MANUAL: Fast Multirotor Performance Prediction Method

Tsaltas, Julia D. and Bramesfeld, Götz
Ryerson University, Toronto, Ontario, Canada, M5B2K3

Table of Contents

I. Introduction

The fast multirotor performance prediction (FMPP) method is a series of modules that predict rotor speeds and power required for steady, straight and level flight. The method finds force and moment trim solutions for a multirotor vehicle over a range of flight speeds. This method is a MATLAB program that can determine steady, level trim solutions of a multirotor vehicle.

This manual includes FMPP setup instructions, descriptions of the main high-level functions used in the MATLAB program, a diagram of the program algorithm, and descriptions of output variables. The Appendix holds more information about the program.

II. FMPP Startup

1) File and Folder Setup

The program folder needs to have the following files and folder: main file, all of the functions, the "input" folder, and the "rotor" folder. It is important that the input and rotor folders start with lower case letters.

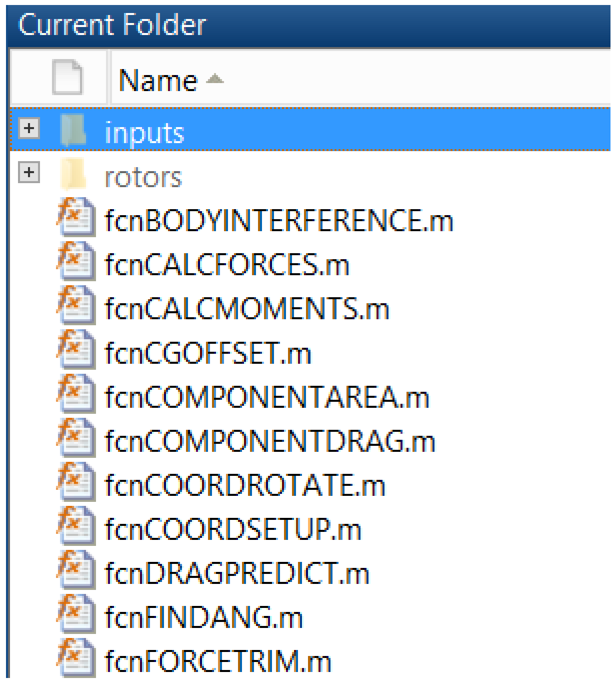


Figure 1: File organization

2) Input File Selection

The text files in the input folder can have any name. The exact file name is entered into the 5th line of the "Main_FMPP" program as the "strFILE" variable (Figure 2). All input files must have the same format as shown in Figure 3.

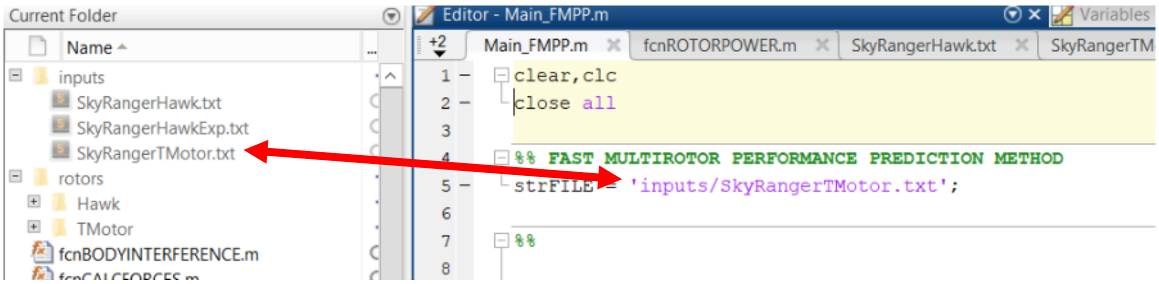


Figure 2: Input file name entry

3) Input File Setup

Figure 3 shows the layout of the input file and sample geometries for the components of the SkyRanger vehicle components using Hawk 15” rotors.

Notes:

- Please note that the program uses equal (=), number (#), quotation marks ("), and colons (:) as special identifiers
- The number directly after the equal sign will be used as the variable value.
- Anything typed after the number will not be considered as the variable value (see Lines 15, 34, and 61 in Figure 3).
- Comments can be written anywhere

```
Input file for Multirotor Vehicle Performance Model
Input file in m/N/secPayload

Platform:       SkyRanger
Rotor type:     Hawk 15"
Payload type:   HDZoom

Please note that the program uses equal, number,", and : signs as special recognizers!

Flow Velocity (m/s)                     seqV        =  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Atmospheric Conditions
----------------------
Temperature [K]:                        flowTEMP    =   281.15
Altitude [m]:                           flowALT     =   320 (Kitchener)
Sea level density [kg/m3]:              flowRHO     =   1.225
Dynamic viscosity:                      flowMU      =   0.00001846
M [kg/mol]:                             flowM       =   0.0289644
R [J/mol*K]:                            flowR       =   8.314
alpha_T [K/m]:                          flowALPHAT  =   0.006


Flight Orientation
------------------
Climb angle [deg from x-y plane]:       angCLIMBdeg = 0
Wind side angle [deg within x-y plane]: angSIDEdeg = 0
Number of leading rotors (1 or 2):      numLEADROTOR= 1


Rotor Geometry and Properties
-----------------------------
Rotor type:                             geomTypeROTOR   ="Hawk"
Number of rotors:                       geomNumROTORS   = 4
Rotor diameter [m]:                     geomDIAMETER    = 0.381 (15")
Number of blades:                       geomNumBLADES   = 2


Vehicle Geometry [m]
----------------
Arm length:                             geomARMlength       = 0.2
Arm radius:                             geomARMradius       = 0.01
Body height (top face to bottom face):  geomBODYheight      = 0.145
Body radius (radius of top face):       geomBODYradius      = 0.1
Leg length:                             geomLEGlength       = 0.295
Leg radius:                             geomLEGradius       = 0.01
Leg centre radius:                      geomLEGcentreradius = 0.179
Leg centre height:                      geomLEGcentreheight = 0.159
Payload length:                         geomPAYLOADlength   = 0.165
Payload radius:                         geomPAYLOADradius   = 0.045
Payload height (from origin to mid axis): geomPAYLOADheight = 0.19
Motor height:                           geomMOTORheight     = 0.03
Motor radius:                           geomMOTORradius     = 0.02
Rotor hub height (mid motor to mid rotor hub): geomHUBheight = 0.0314
CG height (from origin to CG):          geomCGheight        = 0.0275

Vehicle Component Masses [kg]
------------------------
Motor mass:                             massMOTOR   = 0.085
Arm mass:                               massARM     = 0.063
Leg mass:                               massLEG     = 0.029
Payload mass:                           massPAYLOAD = 0 (no payload attached)
Body mass;                              massBODY    = 2
Total vehicle mass:                     massVEHICLE = 2.65
```

