

1. AR-731 Wankel Engine Performance Deck Generation

1.1 Available Data

The AR-731 wankel engine has been purpose built for unmanned UAV applications and is ideal for short lifespan aircraft such as the Loitering Munition (LM) being currently designed. Following are its technical characteristic's which were obtained from a spec sheet made available online by the manufacturer UAV Engines Ltd.

Engine Type	Single Rotor Wankel
Capacity	208 cc chamber size
Power Output	38 bhp at 7800 rpm
Weight	21.7 lbs (9.9 kg)
Specific Fuel Consumption	0.57 lb/hp/hr at max power, 0.52 lb/hp/hr at cruise power,
Ignition System	Electronic Contactless Magneto
Fuel Type	Avgas 100 LL or Mogas Regular Grade

Apart from the general characteristics, Full (Wide Open) Throttle, ground performance data was also provided in the spec sheet as performance chart; the data was later digitized in MATLAB using PlotDigitizer software available online. The original chart and its MATLAB version are shown in Figure 1.1 and Figure 1.2.

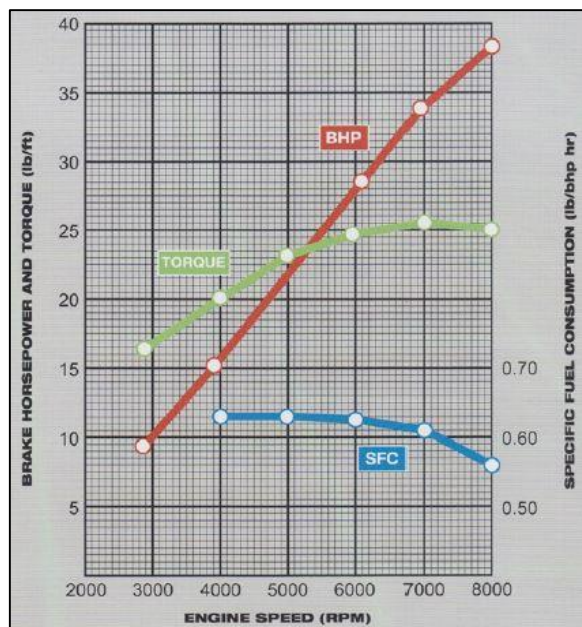


Figure 1.1 Spec Sheet version

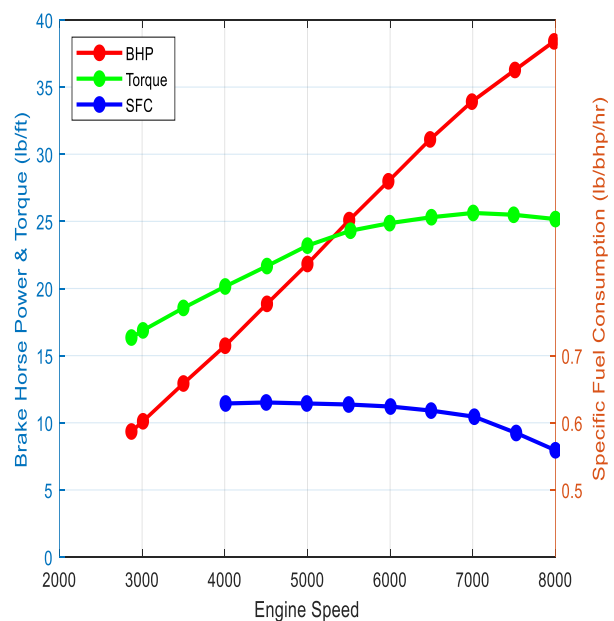


Figure 1.2 MATLAB digitized version

1.2 Methodology for Performance Modelling

The engine deck includes the variation of Power and Fuel consumption with altitude, Mach number and throttle percentage. The power supplied by the engine to the shaft is called shaft power and the power available to the aircraft comes from the propeller, this means that power available is dependent upon the propeller's performance as well and the propeller efficiency and thrust are a function of propeller geometry, altitude, speed and RPM.

Figure 1.3 shows all of the components involved and their interaction in the performance modelling algorithm.

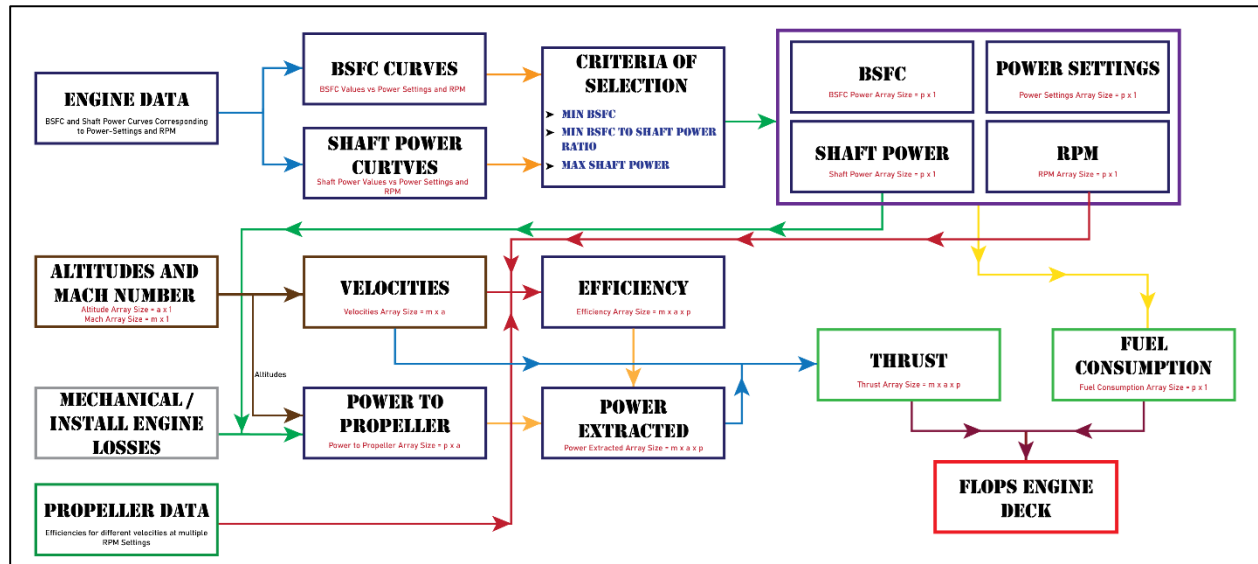


Figure 1.3 Performance Modelling Algorithm Flow Chart

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

1.2.1 Engine Data (AR_731_partial_and_full_throttle_ground_performance)

The data available for the selected engine was at sea level and full throttle as shown in Figure 1.2. 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 partial throttle settings were available. It was decided to use data of TOA 288 in conjunction with some data manipulation to obtain a performance model for the AR-731 engine's partial throttle performance.

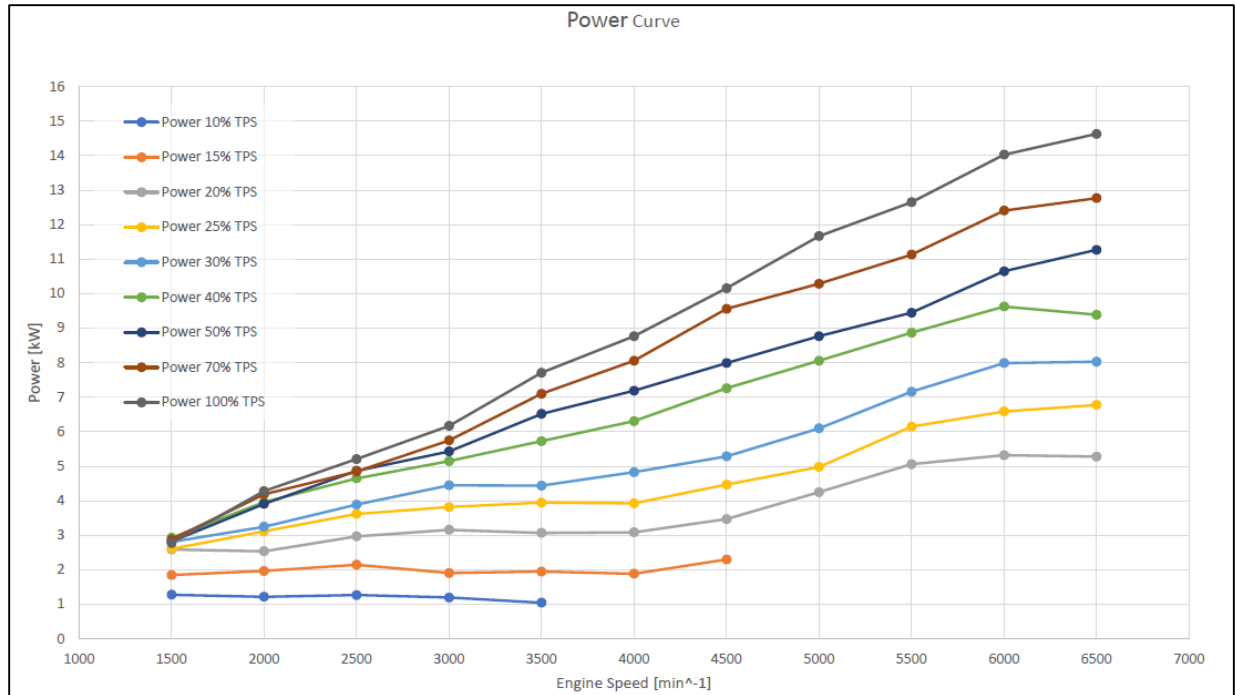


Figure 1.4 TOA 288 power curves at different RPMs and throttle settings

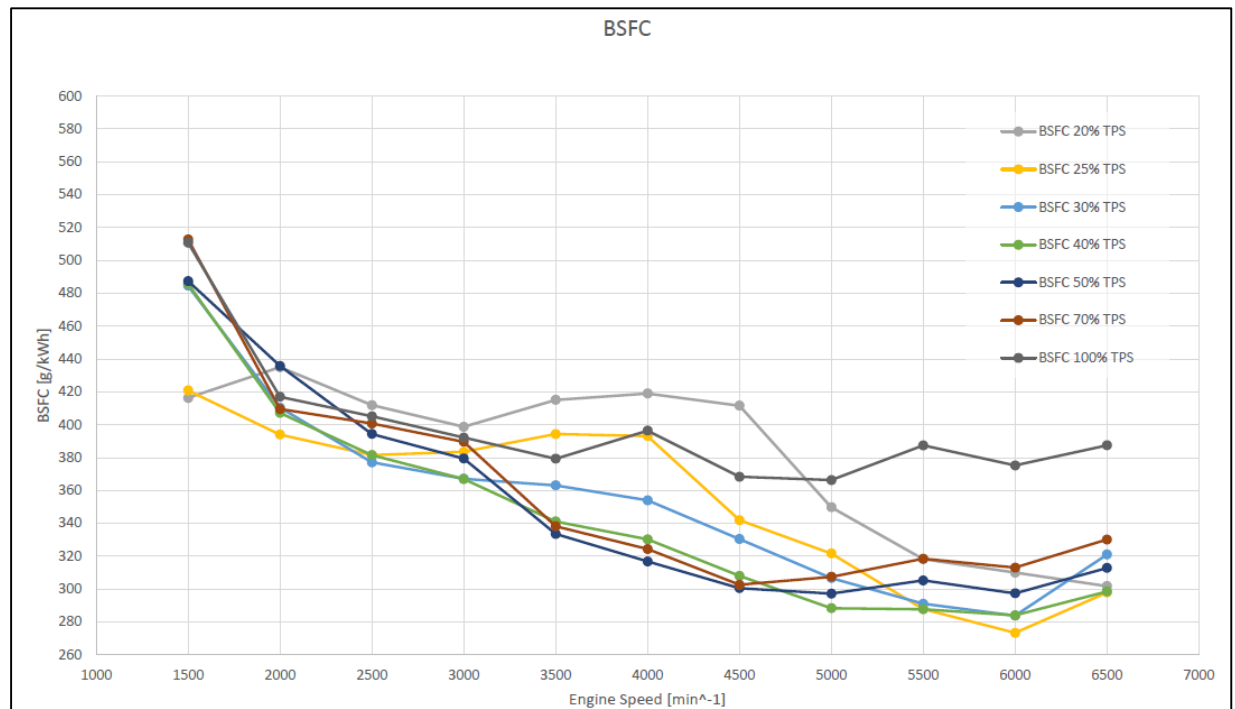


Figure 1.5 TOA 288 BSFC curves at different RPMs and throttle settings

All the curves of Figure 1.4 and Figure 1.5 were digitized as MATLAB arrays. A new overall RPM array (2000:500:8000) was defined and the `interp1()` command was used to generate new power and SFC arrays which would correspond to the overall RPM array.

