## Performance Modelling of Propulsion Unit

There are two approaches to modelling the performance of any engine, physics based and data based; following is the description of a model developed using the later approach.

### Problem Statement

In the design process of a loitering munition, initial design studies indicated the requirement of an engine with 25 hp maximum power. In addition to selection of the power plant, a model of engine shaft power and fuel flow variation with altitude, speed and throttle setting was also required for generating an engine data file compatible with the mission sizing and synthesis software named ‘Flight Optimization System (FLOPS)’ which was to be used in the preliminary design stage.

### Engine Selection

Selection studies involving pros and cons analyses and scoring methods indicated that the wankel type rotary engine would be the ideal power plant for a loitering munition. Several characteristics of the Wankel engine made it a better candidate as opposed to the reciprocating engines, such as, higher power density, smaller package volume, fewer parts and lower procurement costs.

The design required an engine with 25 hp maximum shaft power. Out of all the candidate engines, the high performance configuration of the Rotapower 150 cc rotary engine by Freedom Motors was selected since its maximum power was exactly 25 hp which implied more fuel efficient cruise operation as compared to other overpowered options.

### Methodology for Performance Modelling

The following flow chart shows all of the components involved and their interaction in the performance modelling algorithm.

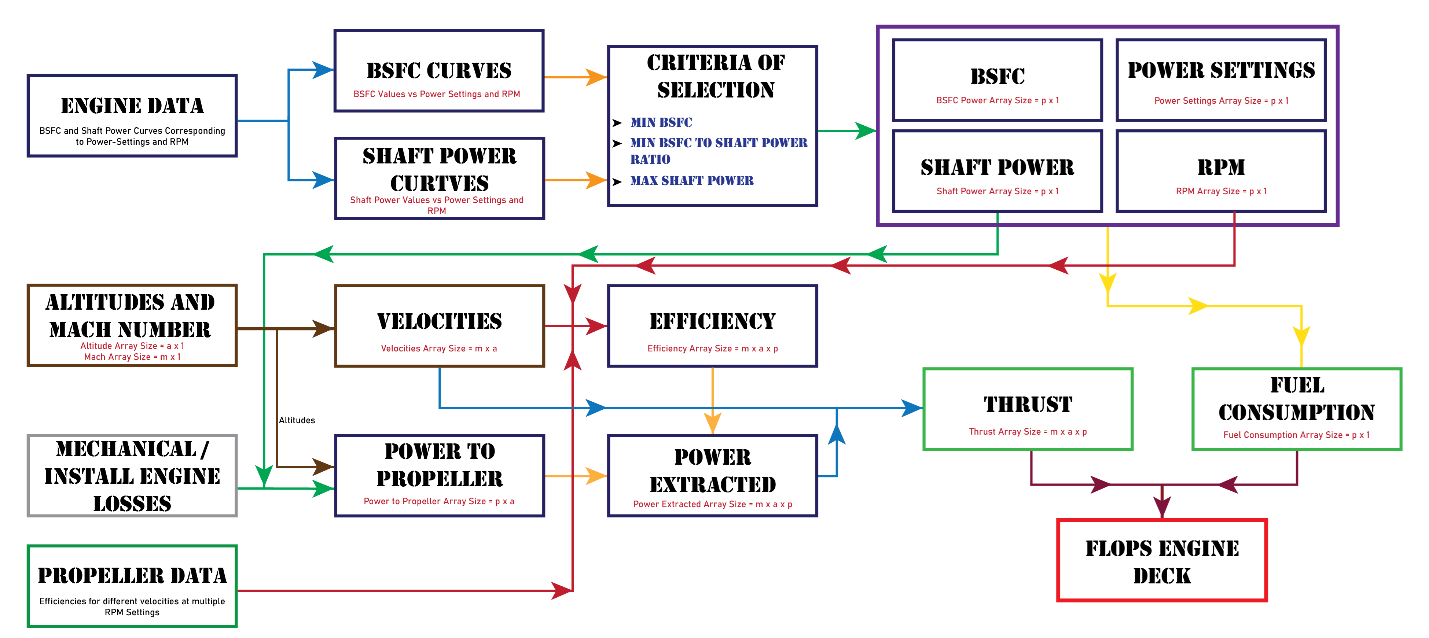


Figure 1.7 Performance Modelling Algorithm Flow Chart {Flow Chart Requires Changes / Replacement}

Following subsections will explain the major components or code blocks and the last subsection will explain how they interact.