Figure 3: Sample input file for FMPP

The following sections describe the input file entries line by line:

Flow Velocity:

- Line 10: [vector] Indicate forward velocity range by entering each number in the range including spaces

```
10   Flow Velocity (m/s)                     seqV        =  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Atmospheric conditions: [values] Properties of the day

- Line 14: [value] Temperature in Kelvin
- Line 15: [value] Altitude in meters
- Line 16: [value] Sea level density in [kg/m3]
- Line 17: [value] Dynamic viscosity in N s/m2 at temperature
- Line 18: [value] Molar mass of air in [kg/mol]
- Line 19: [value] Gas constant in [J/mol*k]
- Line 20: [value] Temperature coefficient in [K/m]

```
12   Atmospheric Conditions
13   ----------------------
14   Temperature [K]:                        flowTEMP    =   281.15
15   Altitude [m]:                           flowALT     =   320 (Kitchener)
16   Sea level density [kg/m3]:              flowRHO     =   1.225
17   Dynamic viscosity:                      flowMU      =   0.00001846
18   M [kg/mol]:                             flowM       =   0.0289644
19   R [J/mol*K]:                            flowR       =   8.314
20   alpha_T [K/m]:                          flowALPHAT  =   0.006
```

Flight Orientation:

- Line 25: [value] “Climb angle” and Line 26: “Wind side angle” are unused in prediction method and currently serve as placeholders
- Line 27: [value 1 or 2 only] One leading rotor for a quadrotor is “+” configuration; Two leading rotors for a quadrotor is “x”.

```
23   Flight Orientation
24   ------------------
25   Climb angle [deg from x-y plane]:       angCLIMBdeg = 0
26   Wind side angle [deg within x-y plane]: angSIDEdeg = 0
27   Number of leading rotors (1 or 2):      numLEADROTOR= 1
```

Rotor Geometry and Properties:

- Line 32: [string] Rotor type is the name of the folder that contains the BEMT generated tables. Do not put space between first quotation “ symbol and equal = sign.
- Line 33: [value] Number of rotors on vehicle
- Line 34: [value] Rotor diameter in meters
- Line 35: [value] Number of blades of selected rotor

```
30   Rotor Geometry and Properties
31   -----------------------------
32   Rotor type:                       geomTypeROTOR    ="Hawk"
33   Number of rotors:                 geomNumROTORS    = 4
34   Rotor diameter [m]:               geomDIAMETER     = 0.381 (15")
35   Number of blades:                 geomNumBLADES    = 2
```

Vehicle Component Masses: [value] mass of component in kilograms

```
56   Vehicle Component Masses [kg]
57   ------------------------
58   Motor mass:                       massMOTOR   = 0.085
59   Arm mass:                         massARM     = 0.063
60   Leg mass:                         massLEG     = 0.029
61   Payload mass:                     massPAYLOAD = 0 (no payload attached)
62   Body mass;                        massBODY    = 2
63   Total vehicle mass:               massVEHICLE = 2.65
```

Vehicle Geometry: [value] distance in meters (See Figure 4 and Figure 5 for diagrams of component geometry locations)

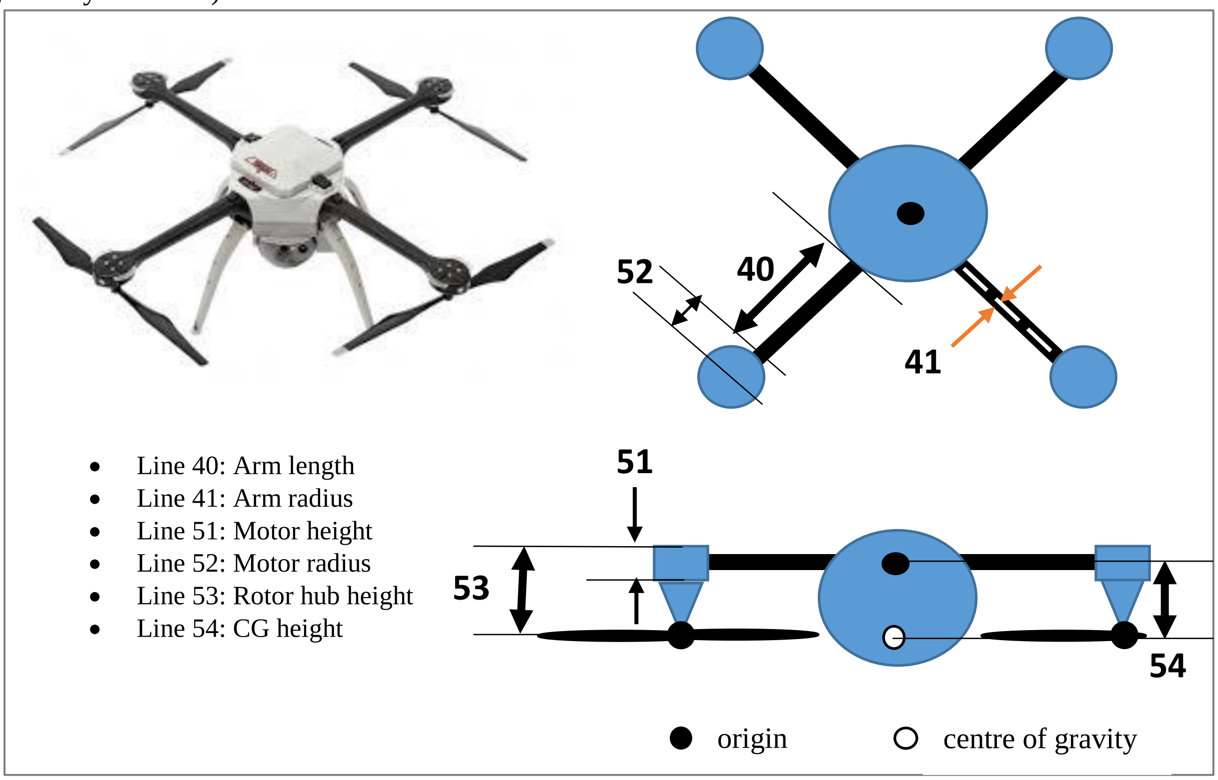
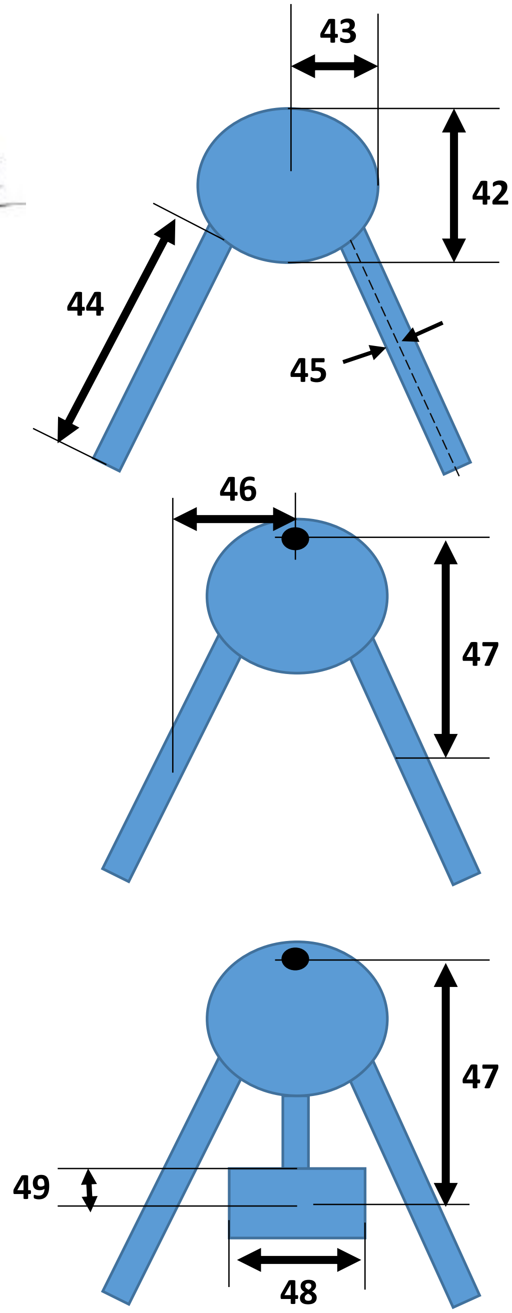


Figure 4: Input geometries of arm, motor, rotor hub and location of centre of gravity from origin