The data in both Figure 1.4 and Figure 1.5 was only given up to 6500 RPM, the tabular data was therefore extrapolated to 8000 RPM so as to match the data of the AR-731 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; two matrices such as the following were obtained:

0.484	0.397	0.364	0.343	0.368	0.402	0.386	0.374	0.364	0.354	0.345	0.337	0.330
0.629	0.513	0.449	0.441	0.427	0.488	0.479	0.466	0.455	0.444	0.435	0.426	0.418
0.710	0.577	0.545	0.520	0.521	0.567	0.575	0.565	0.555	0.547	0.539	0.531	0.525
0.823	0.744	0.716	0.706	0.692	0.701	0.693	0.694	0.695	0.696	0.696	0.697	0.698
0.855	0.833	0.818	0.784	0.752	0.740	0.764	0.769	0.773	0.776	0.780	0.783	0.786
0.919	0.923	0.920	0.931	0.880	0.882	0.893	0.884	0.877	0.870	0.863	0.857	0.852
1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Each row of the normalized matrices for power and SFC was then finally multiplied with the modified full throttle shaft power and SFC arrays of the AR-731 engines thereby mimicking its performance at partial throttle settings. The shaft power and SFC data obtained therein has been visualized in the following figures:

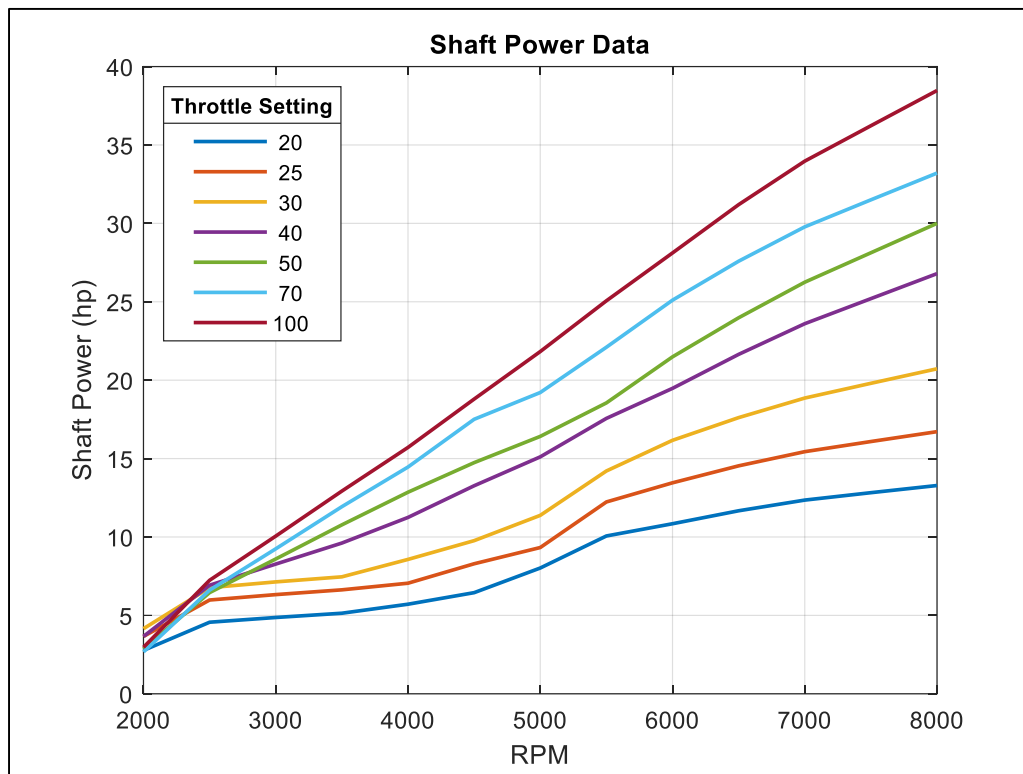


Figure 1.6 Shaft power data for the AR-731engine post data manipulation

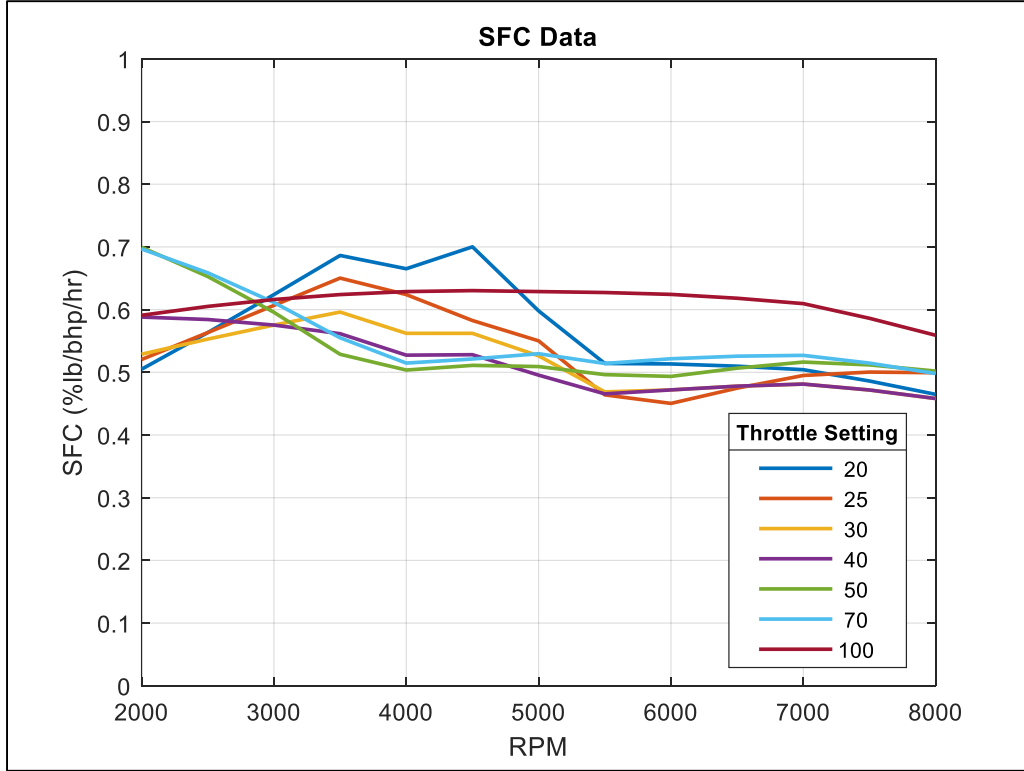


Figure 1.7 Specific Fuel Consumption data for the AR-731 engine post data manipulation

It is clear that the TOA-288 data for SFC is not suitable to mimic the partial throttle SFC characteristics of the AR-731. The power curves in figure { } show that the data driven method is not at fault since the partial throttle curves look reasonable and realistic, a solution to the erratic SFC trends would be to keep the same method but change the data from the 2-stroke piston engine to an actual wankel engine which would better represent the partial throttle SFC trends of the AR-731. Until the procurement of better data, the engine deck will be generated using the available TOA-288 data to ensure timely generation and maturity of the framework surrounding data driven engine deck generation.

1.2.2 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 1.2.1. 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 1.2.1. The working of each algorithm is discussed as follows:

1.2.2.1 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.

1.2.2.2 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.

1.2.2.3 Maximum Power

This algorithm is similar to the minimum SFC algorithm in section 1.2.2.1, 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.

1.2.3 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:

$$T = \frac{\eta_p P}{V}$$

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 1.2.1 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 1.8.

