matrice such as the following was obtained:

A scaling row was also generated using the last row of the extrapolated data matrix, the entire power row which corresponded to 100 % throttle was divided by the last value of that row and this normalized vector was multiplied with the maximum rated power of the DLA 180cc, i.e., 18 hp. Once the array was multiplied and a maximum power array for 100% throttle at all RPMs was obtained, this array was then multiplied to the entire normalized power matrix shown above such that each row of the normalized matrix was multiplied by the scaled maximum power array. The result was the power trends of the TOA 288 at different throttles and RPMs scaled to depict those of the DLA 180cc.



Figure . Scaled Engine Shaft Power Data for the DLA 180cc

Since the TOA and DLA are 2-cylinder piston engines, their SFC trends are expected to be similar. Although a power or SFC curve solely for the engine has not been found, an SFC curve with a 32-inch propeller shows an erratic trend with values similar to those of the TOA. Since the SFC data does not follow a clearly discernable increasing or decreasing trend, we could not accurately adjust it for a singular value of the DLA trend, instead we simply extrapolated the TOA trend for the missing RPM with the assumption that since both engines are of the same type and since SFC is a normalized quantity, simply extrapolating the TOA data to match the DLA data would be a reasonably accurate depiction of the actual specific fuel consumption trends.



Figure . Scaled Specific Fuel Consumption Data for the DLA 180cc

### Data Acquisition Based on Calculation Condition (bsfc\_and\_shaft\_power\_from\_engine)

This function is responsible for returning singular RPM, SFC and Shaft Power values against each of the throttle settings of the engine supplied to the function. The function uses one of three available algorithms, each algorithm having a different criterion of selecting singular values for each power setting from its associated arrays in the engine data matrices discussed in section **Error! Reference source not found.**. Before the execution of any algorithm, the one-dimensional arrays of power and SFC, corresponding to the power setting under consideration, are extracted from the two-dimensional arrays of the data from section **Error! Reference source not found.**. The working of each algorithm is discussed as follows:

#### Minimum SFC

This algorithm, as the name suggests, picks the RPM and Shaft Power against the minimum SFC value from the array corresponding to the throttle setting under consideration. In case there are two same minimum SFC values, the lower RPM and is then selected.

#### Minimum SFC to Power Ratio

The algorithm begins by creating a one-dimensional array which consists of the element by element ratios of the aforementioned arrays of SFC and Shaft Power (SFC / Shaft Power); SFC and Shaft Power values are then selected against the RPM which corresponds to the minimum value in the aforementioned array of SFC to Power ratios.

#### Maximum Power

This algorithm is similar to the minimum SFC algorithm in section 1.1.1.1, i.e., the RPM and SFC values corresponding to the maximum Shaft Power value in the Shaft Power array are selected for the throttle setting under consideration.

### Propeller Data Read (prop\_data\_read\_write\_function\_28)

An online catalogue of propeller geometry and performance data is available from APC Propellers website. The performance data file of the 28 x 20 propeller was used to extract the propeller efficiency values at different RPMs and velocities. The propeller efficiency values corresponding to velocities were then used to obtain thrust from the following relation:

The function was responsible for reading data off the data file as well as extrapolating the data to higher RPMs before exporting it to other functions. The engine performance data mentioned in section **Error! Reference source not found.** goes up to 8000 RPM while the propeller data file has performance data up to 7000 RPM i.e. the rest of the data was extrapolated based on the existing data in the file. A sample of the performance data file is shown in Figure 1.5.