### Engine Data (engine\_data\_rotapower\_150cc)

The data available for the selected engine was scarce, such that only the maximum throttle power curve against the RPM was available, that too for the base configuration, whereas the selection was that of the high performance configuration of the Rotapower engine. Only a single value of specific fuel consumption (SFC) was given for the base configuration against a rated shaft power.

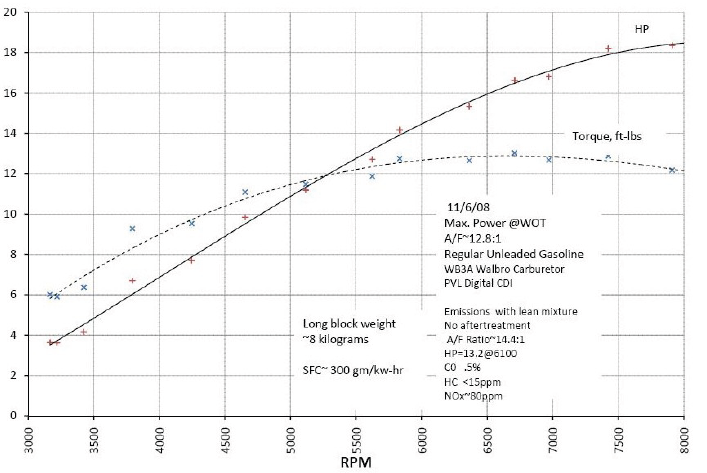


Figure 1.8 Performance data of the base configuration of the Rotapower engine [1]

Data available for the TOA 288 2-stroke piston engine was plentiful such that the variations of shaft horsepower and specific fuel consumption against RPM at different throttle settings were available.

Since both engines were roughly in the same power bracket and both were purposely built for UAVs, it was decided to use the better rounded data of TOA 288 in conjunction with some data manipulation to obtain a performance model for the Rotapower engine. The first step was to generate the sea-level data for the Rotapower which would show the variation of shaft power and specific fuel consumption with RPM at different throttle settings, similar to what is shown in Figure 1.9 and Figure 1.10.