PROP RPM = 1000													
V (mph)	J (Adv_Ratio)	Pe -	Ct -	Cp -	PMR (Hp)	Torque (In-Lbf)	Thrust (Lbf)	PMR (W)	Torque (N-m)	Thrust (N)	THR/PMR (g/W)	Mach -	Reyn -
0.0	0.00	0.0000	0.2255	0.1202	0.161	10.123	4.291	119.772	1.144	19.088	16.251	0.11	93275.
0.8	0.03	0.0575	0.2223	0.1214	0.162	10.221	4.229	120.930	1.155	18.812	15.863	0.11	93283.
1.7	0.06	0.1121	0.2187	0.1224	0.164	10.306	4.162	121.938	1.164	18.511	15.480	0.11	93388.
2.5	0.09	0.1641	0.2148	0.1233	0.165	10.376	4.088	122.768	1.172	18.183	15.103	0.11	93349.
3.3	0.13	0.2134	0.2106	0.1239	0.165	10.429	4.008	123.392	1.178	17.826	14.731	0.11	93487.
4.1	0.16	0.2602	0.2061	0.1243	0.166	10.462	3.921	123.779	1.182	17.439	14.367	0.11	93482.
5.0	0.19	0.3045	0.2011	0.1244	0.166	10.471	3.826	123.894	1.183	17.020	14.008	0.11	93572.
5.8	0.22	0.3463	0.1958	0.1242	0.166	10.456	3.725	123.711	1.181	16.567	13.656	0.11	93680.
6.6	0.25	0.3857	0.1900	0.1237	0.165	10.413	3.615	123.202	1.176	16.080	13.309	0.11	93803.
7.4	0.28	0.4227	0.1838	0.1228	0.164	10.340	3.497	122.336	1.168	15.556	12.967	0.11	93943.
8.3	0.31	0.4575	0.1772	0.1216	0.162	10.233	3.371	121.072	1.156	14.994	12.629	0.11	94099.
9.1	0.35	0.4899	0.1701	0.1199	0.160	10.091	3.236	119.396	1.140	14.395	12.295	0.11	94271.
9.9	0.38	0.5199	0.1625	0.1177	0.157	9.910	3.092	117.256	1.120	13.755	11.962	0.11	94459.
10.7	0.41	0.5476	0.1546	0.1152	0.154	9.700	2.942	114.765	1.096	13.087	11.629	0.11	94663.
11.6	0.44	0.5728	0.1465	0.1124	0.150	9.464	2.788	111.970	1.069	12.402	11.294	0.11	94883.
12.4	0.47	0.5954	0.1382	0.1093	0.146	9.200	2.630	108.858	1.039	11.698	10.958	0.11	95119.
13.2	0.50	0.6155	0.1297	0.1058	0.141	8.908	2.468	105.397	1.006	10.977	10.620	0.11	95370.
14.0	0.53	0.6330	0.1210	0.1020	0.136	8.584	2.302	101.563	0.970	10.238	10.280	0.11	95636.
14.9	0.56	0.6478	0.1120	0.0977	0.131	8.226	2.132	97.327	0.929	9.483	9.935	0.11	95918.
15.7	0.60	0.6603	0.1029	0.0930	0.124	7.826	1.959	92.598	0.884	8.712	9.593	0.11	96216.
16.5	0.63	0.6760	0.0938	0.0871	0.116	7.333	1.785	86.762	0.828	7.939	9.331	0.11	96528.
17.4	0.66	0.6813	0.0843	0.0815	0.109	6.864	1.604	81.217	0.776	7.134	8.957	0.11	96855.
18.2	0.69	0.6818	0.0746	0.0755	0.101	6.358	1.419	75.223	0.718	6.311	8.555	0.11	97197.
19.0	0.72	0.6759	0.0646	0.0690	0.092	5.812	1.230	68.762	0.657	5.471	8.113	0.11	97554.
19.8	0.75	0.6614	0.0545	0.0621	0.083	5.226	1.037	61.836	0.590	4.614	7.608	0.11	97925.
20.7	0.78	0.6304	0.0441	0.0549	0.073	4.619	0.839	54.648	0.522	3.731	6.962	0.11	98311.
21.5	0.82	0.5731	0.0334	0.0475	0.063	3.999	0.635	47.316	0.452	2.824	6.086	0.11	98710.
22.3	0.85	0.4783	0.0224	0.0398	0.053	3.347	0.427	39.596	0.378	1.899	4.890	0.11	99124.
23.1	0.88	0.3122	0.0113	0.0318	0.042	2.678	0.215	31.683	0.303	0.956	3.078	0.11	99551.
24.0	0.91	-0.0016	0.0000	0.0242	0.032	2.036	-0.001	24.089	0.230	-0.004	-0.015	0.11	99992.
PROP RPM = 2000													
V (mph)	J (Adv_Ratio)	Pe -	Ct -	Cp -	PMR (Hp)	Torque (In-Lbf)	Thrust (Lbf)	PMR (W)	Torque (N-m)	Thrust (N)	THR/PMR (g/W)	Mach -	Reyn -
0.0	0.00	0.0000	0.2292	0.1197	1.279	40.295	17.444	953.510	4.552	77.591	8.298	0.22	186550.
1.7	0.03	0.0589	0.2257	0.1205	1.287	40.568	17.179	959.988	4.583	76.413	8.117	0.22	186567.
3.3	0.06	0.1152	0.2219	0.1211	1.294	40.786	16.892	965.137	4.608	75.135	7.938	0.22	186616.
5.0	0.09	0.1690	0.2179	0.1216	1.299	40.940	16.581	968.775	4.625	73.752	7.763	0.22	186699.
6.6	0.13	0.2204	0.2134	0.1218	1.302	41.020	16.244	970.682	4.634	72.255	7.590	0.22	186816.
8.3	0.16	0.2693	0.2087	0.1218	1.302	41.020	15.881	970.667	4.634	70.639	7.421	0.22	186905.
9.9	0.19	0.3159	0.2035	0.1215	1.299	40.927	15.489	968.469	4.624	68.896	7.254	0.22	187147.
11.6	0.22	0.3602	0.1980	0.1210	1.293	40.735	15.068	963.924	4.602	67.023	7.090	0.22	187362.
13.2	0.25	0.4033	0.1928	0.1204	1.284	40.438	14.625	956.954	4.568	65.043	6.930	0.22	187546.

Figure 1.8 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 fgetl 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 1.8. 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:

$$y = y_1 + \frac{(x - x_1)(y_2 - y_1)}{x_2 - x_1}$$

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

1.2.4 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 1.2.1 and 1.2.2 and reduces the values of the array for the altitude specified in the input using the following relation:

$$Power\ to\ propeller = (Shaft\ Power) \times \sigma \times \eta_m$$

Where:

$$\eta_m = mechanical\ efficiency\ (0.8)$$

$$\sigma = \frac{density\ at\ input\ altitude}{density\ at\ sea\ level}$$

1.2.5 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 1.2.4, 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 1.2.3 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.

1.2.6 Extracted Thrust and Fuel Consumption (extract_thrust_and_fuel_consumption)

This function uses the function discussed in section 1.2.4 to adjust the engine shaft power data for density and mechanical losses, then multiplies the obtained shaft power to the SFC data from the engine data extracted along with the power data using the function discussed in section 1.2.2 in order to get fuel flow values.

The function then uses the function discussed in section 1.2.5 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:

$$T = \frac{\eta_p P}{V}$$

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 FLOPS, a sample image of a larger deck is shown as follows:

0.010	40000.000	100	15.584	0.000	5.295	8000.000
0.010	45000.000	20	4.231	0.000	1.195	8000.000
0.010	45000.000	25	5.325	0.000	1.616	8000.000
0.010	45000.000	30	6.602	0.000	1.838	8000.000
0.010	45000.000	40	8.535	0.000	2.376	8000.000
0.010	45000.000	50	9.556	0.000	2.915	8000.000
0.010	45000.000	70	10.577	0.000	3.201	8000.000
0.010	45000.000	100	12.255	0.000	4.164	8000.000
0.010	50000.000	20	3.327	0.000	0.940	8000.000
0.010	50000.000	25	4.188	0.000	1.271	8000.000
0.010	50000.000	30	5.191	0.000	1.446	8000.000
0.010	50000.000	40	6.712	0.000	1.869	8000.000
0.010	50000.000	50	7.515	0.000	2.293	8000.000
0.010	50000.000	70	8.318	0.000	2.517	8000.000
0.010	50000.000	100	9.637	0.000	3.275	8000.000
0.020	0.000	20	22.601	0.000	6.174	8000.000
0.020	0.000	25	28.446	0.000	8.348	8000.000
0.020	0.000	30	35.266	0.000	9.496	8000.000
0.020	0.000	40	45.592	0.000	12.276	8000.000
0.020	0.000	50	51.047	0.000	15.060	8000.000
0.020	0.000	70	56.503	0.000	16.537	8000.000
0.020	0.000	100	65.465	0.000	21.510	8000.000
0.020	5000.000	20	19.482	0.000	5.320	8000.000
0.020	5000.000	25	24.520	0.000	7.193	8000.000
0.020	5000.000	30	30.399	0.000	8.182	8000.000
0.020	5000.000	40	39.300	0.000	10.578	8000.000
0.020	5000.000	50	44.002	0.000	12.977	8000.000
0.020	5000.000	70	48.705	0.000	14.249	8000.000
0.020	5000.000	100	56.431	0.000	18.535	8000.000
0.020	10000.000	20	16.703	0.000	4.559	8000.000
0.020	10000.000	25	21.023	0.000	6.165	8000.000
0.020	10000.000	30	26.063	0.000	7.012	8000.000
0.020	10000.000	40	33.694	0.000	9.066	8000.000
0.020	10000.000	50	37.726	0.000	11.121	8000.000
0.020	10000.000	70	41.758	0.000	12.212	8000.000
0.020	10000.000	100	48.382	0.000	15.885	8000.000
0.020	15000.000	20	14.238	0.000	3.885	8000.000
0.020	15000.000	25	17.921	0.000	5.253	8000.000
0.020	15000.000	30	22.217	0.000	5.975	8000.000
0.020	15000.000	40	28.722	0.000	7.725	8000.000
0.020	15000.000	50	32.159	0.000	9.476	8000.000

Figure 1.9 Engine Deck Output

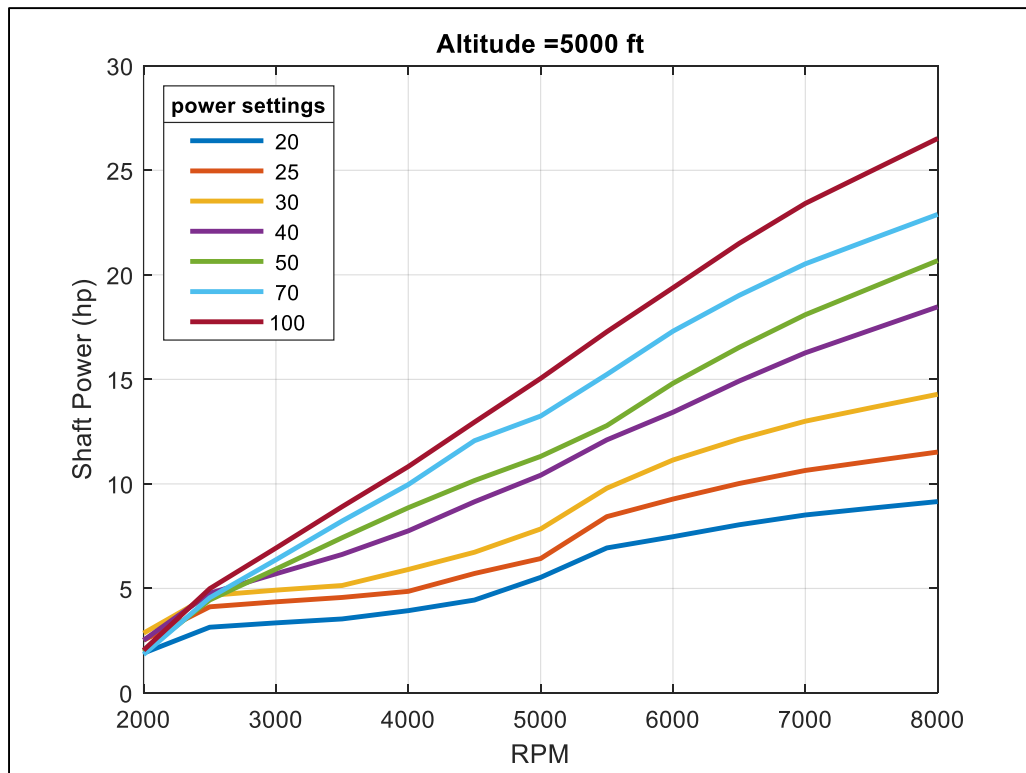
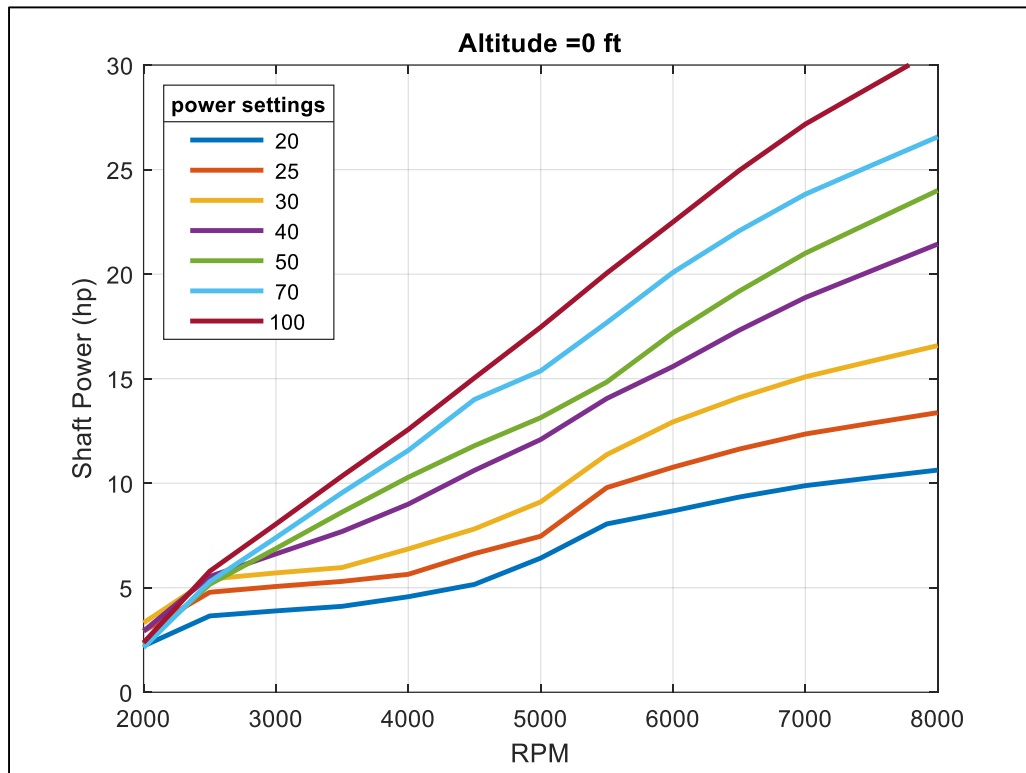
1.2.7 FLOPS Deck Generation (generate_flops_deck)