```
Vehicle Geometry [m]
----------------
Arm length:                            geomARMlength        = 0.2
Arm radius:                            geomARMradius        = 0.01
Body height (top face to bottom face): geomBODYheight       = 0.145
Body radius (radius of top face):      geomBODYradius       = 0.1
Leg length:                            geomLEGlength        = 0.295
Leg radius:                            geomLEGradius        = 0.01
Leg centre radius:                     geomLEGcentreradius = 0.179
Leg centre height:                     geomLEGcentreheight = 0.159
Payload length:                        geomPAYLOADlength   = 0.165
Payload radius:                        geomPAYLOADradius   = 0.045
Payload height (from origin to mid axis): geomPAYLOADheight = 0.19
Motor height:                          geomMOTORheight      = 0.03
Motor radius:                          geomMOTORradius      = 0.02
Rotor hub height (mid motor to mid rotor hub): geomHUBheight = 0.0314
CG height (from origin to CG):         geomCGheight         = 0.0275
```



- Line 42: Body height
- Line 43: Body radius
- Line 44: Leg length
- Line 45: Leg radius
- Line 46: Leg centre radius
- Line 47: Leg centre height
- Line 48: Payload length
- Line 49: Payload radius
- Line 50: Payload height

● origin ○ centre of gravity

Figure 5: Input geometries of body, landing gear (leg), and payload

4) Rotor Folder Setup

Figure 6 shows the structure of the "rotor" folder using Hawk rotor and TMotor rotor examples. The rotor name entered in Line 32 of the input text file must be the same name as the folder within the "rotor" folder.

The naming convention of the rotor performance file names is:

pitch angle _ rotor name

If the pitch angle, in degrees, is negative, the letter "n" is placed as the first character of the file name.

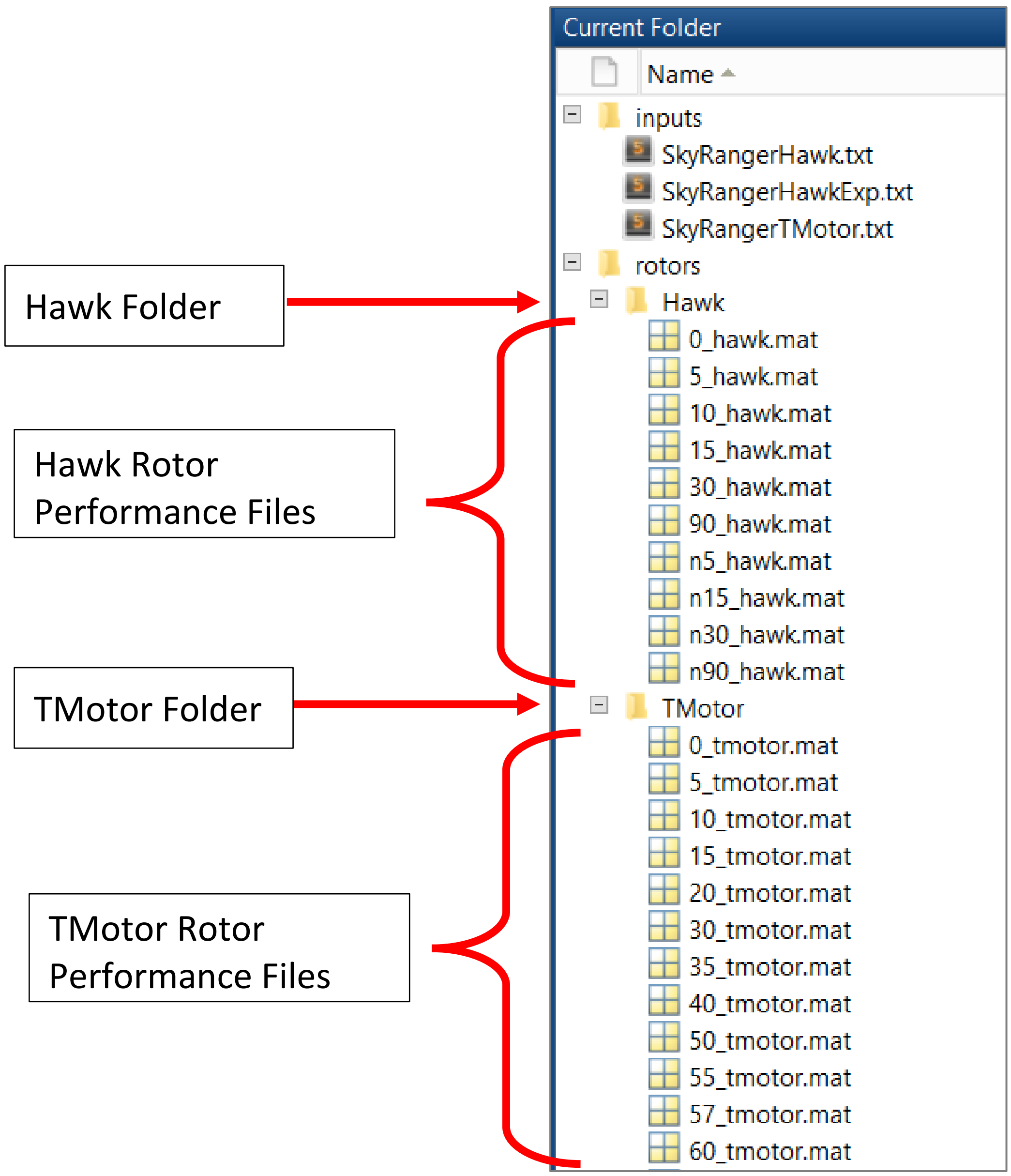


Figure 6: Rotor folder and file organization

Figure 7 shows an example of the MATLAB table format for the file named "5_tmotor.mat". The workplace variable will have a different name as the file name because MATLAB does not allow for numbers to start a variable name (see tmotor_5deg table name in **Figure 7**).

Note:

- Columns containing "_rho" such as Thrust_rho indicated that the values in the columns are force and moments divided by density.

tmotor_5deg

164x10 table

	1 q	2 mu_inf	3 Thrust_rho	4 RPM	5 CP	6 Q_rho	7 Px_rho	8 Mx_rho	9 Py_rho	10 My_rho
1	0	0	3.6174	2000	9.3641e-04	0.0806	-4.1539e-17	3.7763e-18	-6.0420e-17	-6.0420e-17
2	0.6125	0.0209	3.6425	2000	9.2061e-04	0.0792	-0.0258	-0.0327	-0.0058	-0.0645
3	2.4500	0.0418	3.7648	2000	8.7706e-04	0.0755	-0.0497	-0.0625	-0.0031	-0.1131