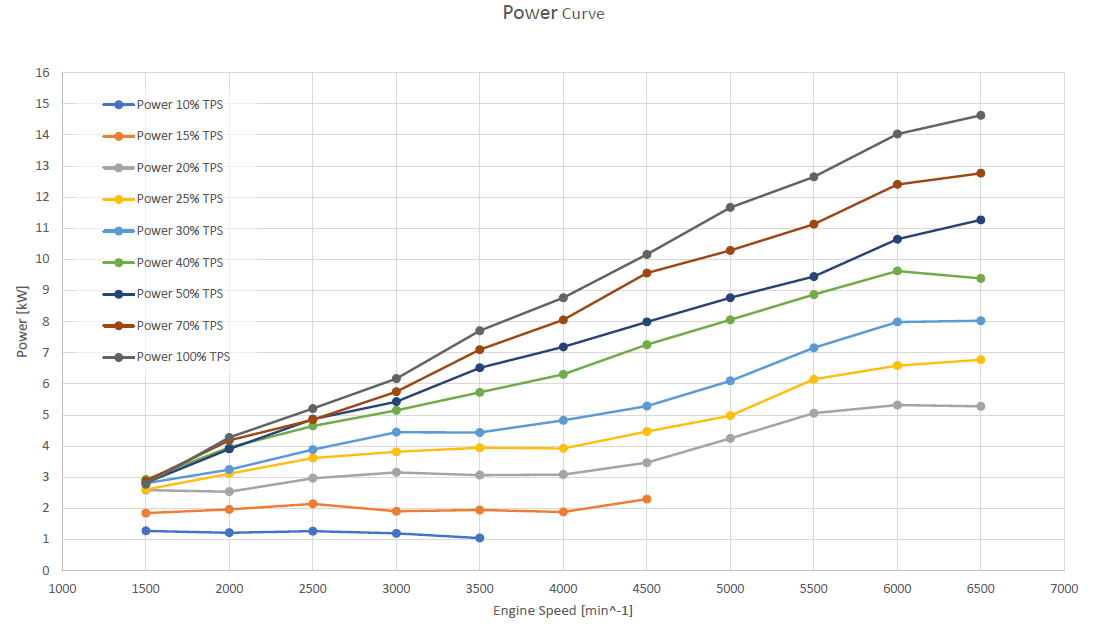


Figure 1.9 TOA 288 power curves at different RPMs and throttle settings [2]

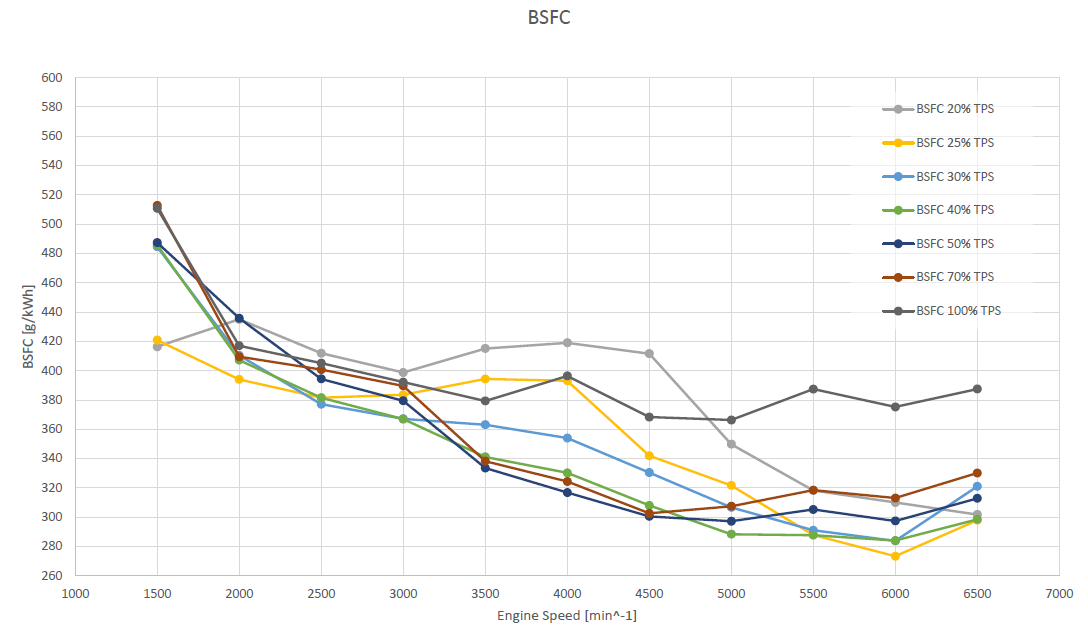


Figure 1.10 TOA 288 BSFC curves at different RPMs and throttle settings [2]

First of all, the shaft power data of the base configuration Rotapower engine from Figure 1.8 was adjusted to mimic the performance of the high performance configuration. The Rotapower webpage {cite} showed that the high performance configuration of the selected engine would give its maximum 25 hp shaft power at 9000 RPM, the data from Figure 1.8 was therefore extrapolated to 9000 RPM and all values in the set were increased by 30% to match the maximum power of 25 hp at 9000 RPM. The cubic method of extrapolation was selected since it was the only method which did not give abnormal values for RPMs below 3000.