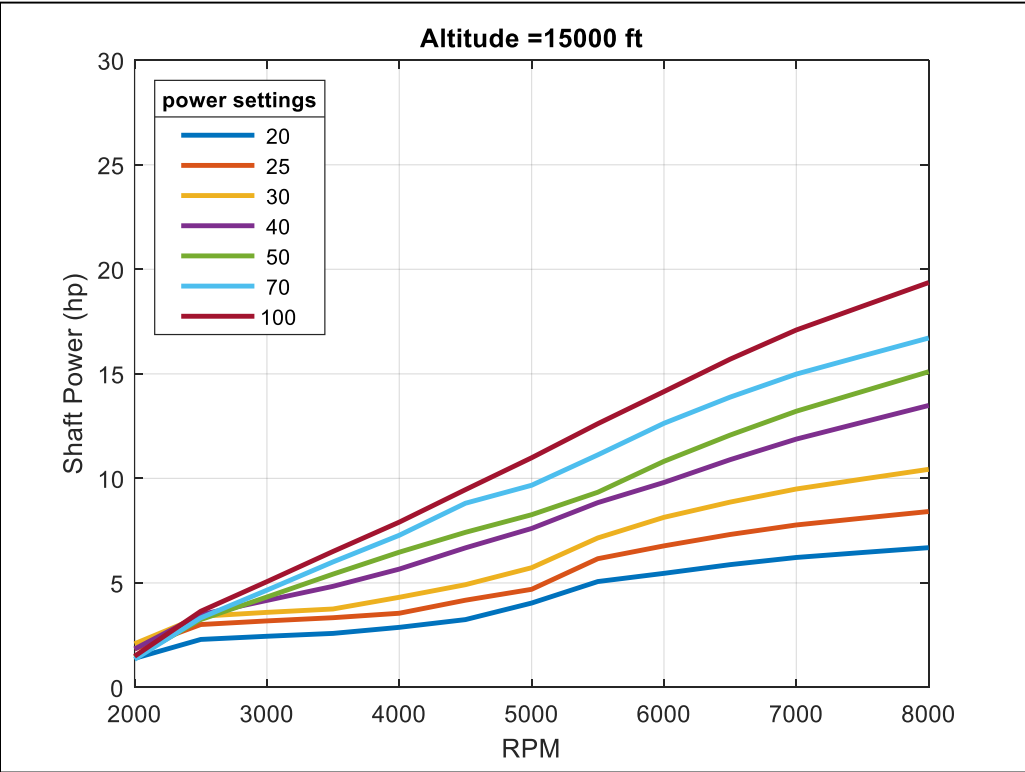
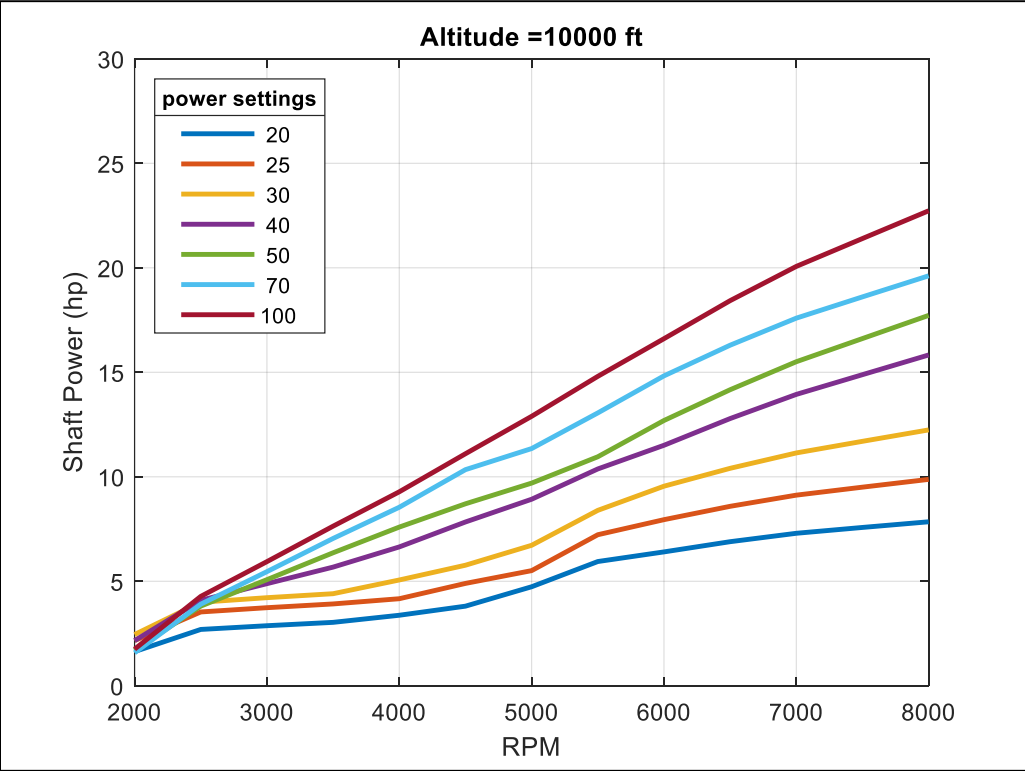
This script takes the output from the function discussed in section 1.2.6 and organises the thrust and fuel consumption values against their corresponding Mach numbers, altitude and throttle settings, as per the format of an engine deck input file for the ‘Flight Optimization System (FLOPS)’ software.

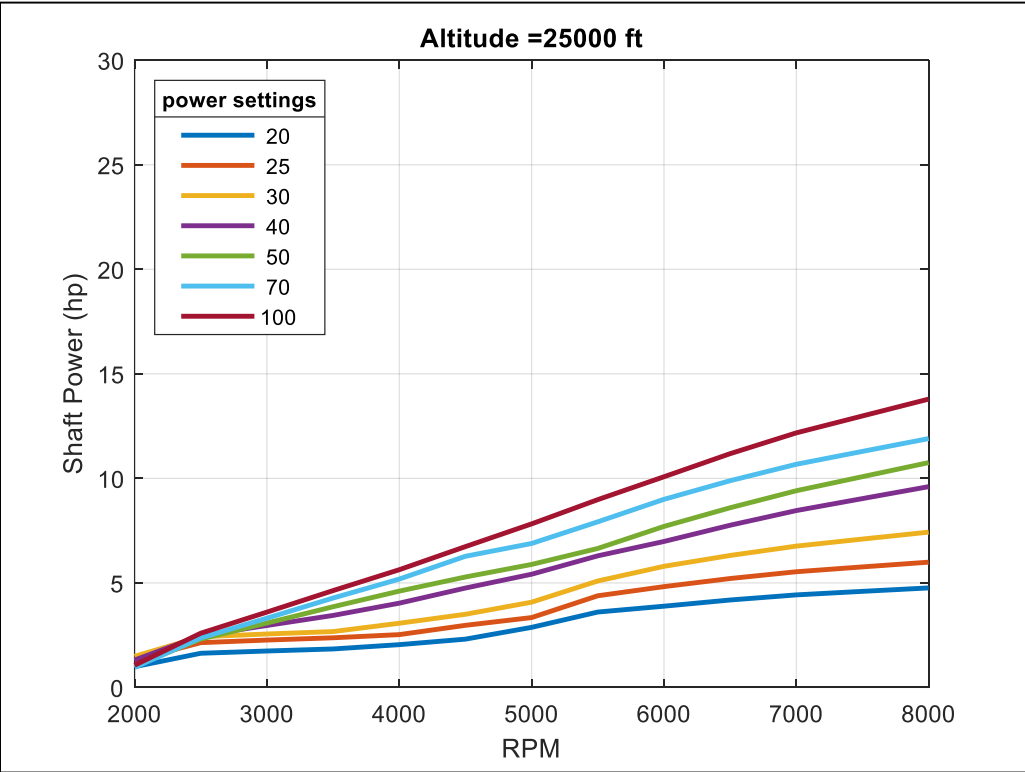
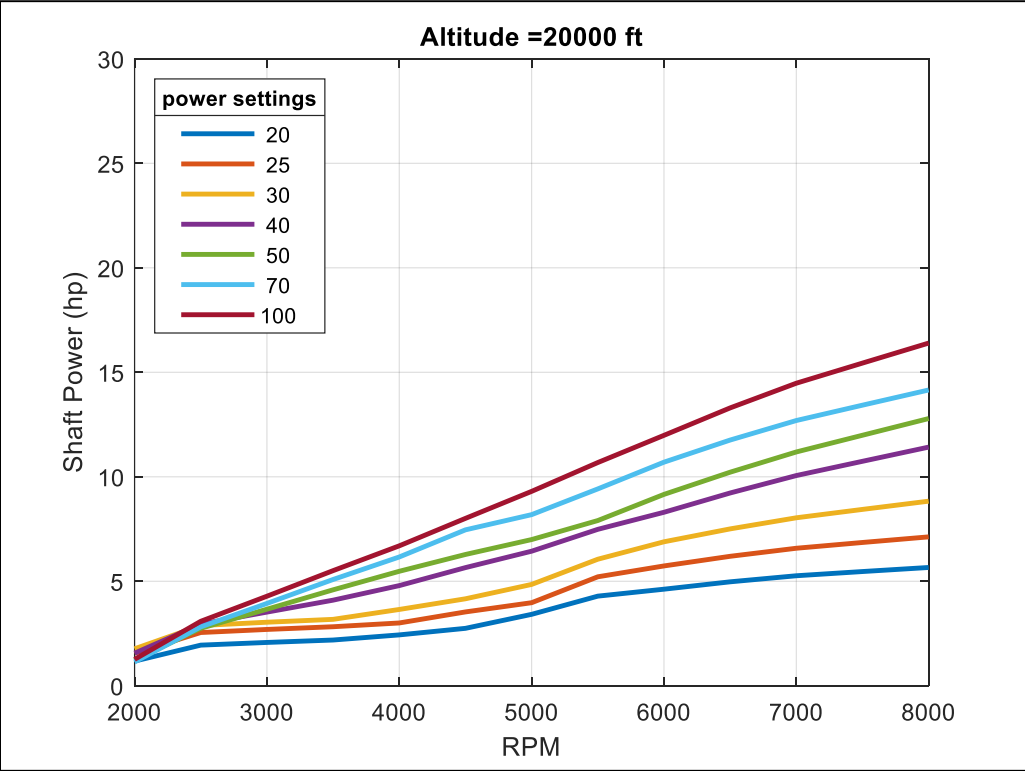
1.3 Performance Modelling Results