4	5.5125	0.0627	3.9230	2000	8.3261e-04	0.0716	-0.0724	-0.0927	0.0044	-0.1463
5	9.8000	0.0835	4.0964	2000	8.0615e-04	0.0694	-0.0926	-0.1227	0.0122	-0.1613
6	15.3125	0.1044	4.2776	2000	7.9624e-04	0.0685	-0.1126	-0.1519	0.0198	-0.1639
7	22.0500	0.1253	4.4474	2000	7.9385e-04	0.0683	-0.1269	-0.1807	0.0275	-0.1601
8	30.0125	0.1462	4.5990	2000	8.0172e-04	0.0690	-0.1493	-0.2064	0.0318	-0.1532

Figure 7: Sample format of 5_tmotor rotor performance lookup table

The general format for the rotor performance lookup table is shown in **Table 1** and **Table 2**. **Table 1** has the file name a1_rotorname.mat, where a1 is the pitch angle in degrees. The file is organized by sequential dynamic pressure values, q1, q2, q3, etc. for the same rotor speed. The dynamic pressure values are then repeated for the next rotor speed set. **Table 2** has the same table format for a second file name "a2_rotorname.mat", where a2 is a different pitch angle than a1.

Table 1: Example file name – a1_rotorname.mat

q	**mu_inf**	**Thrust_rho**	**RPM**	**CP**	**Q_rho**	**Fx_rho**	**Mx_rho**	**Fy_rho**	**My_rho**
q1			2000						
q2									
q3									
q4									
…									
…									
q1			3000						
q2									

---									
q1			4000						
q2									
…									
…									

Table 2: Example file name – a2_rotorname.mat

q	mu_inf	Thrust_rho	RPM	CP	Q_rho	Fx_rho	Mx_rho	Fy_rho	My_rho
q1			2000						
q2									
…									
…									
q1			3000						
q2									
…									
…									

5) Setting Up Rotor Performance Tables Using BEMT Rotor Analysis Code

The prediction lookup tables provided in the rotor folder were generated using the blade-element momentum theory model by Tim Carroll. Load up the "BEMT_analysis" file to setup the performance sweep inputs for the desired rotor.

To generate rotor files of the correct format used in the FMPP code, changes were made to the "Performance_sweep.m" file in the BEMT Rotor Analysis Code. The "Performance_sweep" file is accessed in the BEMT Rotor Analysis Code in the following sequence:

Rotor Analysis Code > BEMT Module > Performance Sweeps > Performance_sweep.m

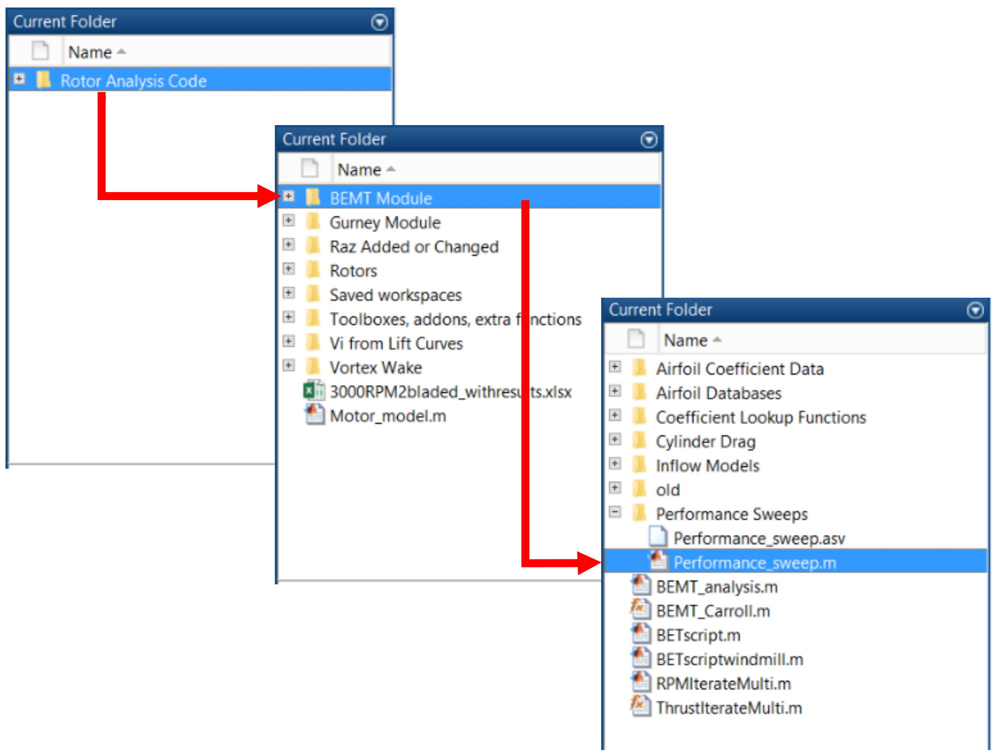


Figure 8: Folder tree to access "Performance_sweep.m" file

The following lines of code were modified in the BEMT model under "case 1" within the "Performance_sweep" file to accommodate the required FMPP table format. The modifications include adding loops for the rotor speed variable "rpm" as highlighted by the box labelled #1. Box #2 shows the order of variables for the lookup table with the air density divided from each of the rotor force and moment results. Finally, box #3 shows the workspace variable name given to the lookup table data with the corresponding table variable names.