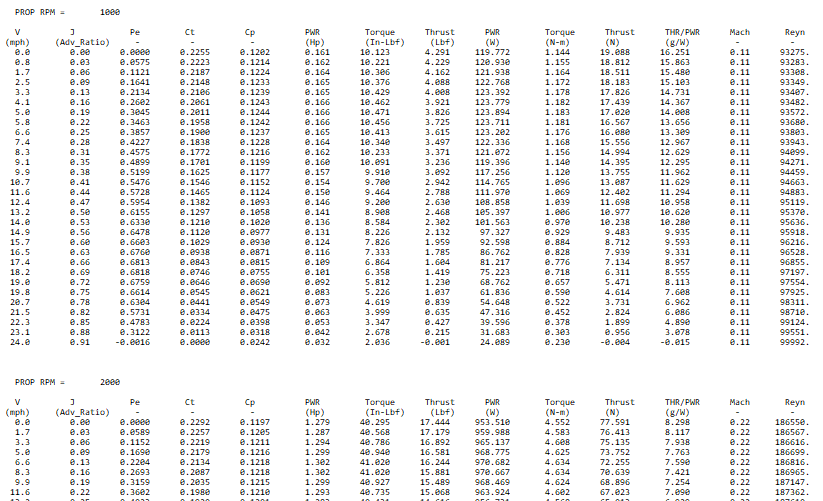


Figure . Sample data from propeller performance data file

The data reading part of the function opens the performance file and initializes the arrays which will be used to store the file data in the function’s output arrays. ‘fopen’ command is used to open the file and ‘fgetl’ command is used to go line by line in the file. A while loop is initiated such that it keeps going through the file until the lines read by fget1 have characters in them. Inside the loop there are checks, which use the ‘if’ statement, in place to avoid reading data from column headers and to skip empty lines. There is also a similar check to differentiate between the data for each RPM since the data is grouped on the basis of RPMs as shown in Figure 1.5. Once the function finishes reading and storing the data for all RPM values, the reading part of the function ends with ‘fclose’ function.

The function then utilizes the above mentioned file data to generate data for missing higher RPMs using the following linear extrapolation formula:

Both Velocity and Propeller Efficiency sets for the higher RPMs are extrapolated using this formula.

### Losses due to Mechanical Inefficiencies and Increasing Altitude (altitude\_variations\_with\_losses)

This function is responsible for introducing the effect of altitude on the engine’s performance as well as introducing the impact of mechanical inefficiencies of the engine to the engine performance data generated by the aforementioned functions. It receives the Shaft Power array generated by the functions discussed in sections **Error! Reference source not found.** and 1.1.1 and reduces the values of the array for the altitude specified in the input using the following relation:

Where:

### Power Delivered by Propeller or Power Extracted (Calculate\_Power\_Extracted)

This function is responsible for generating a final Power Available three dimensional array with dimensions corresponding to the input velocities, altitude and shaft power values after accounting for the propeller efficiency losses at different RPMs and velocities.

The function receives a Shaft Power array which has been accounted for density and mechanical losses by the function discussed in section 1.1.3, it also receive an array of corresponding RPMs and finally a two dimensional velocities array. The user inputs a set of Mach numbers and a set of altitudes; since Mach numbers translate to different velocities at different altitudes, the resulting velocity matrix is understandable two-dimensional. The two dimensions of the velocity array allows the function to set two dimensions of the output power array, the third dimension being the number of corresponding RPMs with the input power array.

The function then uses the propeller data read function discussed in section 1.1.2 to get the propeller efficiency data at different speeds and RPMs. This function plays the vital role of bridging the gap between the disconnect of the propeller data and the engine data. The RPM array of the propeller starts at 1000 RPM and ends at 9000 with a 1000 RPM interval; however, the engine data starts at 3000 RPM and ends at 9000 RPM with a 500 RPM interval. The interpolation in between the individual data sources is made trivial by the ‘interp1’ command; however, the simultaneous use of both data sources to produce a three-dimensional power matrix requires complicated data manipulation.

The function uses a nested loop with three levels, the inner most level corresponds to the RPMs while the outer loops correspond to altitudes and Mach numbers, which were determined using the velocity matrix as discussed earlier in the section. In the inner most nested loop, the function uses ‘if-else’ conditions to check the propeller data for the closest higher and lower values against the current shaft power RPM in the inner most loop. Once the RPMs are located in the propeller data, the function then finds the closest higher and lower velocities in the data corresponding to those RPMs in the propeller data, upon finding the velocities, the function also stores the corresponding propeller efficiency values. The function then uses the shaft power RPM, the current speed in the outer loop and the above mentioned, RPMs, velocities and efficiencies from the propeller data, to linearly interpolate for the required propeller efficiency. In case some value lies at the higher edge of the propeller data, linear extrapolation is used. The found propeller efficiency is then multiplied to the shaft power and placed against the appropriate indexes of the output power matrix.