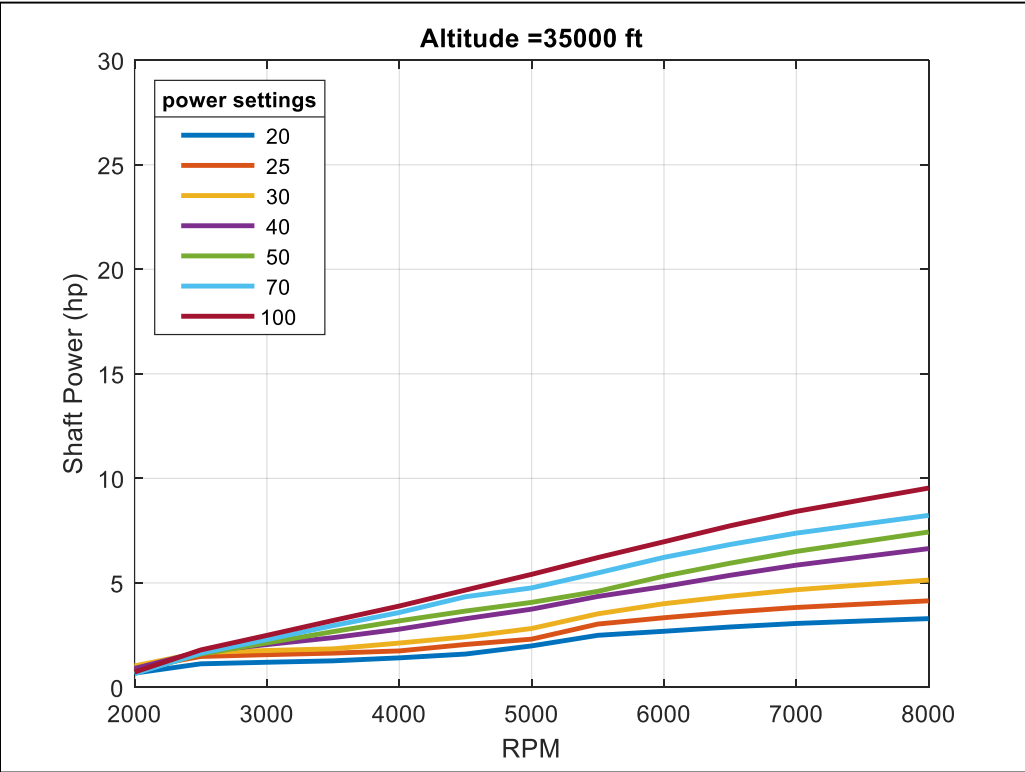
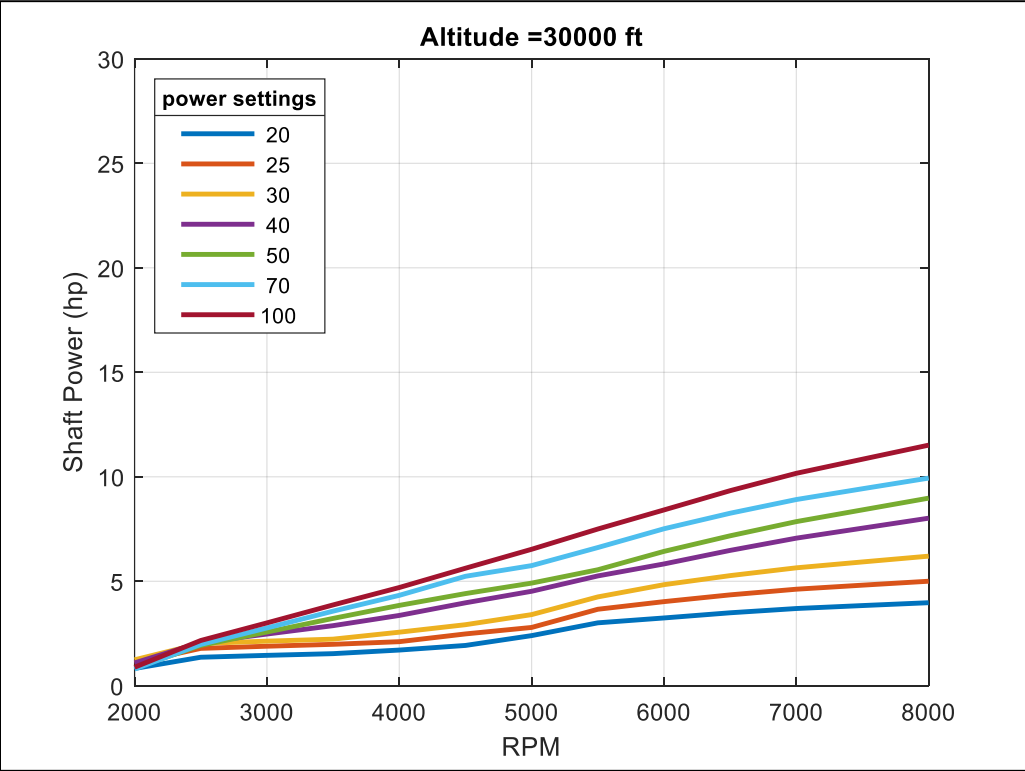
Following are the Thrust and Fuel Flow charts obtained from the aforementioned data drive engine deck generation algorithm:

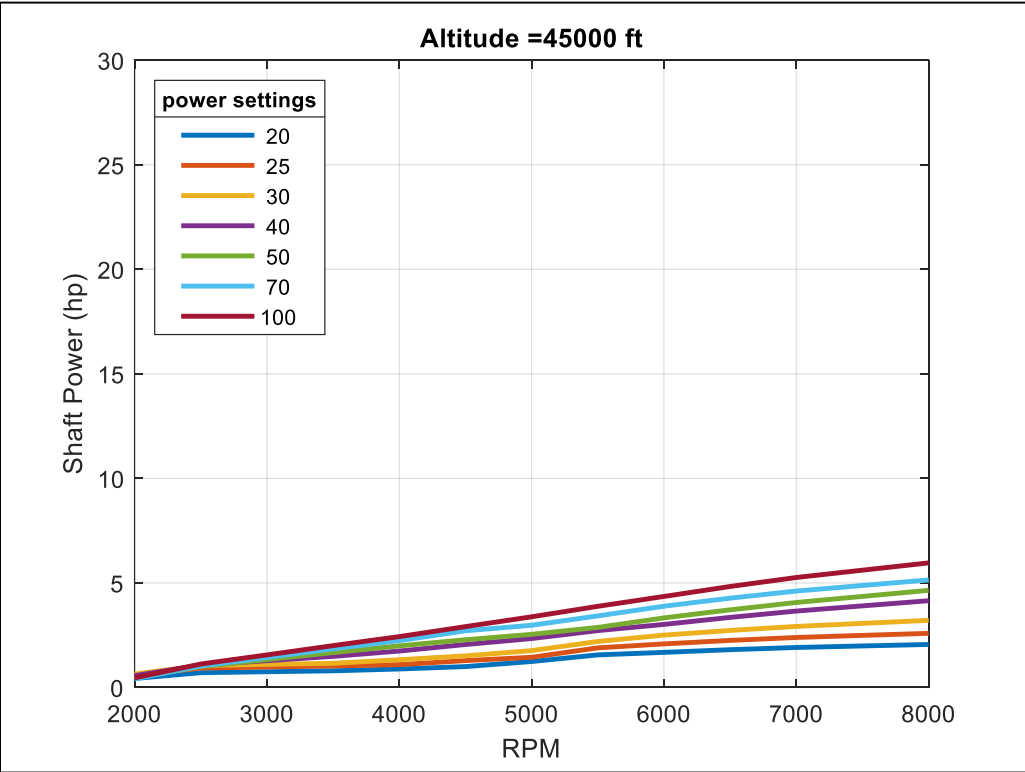
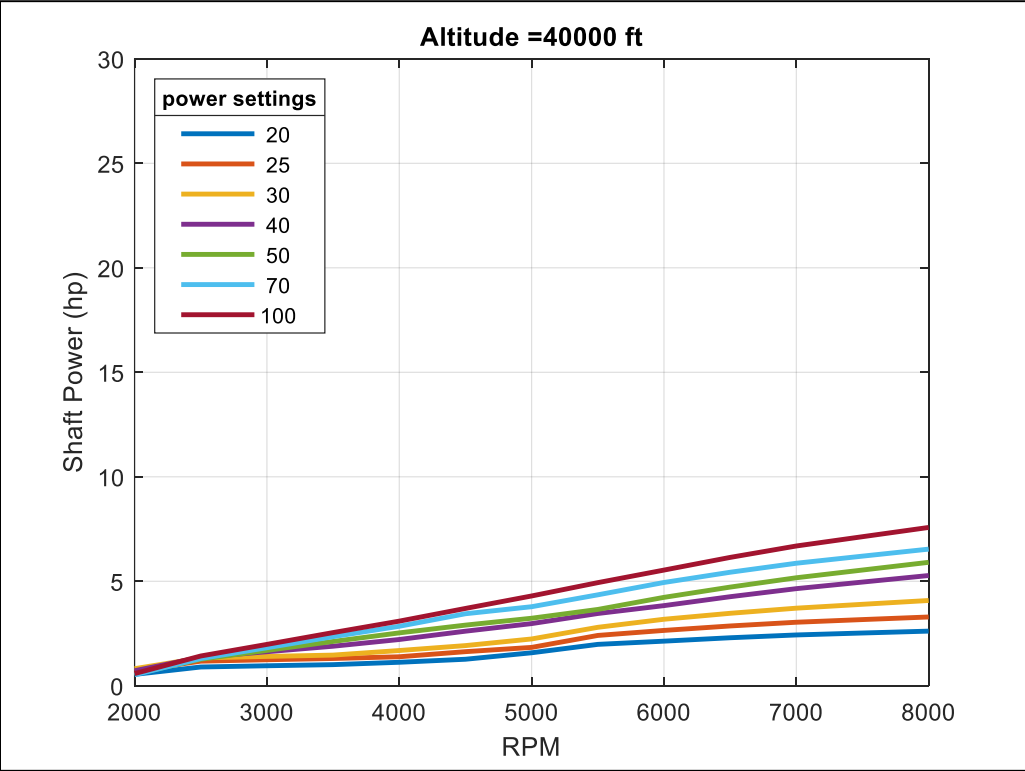
1.3.1 Shaft Power Maps