```
% PERFORMANCE SWEEP SCRIPT

  switch options.sweep_type

    case 1

  flow.inflow_angle = 90
  perf_Vsweep = [];

  for rpm=2000:1000:7000

      oper.rpm = rpm;

    for V = options.sweep_range(1):options.sweep_range(2):options.sweep_range(3)

      flow.V        =   V;
      [perf]        =   BEMT_Carroll(blade, flow, oper, rotor, wake, options);
      perf_Vsweep   = [perf_Vsweep; perf];

    end

  end

        q            = [perf_Vsweep.q];
        mu_inf       = [perf_Vsweep.mu_freestream];
        T_rho        = [perf_Vsweep.T]./flow.rho;
        RPM          = [perf_Vsweep.rpm];
        CP           = [perf_Vsweep.CP];
        Q_rho        = [perf_Vsweep.Q]./flow.rho;
        Nx_rho       = [perf_Vsweep.Nx]./flow.rho;
        Ny_rho       = [perf_Vsweep.Ny]./flow.rho;
        Mx_rho       = [perf_Vsweep.Mx]./flow.rho;
        My_rho       = [perf_Vsweep.My]./flow.rho;

tmotor_90deg = table(q', mu_inf', T_rho', RPM', CP', Q_rho', Nx_rho', ...
                     Mx_rho', Ny_rho', My_rho',...
     'VariableNames',{'q' 'mu_inf' 'Thrust_rho' 'RPM' 'CP' 'Q_rho' 'Fx_rho'
                     'Mx_rho' 'Fy_rho' 'My_rho'});

%%
    case 2 % Freestream advance ratio (mu_freestream)
```

…etc.

#1

#2

#3

The workspace variable “tmotor_90deg” can be saved into the current document folder with a file name usable by the FMPP code by entering the following into the command window:

```
save('90_tmotor','tmotor_90deg')
```

To update the pitch angle for the next set of rotor performance lookup table data, change the variable `flow.inflow_angle = 90` to the next pitch angle and update the workspace file name `tmotor_90deg` to the next pitch value.

Update rotor names as required. Save files into folder with the same rotor name as the files. Save folder in “rotor” folder in FMPP folder, as shown in **Figure 6**.

Note:

- It is important to ensure that there is sufficient RPM data for rotor file and sufficient angle of attack cases for smooth and connected results.

6) Analysis Flags

There are four analysis flags that can be turned on (1) or off (0). These flags are in the main FMPP program as shown in Figure 9.

The following results will not be calculated if the analysis flag is turned off.

- Moment trim – if zero → rotor thrusts will not be adjusted for moment trim
- Rotor interference – if zero → interference due to the surrounding rotors will not be added to the inflow model
- Body interference – if zero → interference due to the body and freestream interactions will not be added to the inflow model
- Turn body forces on or off – if zero → induced drag and lift of the central body will not be calculated

Turning the analysis types off will improve processing time.

```matlab
%% FAST MULTIROTOR PERFORMANCE PREDICTION METHOD
clear,clc
close all

%% File Input
strFILE = 'inputs/SkyRangerTMotor.txt';

%% Turn analysis types ON (1) or OFF (0)

    analysisMOMENTtrim              = 0; % Turn moment trim on or off
    analysisROTORinterference       = 1; % Turn mutual wake interference velocity on or off
    analysisBODYinterference        = 1; % Turn body interference velocity on or off
    analysisBODYforces              = 1; % Turn body induced drag and lift forces on or off

```

Figure 9: Analysis type selection in FMPP program

7) Adding a Mass Offset

A point mass can be added to the analysis by entering the mass in kilograms and position in meters into the following section of the FMPP program. The centre of gravity will also be updated. This section was created as an option to add a non-symmetric mass distribution.

```
%% FAST MULTIROTOR PERFORMANCE PREDICTION METHOD
clear,clc
close all

%% File Input
strFILE = 'inputs/SkyRangerTMotor.txt';

%% Turn analysis types ON (1) or OFF (0)

    analysisMOMENTtrim              = 1; % Turn moment trim on or off
    analysisROTORinterference       = 1; % Turn mutual wake interference velocity on or off
    analysisBODYinterference        = 1; % Turn body interference velocity on or off
    analysisBODYforces              = 1; % Turn body induced drag and lift forces on or off