### Extracted Thrust and Fuel Consumption (extract\_thrust\_and\_fuel\_consumption)

This function uses the function discussed in section 1.1.3 to adjust the engine shaft power data for density and mechanical losses, then multiples the obtained shaft power to the SFC data from the engine data extracted along with the power data using the function discussed in section 1.1.1 in order to get fuel flow values.

The function then uses the function discussed in section 1.1.4 to add propeller efficiency effects to the shaft power data with varying speeds and RPMs. The function finally uses the following formula to give a thrust matrix to a final script:

The values of Propeller power, velocity and propeller efficiency are all appropriately correlated by virtue of the indexing of the aforementioned three-dimensional propeller power matrix. The function gives a three dimensional thrust array and a two dimensional fuel consumption array as outputs, along with the throttle settings and RPM arrays received from other functions.

The engine deck generated by the code above shows the variation of Thrust and Fuel Consumption with altitude, Mach and percent throttle as per the format required for input in mission synthesis software, a sample image of a larger deck is shown as follows:

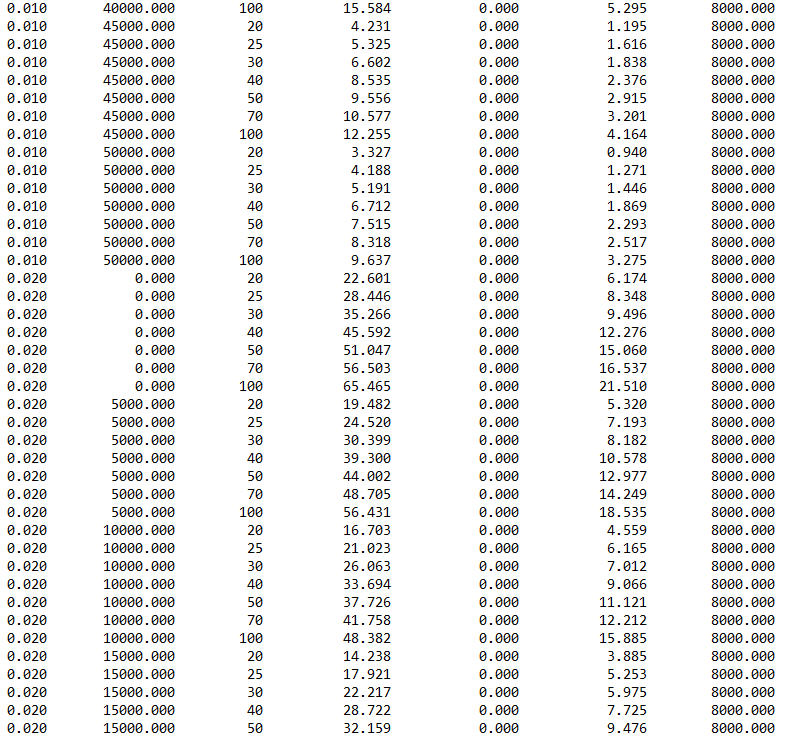


Figure . Engine Deck Output

### MISSION SYNTHESIS SOFTWARE Deck Generation (generate\_mission synthesis software\_deck)

This script takes the output from the function discussed in section 1.1.5 and organises the thrust and fuel consumption values against their corresponding Mach numbers, altitude and throttle settings, as per the format of an engine deck input file for the mission synthesis software Performance Modelling Results

Following are the Thrust and Fuel Flow charts obtained from the aforementioned data drive engine deck generation algorithm:

### Shaft Power Maps













### Fuel Flow Maps













### Thrust Maps

The following thrust maps have been developed using the power plant performance deck generation algorithm which has been outlined in detail in section **Error! Reference source not found.**; it involves the manipulation of the engine data shown in sections 1.1.7 and 1.1.8 alongside the propeller performance data sampled in Figure 1.5 and using an RPM selection criterion such as maximum thrust or minimum SFC to generate the values of thrust at different flight conditions expected in the aircraft’s flight envelope.