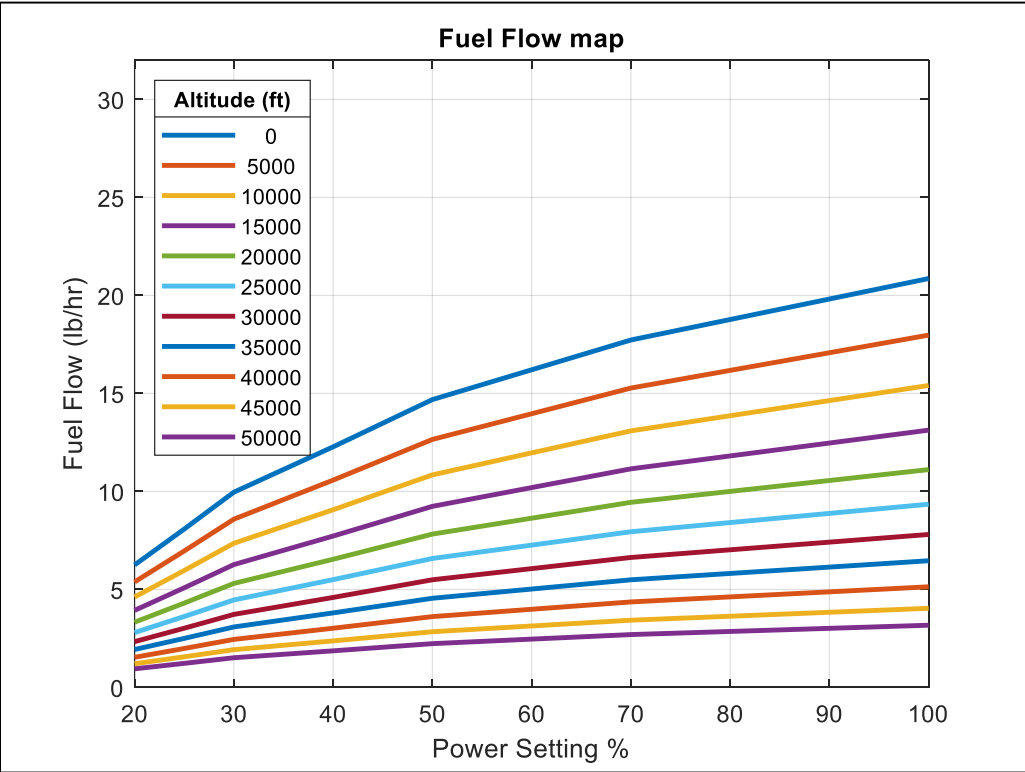
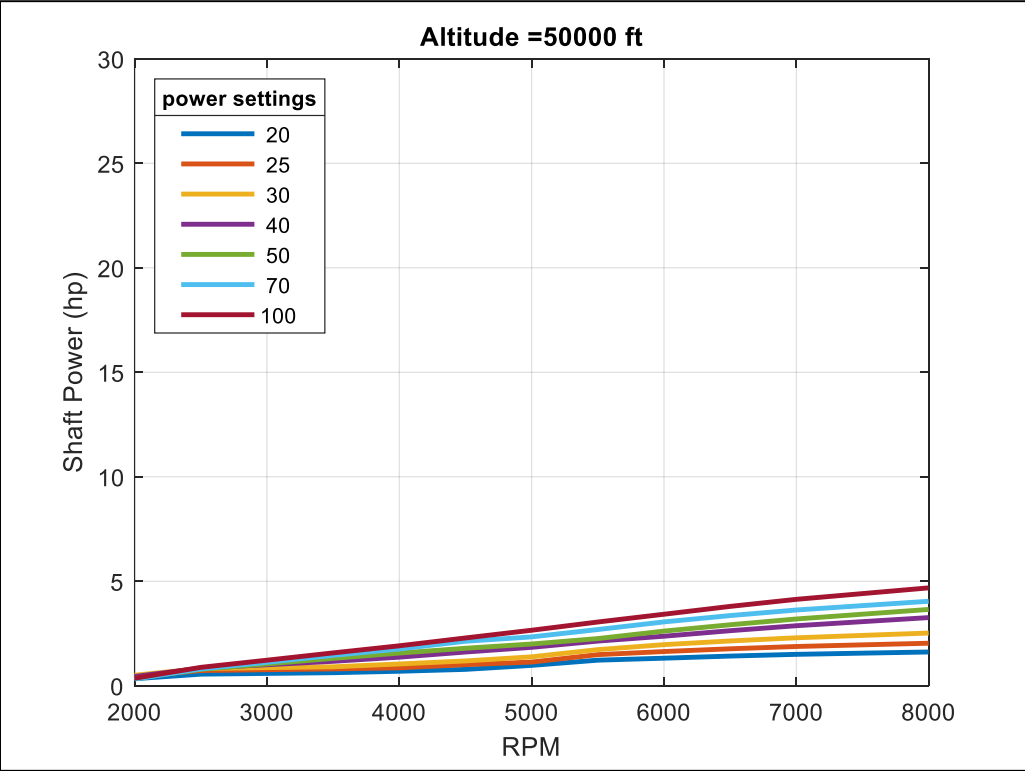




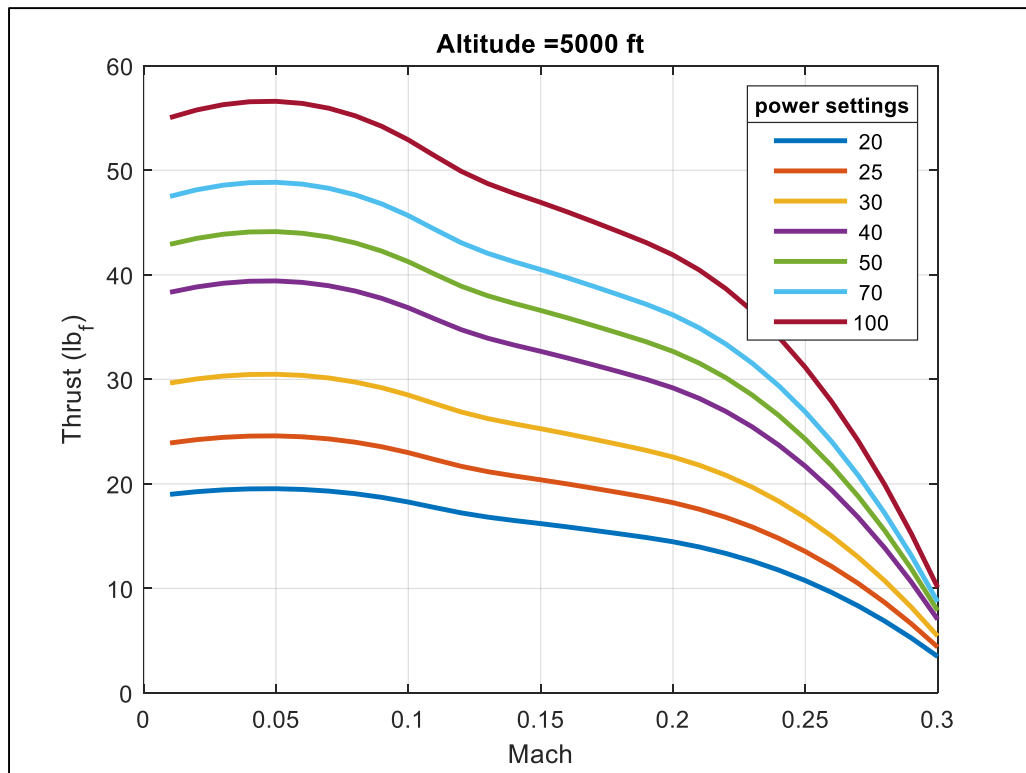
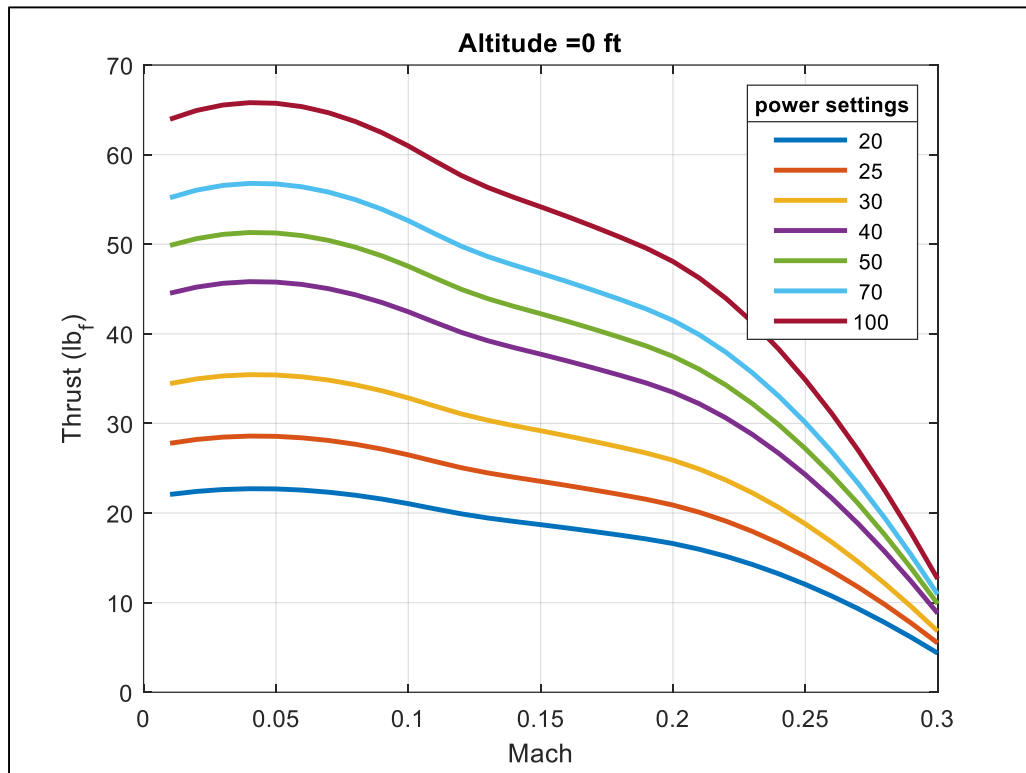