%% Add point mass for mass offset test
    massOFFSET = 0; %kg
    positionOFFSET = [0, 0, 0];
```

Figure 10: Adding a mass offset

III. Running the FMPP Program

Once the file setup and rotor file setup are complete, run the “Main_FMPP.m” file.

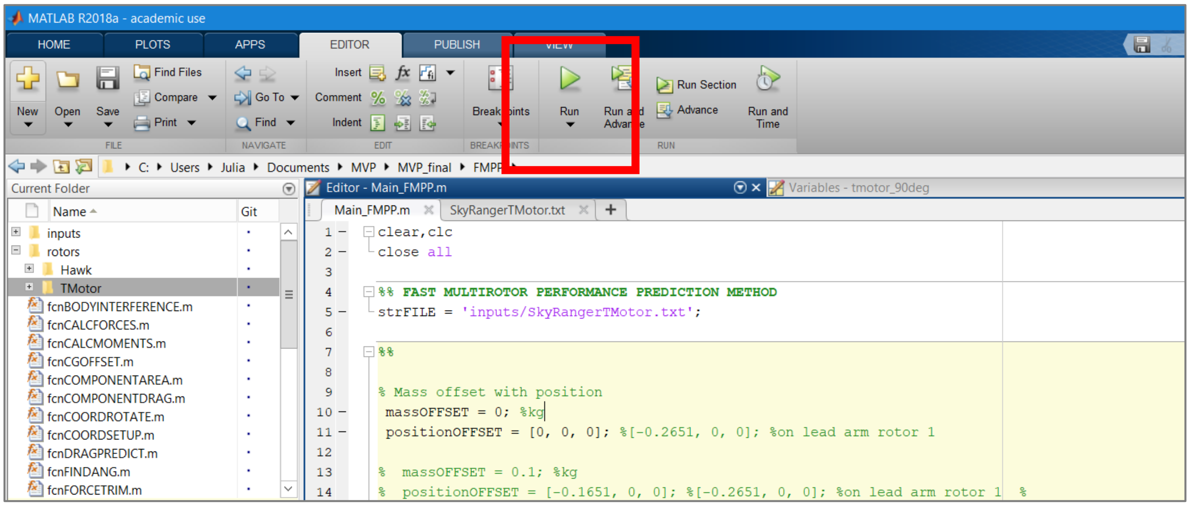


Figure 11: Running FMPP Program

In the command window, updates to the current velocity will show incrementally until the program finishes.

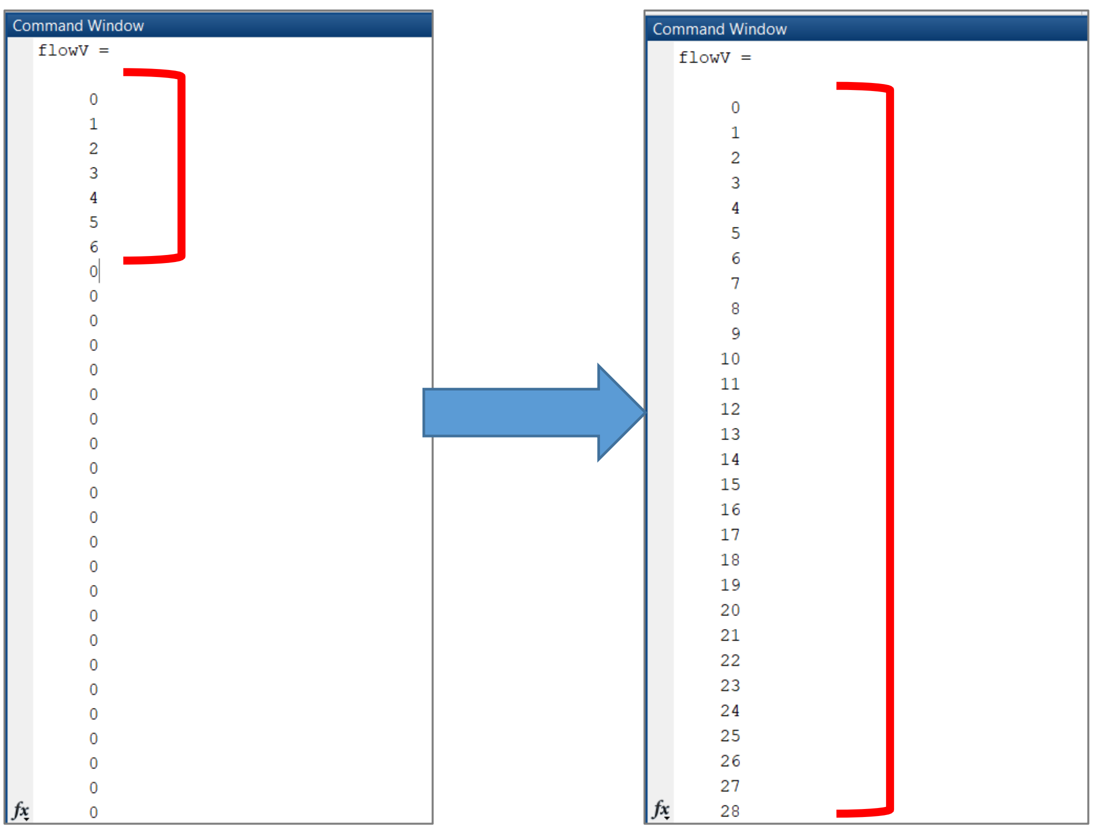


Figure 12: FMPP program command window during run

IV. Algorithm Functions

The main file consists of the following high-level functions. Figure 13 shows the algorithm used for the FMPP method. There are three points of iterations in the algorithm. The first is at the force trim model where the pitch variable is checked for convergence. The second checks for rotor speed convergence within the rotor interference model. The third checks for the total moment trim of the vehicle to be zero.

The Appendix provides a quick reference to the multirotor vehicle component definitions, rotor numbering convention, the method for determining force trim and moment trim, as well as a brief description of the vector summation of interference velocities to the freestream velocity.

Table 3: High-level functions in FMPP MATLAB model in order of appearance

	Figure 13 Label	Function	Description
FMPP Setup		fcnMVPREAD	Reads user input file and assigns MATLAB variables to inputs.
		fcnRECURVE	Develops a Reynolds number log interpolation of drag coefficients of cylinders and spheres.
		fcnCOMPONENTAREA	Calculates the wetted area of each vehicle component.
		fcnLOADTABLES	Reads the lookup table from the provided database.
		fcnCGOFFSET	Updates the CG position if a mass offset is applied to vehicle.
		fcnCOORDSETUP	Sets up coordinates for each component based on input component geometries and vehicle orientation.
Velocity Loop	Fuselage Parasitic Drag	fcnDRAGPREDICT	Uses component geometries, component wetted areas, and flight speed to predict component drag forces and vehicle parasitic power.
	Force Trim	fcnFORCETRIM	Uses an iterative approach to predict vehicle pitch attitude and rotor thrusts for force trimmed flight. All rotors are assigned the same thrust and rotor speed values here.