To get shaft power data at partial throttle settings, the data in Figure 1.9 was translated into tabular or matrix form. The data in both Figure 1.9 and Figure 1.10 was only given up to 6500 RPM, the tabular data was therefore extrapolated to 9000 RPM so as to match the data of the Rotapower 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; the following matrix was obtained:

Each row of the normalized matrix was then finally multiplied with the modified shaft power array of the Rotapower engine thereby mimicking its performance at partial throttle settings. The shaft power data obtained therein has been visualized in the following figure:



Figure 1.11 Shaft power data for the Rotapower engine post data manipulation

The approach was slightly different for obtaining the specific Fuel Consumption (SFC) array owing to the lack of data. There was only a single SFC value of the base configuration of the Rotapower engine against a rated power of 15 hp; this SFC was multiplied by the quotient of 25 and 15 hp to get an estimated SFC for the high performance configuration of the engine. The SFC data from Figure 1.10 was extrapolated to 9000 RPM and a normalized scaling array was obtained from this data by normalizing the numbers corresponding to 100 percent throttle by dividing them by the value in this set which corresponded to 9000 RPM. The normalized scaling array was multiplied by the aforementioned SFC value obtained for the high performance configuration of the Rotapower engine. In similar fashion to the extrapolated power matrix, the extrapolated SFC matrix was normalized by dividing all columns by their last value and then multiplying each row with the scaling array. The SFC data obtained therein has been visualized in the following figure:



Figure 1.12 Specific Fuel Consumption data for the Rotapower engine post data manipulation

From this function, the outputs are the sets of the considered throttle settings and RPMs, as well as the two dimensional matrices of shaft power and specific fuel consumption at these throttle settings and RPMs. The shaft power and SFC matrices are representative of the engine’s performance at sea-level.

### 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 1.3.4. 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 1.3.4. 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 1.3.5.1, 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 1.3.4 goes up to 9000 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 1.13.

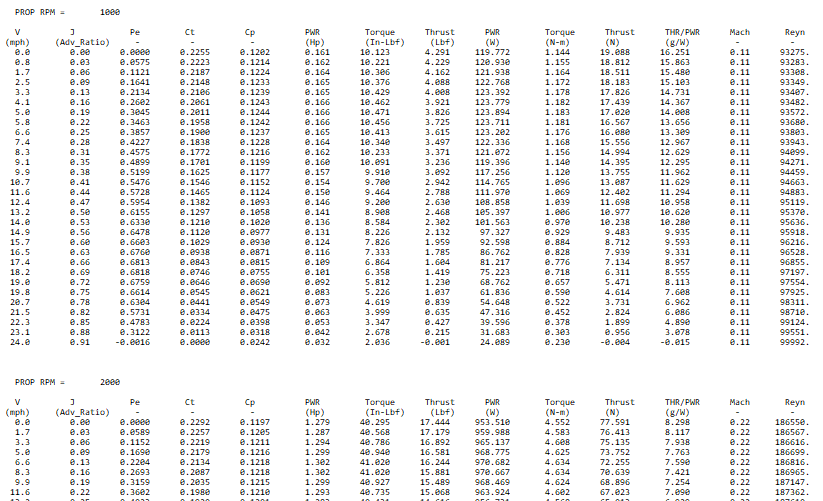


Figure 1.13 Sample data from propeller performance data file.

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 1.13. 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 1.3.4 and 1.3.5 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 1.3.7, 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 1.3.6 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 1.3.7 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 1.3.5 in order to get fuel flow values.

The function then uses the function discussed in section 1.3.8 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.

### FLOPS Deck Generation (generate\_flops\_deck)

This script takes the output from the function discussed in section 1.3.9 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 ‘Flight Optimization System (FLOPS)’ software.