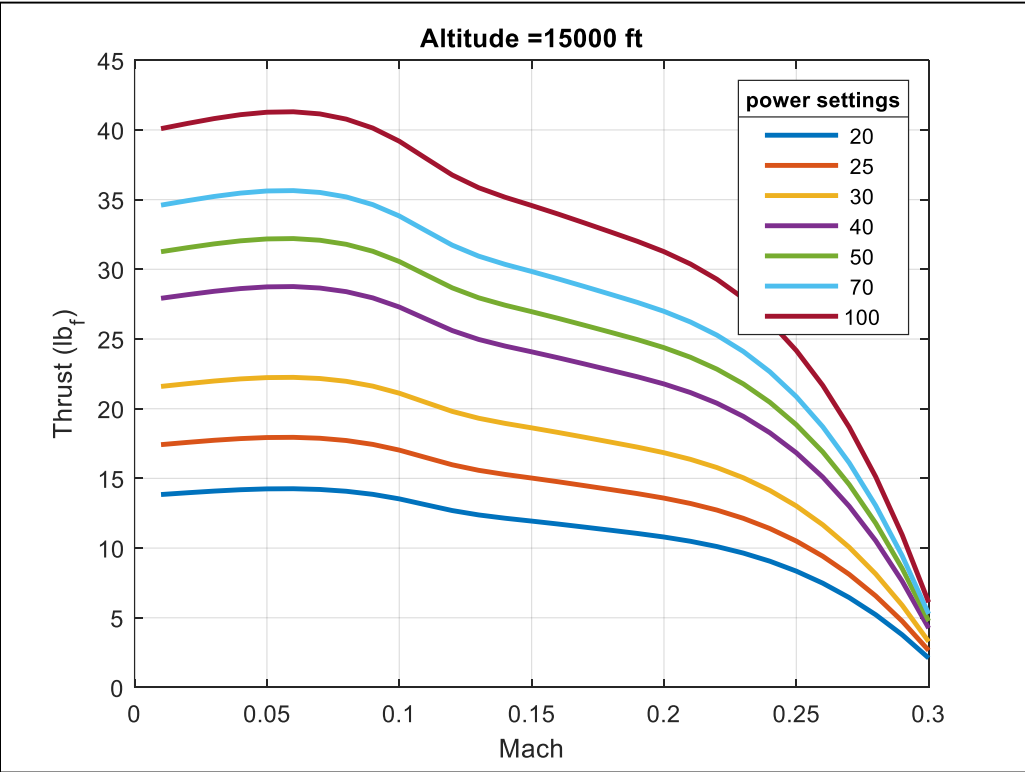
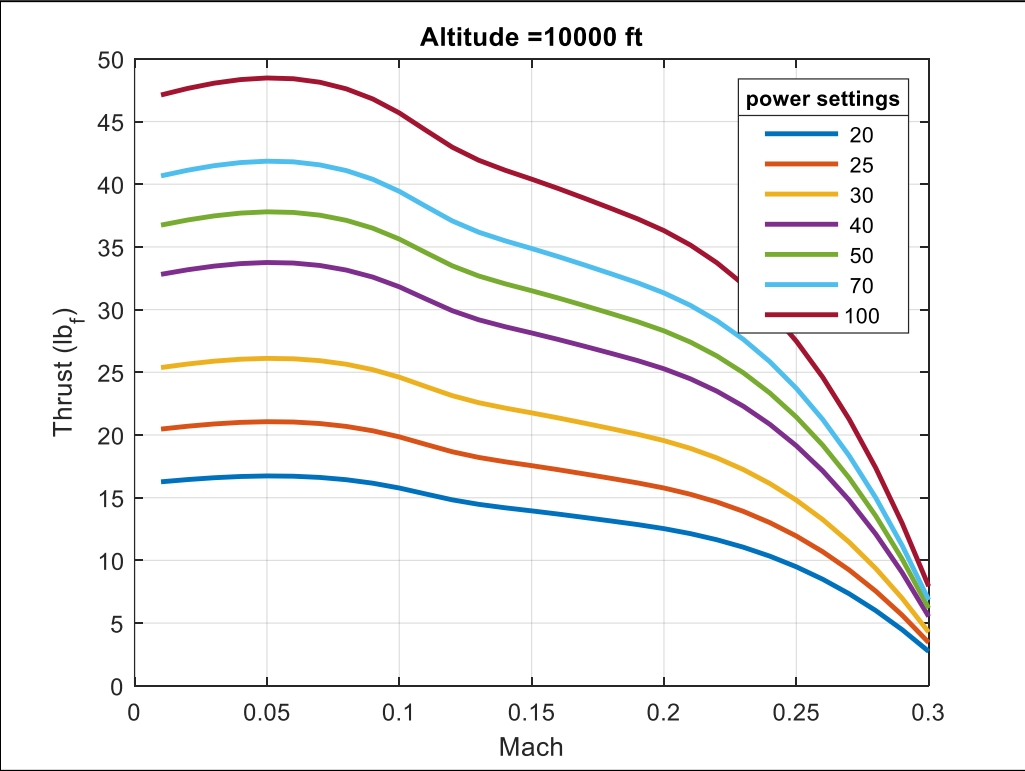


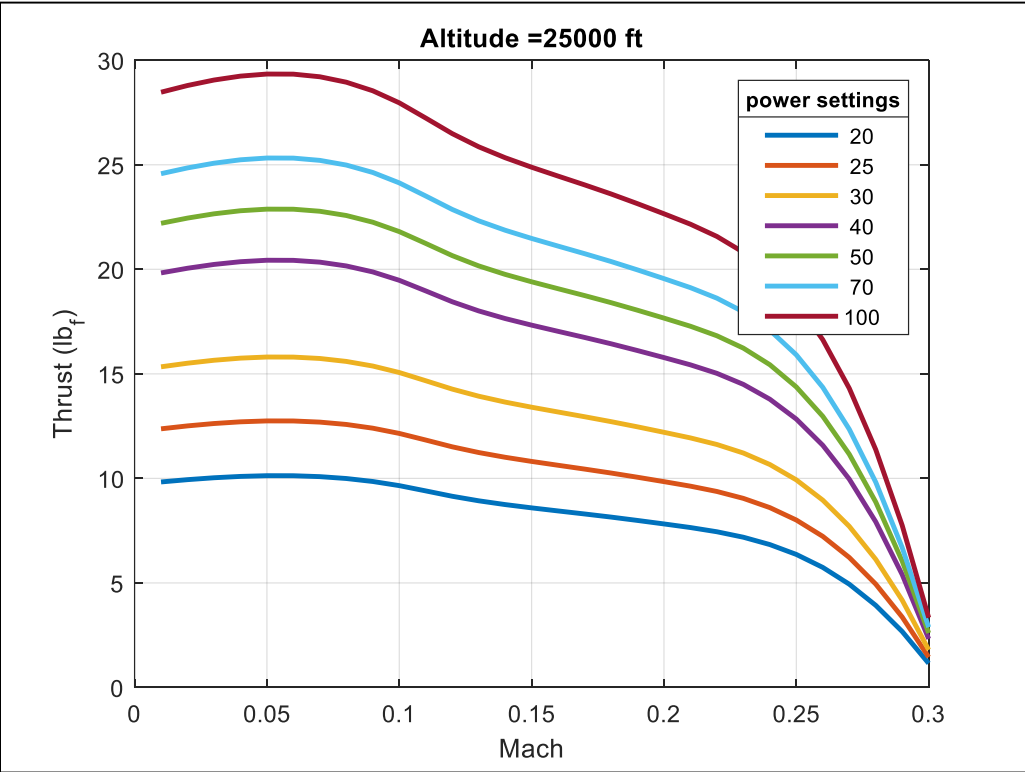
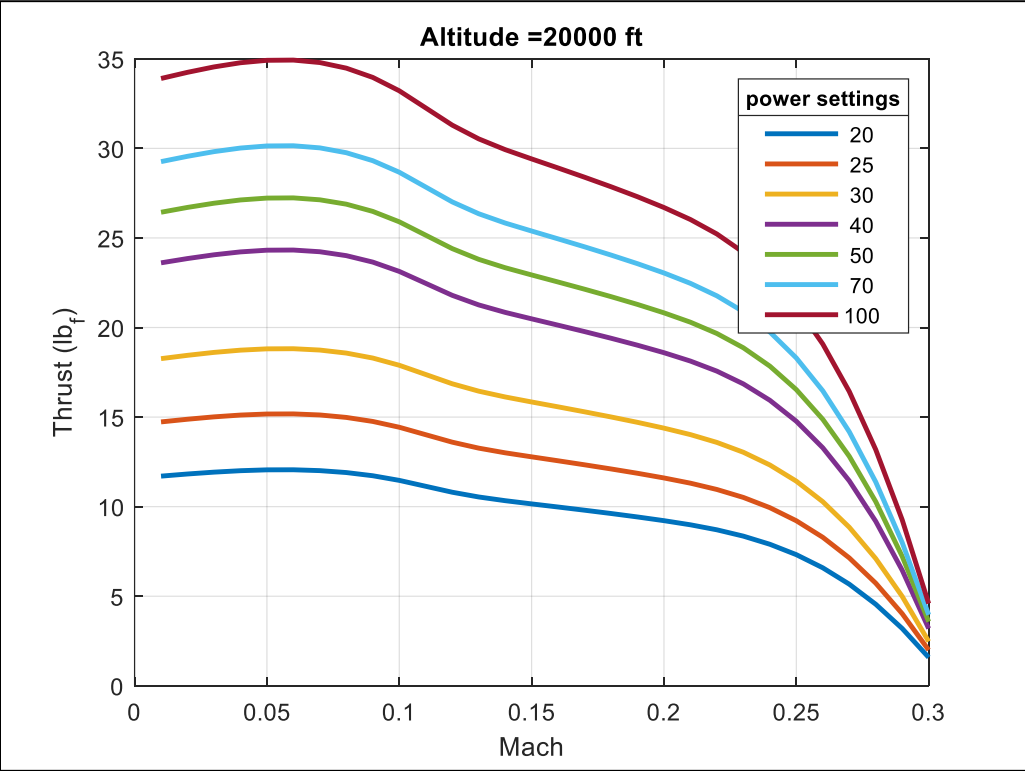