	Fuselage Interference	fcnBODYINTERFERENCE	Determines interference velocities applied to each rotor due to the fuselage-freestream interactions.
	Rotor Interference	fcnPREDICTRPM	Iterates between the wake interference model and a rotor inflow prediction model to predict rotor inflow conditions needed to predict individual rotor speeds.
	Moment Trim	fcnCALCMOMENTS	Assigns coordinates of vehicle components to predicted rotor and vehicle forces and calculates moments applied to the vehicle.
		fcnMOMENTTRIM	Modifies lead and rear rotor thrusts to drive total residual pitching moments to zero.
	Power Prediction	fcnROTORPOWER	Calculates total rotor and vehicle power based on final rotor CP values.

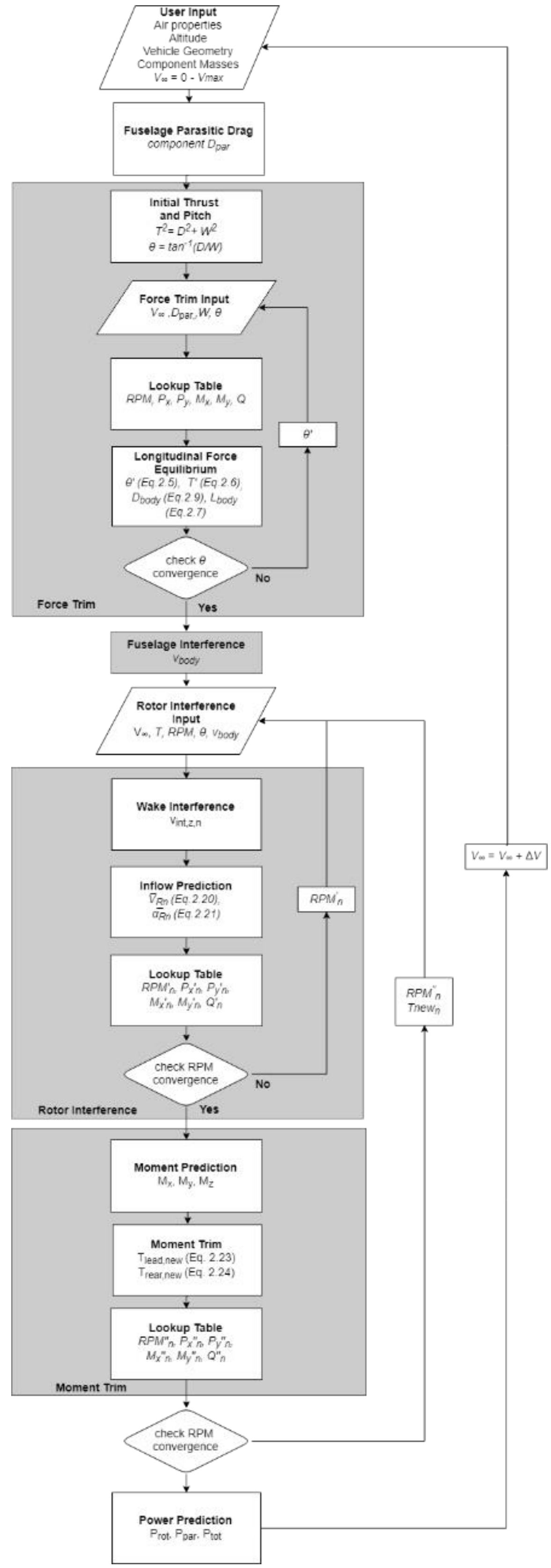


Figure 13: Fast multirotor performance prediction algorithm

V. Outputs

This section describes the output variables provided by the FMPP code. The variable types are categorized by the following:

- rotor forces and moments
- drag and body lift forces
- moments of components
- interference velocities
- power and pitch output variables

In addition, array dimensions are provided describing the variable structure as it is output in the MATLAB workspace.

m = velocity increments set by “seqV” vector input
n = number of rotors

Rotor Forces and Moments

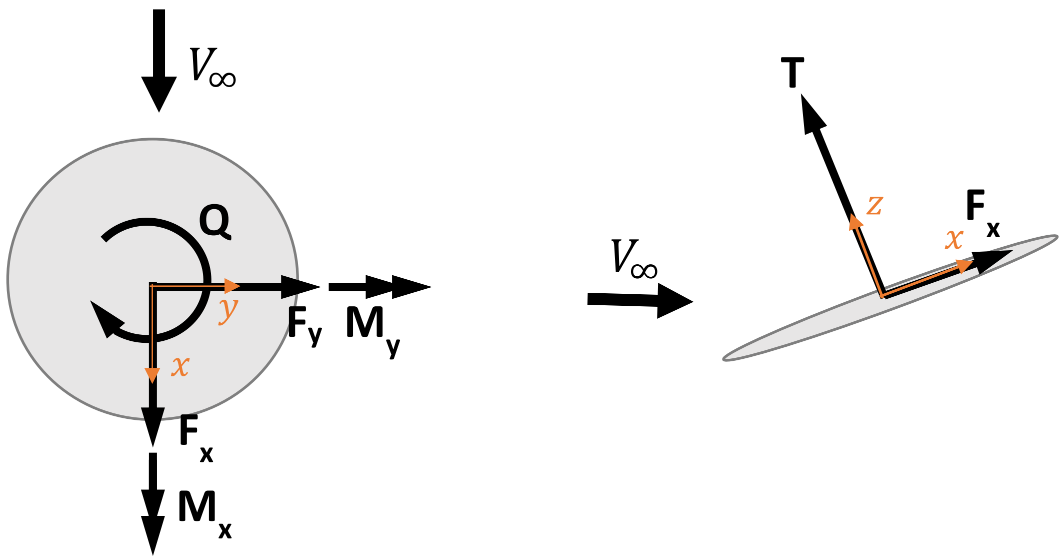


Figure 14: Rotor force and moment convention

Table 4: Output rotor forces and moments

Figure 14 Symbol	Unit	Variable	Variable Name	Array Dimension
T	N	Rotor Thrust	rotorTHRUST	mx1xn
Fx	N	Longitudinal Hub Drag	rotorFx	mx1xn
Fy	N	Lateral Hub Drag	rotorFy	mx1xn
Mx	Nm	Rotor Rolling Moment	rotorMy	mx1xn
My	Nm	Rotor Pitching Moment	rotorFx	mx1xn
Q	Nm	Rotor Torque	rotorQ	mx1xn
	-	Rotor Power Coefficient	rotorCP	mx1xn
	W	Rotor Power	rotorPOWER	mx1xn
	RPM	Rotor Speed	rotorRPM	mx1xn

Drag and Body Lift Forces [N]

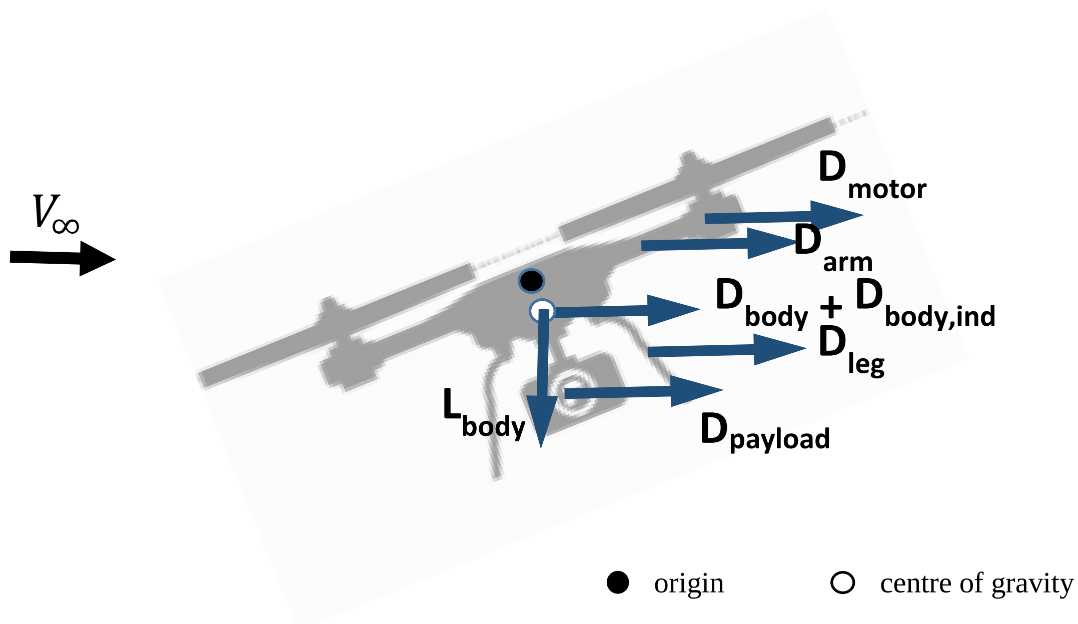


Figure 15: Example of vehicle drag forces and the moment arm of the payload drag, $h_{payload}$.

Table 5: Output moments used to calculate total vehicle moments

Figure 15 Symbol	Variable	Variable Name	Array Dimension
D_{motor}	Drag of the motors	dragMOTOR	mx3xn
D_{arm}	Drag of the arms	dragARM	mx3xn
D_{leg}	Drag of the legs	dragLEG	mx3x4
D_{body}	Parasitic drag of central body	dragBODY	mx3x1
$D_{body,ind}$	Induced drag of central body	dragBODYinduced	mx3x1