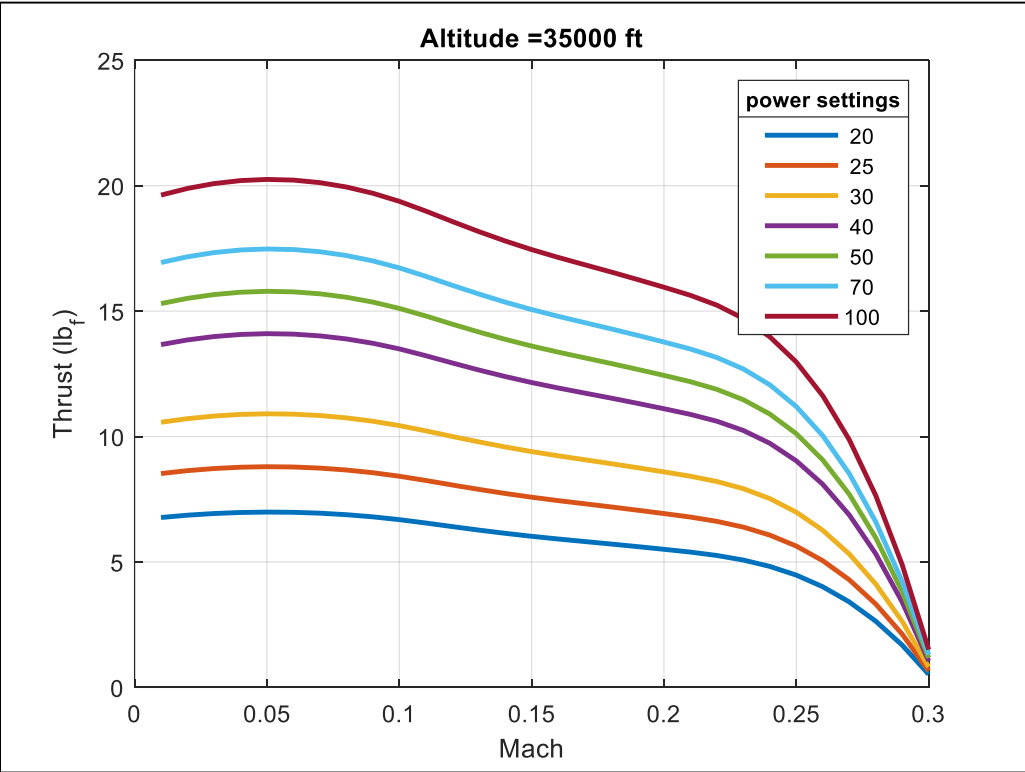
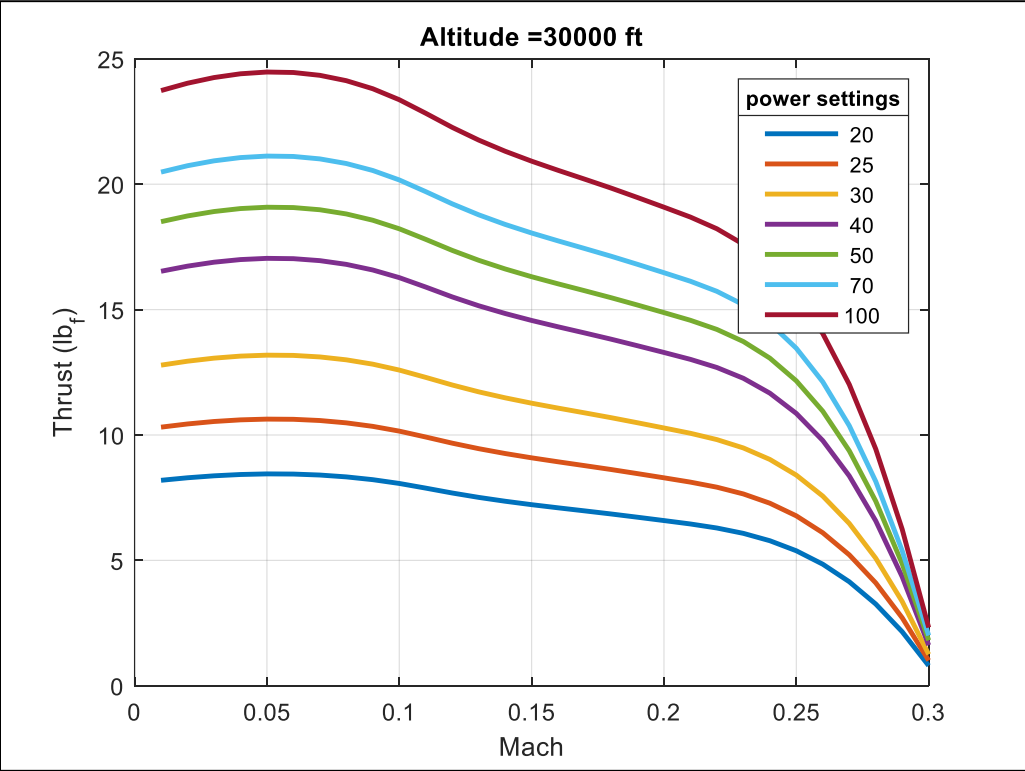


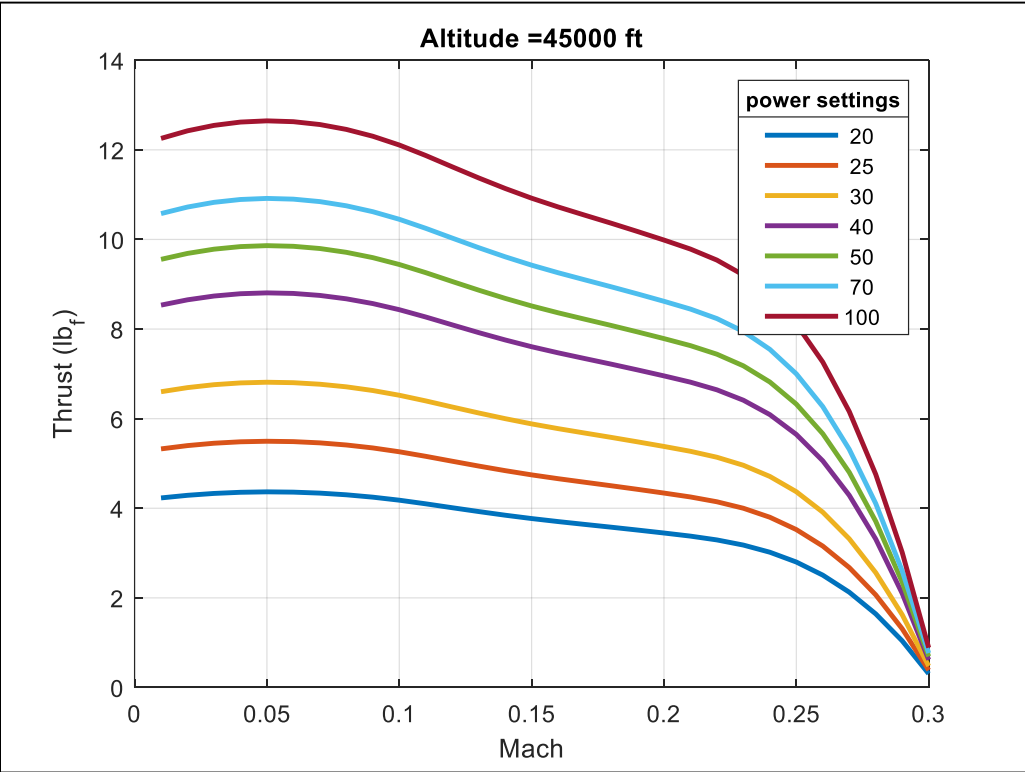
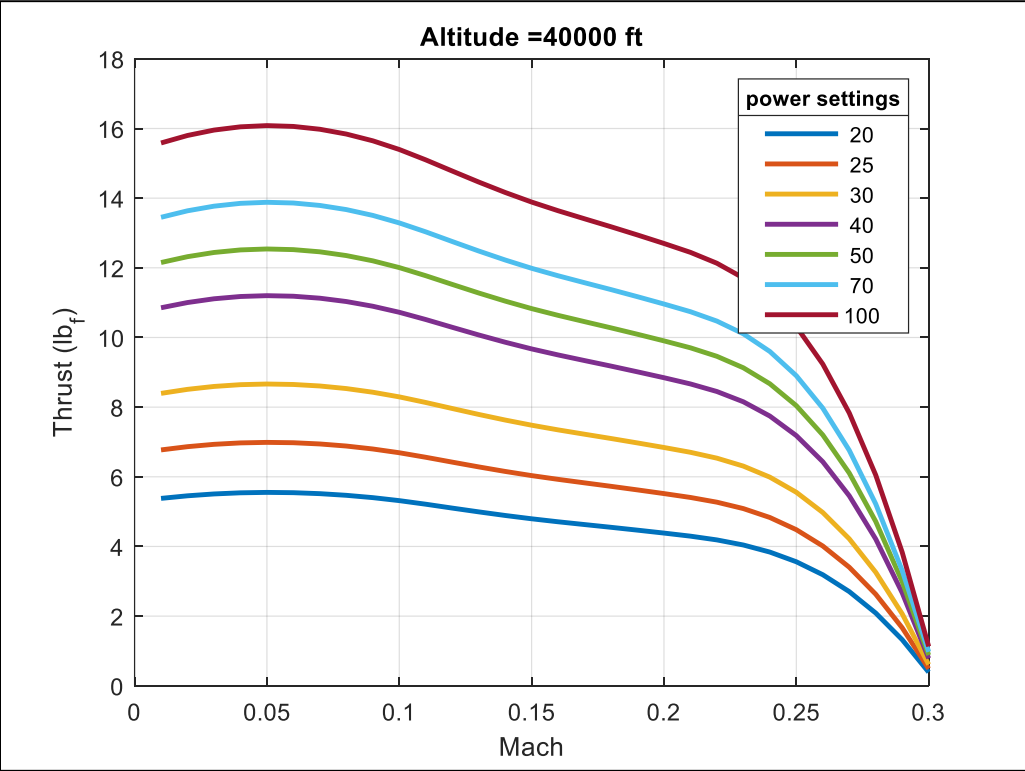
1.3.2 Thrust Maps

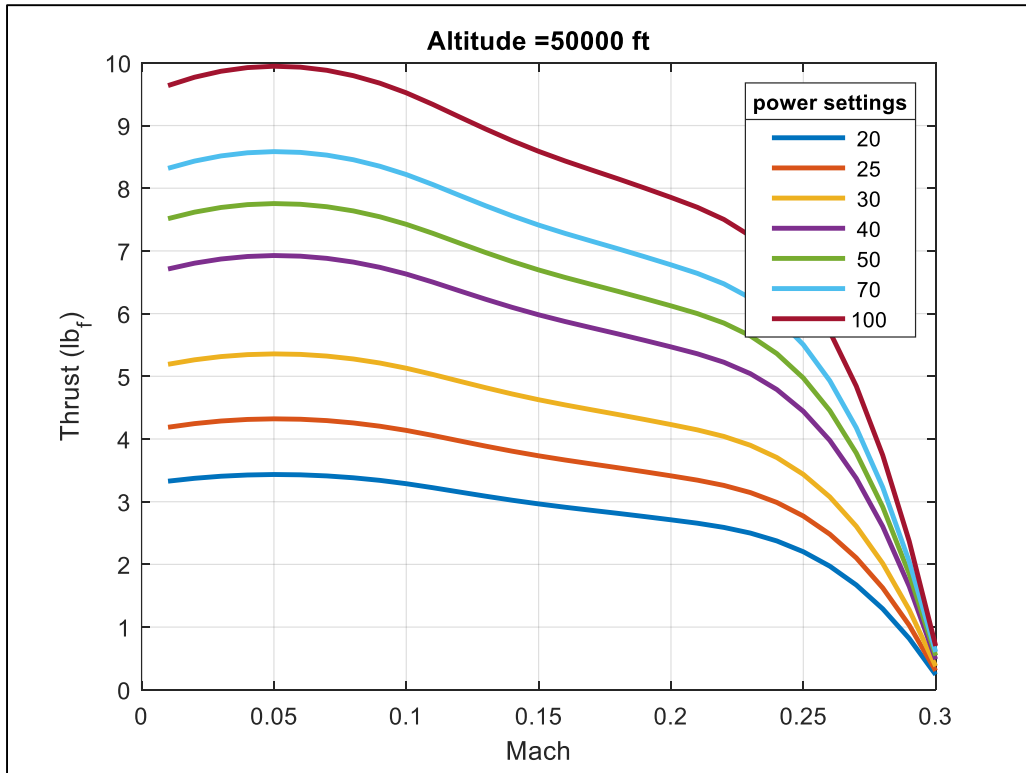












1.3.3 Fuel Flow Map