L_{body}	Lift of central body	liftBODY	mx3x1
$D_{payload}$	Drag of payload	dragPAYLOAD	mx3x1

Moments of Components [Nm]

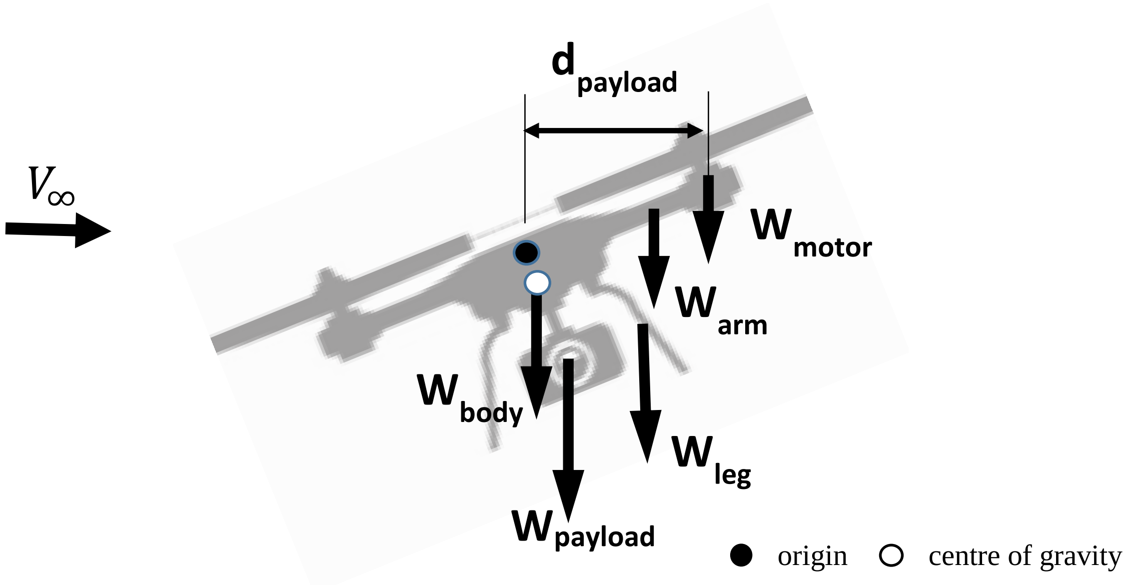


Figure 16: Example of vehicle weight forces and the moment arm of the motor weight, $d_{payload}$.

Table 6: Output moments used to calculate total vehicle moments

Figure 16 Symbol	Variable	Variable Name	Array Dimension
	Moment due to the rotor thrusts	momentTHRUST	mx3xn
	Moment due to the long. hub drag	momentROTORFx	mx3xn
	" " " lateral hub drag	momentROTORFy	mx3xn
	" " " rotor rolling moment	momentROTORMx	mx3xn
	" " " rotor pitching moment	momentROTORMy	mx3xn
	" " " rotor torque	momentROTORQ	mx3xn
	" " " weight of the motors	momentWEIGHTMOTOR	mx3xn
	" " " drag of the motors	momentDRAGMOTOR	mx3xn
	" " " weight of the arms	momentWEIGHTARM	mx3xn
	" " " drag of the arms	momentDRAGARM	mx3xn
	" " " weight of the legs	momentWEIGHTLEG	mx3x4
	" " " drag of the legs	momentDRAGLEG	mx3x4
	" " " weight of the body	momentWEIGHTBODY	mx3x1
	" " " parasitic drag of central body	momentDRAGBODY	mx3x1
	" " " induced drag of central body	momentDRAGBODYinduced	mx3x1
	" " " lift of central body	momentLIFTBODY	mx3x1
	" " " weight of payload	momentWEIGHTPAYLOAD	mx3x1
	" " " drag of payload	momentDRAGPAYLOAD	mx3x1
	" " " weight of mass offset	momentWEIGHTOFFSET	mx3x1
	Total moments of vehicle	momentTOTAL	mx3x1

Interference Velocities [m/s]

Total rotor inflow velocity is the sum of the freestream

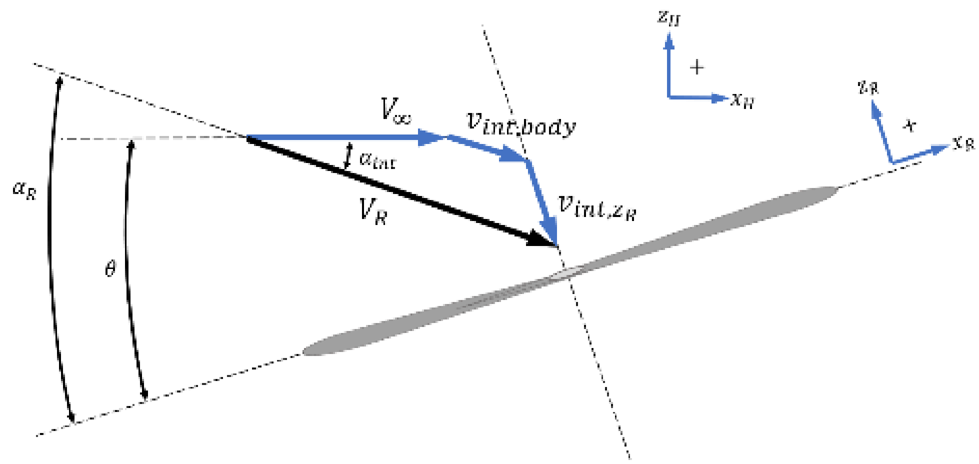


Figure 17: Mutual interference velocity applied to a rotor resulting in an increased inflow velocity and inflow angle relative to freestream velocity and angle

Table 7: Output moments used to calculate total vehicle moments

Figure 17 Symbol	Variable	Variable Name	Array Dimension
V_∞	Freestream velocity	flowV	mx1
$v_{int,zR}$	Mutual interference velocity	vi_int	mx3xn
	Self induced velocity	vi_self	mx3xn
	Mutual interference component	wi	mx3xn
$v_{int,body}$	Fuselage interference	vi_body	mx3xn
α_R	Rotor resultant inflow angle	rotorANGinflow	mx1xn
V_R	Rotor resultant	rotorVELinflow	mx1xn

Power and Pitch Output Variables

Table 8: Power and pitch output variables

Figure Symbol	Unit	Variable	Variable Name	Array Dimension
	W	Rotor power	powerROTOR	mx1xn
	W	Parasitic power	powerPARASITIC	mx1
	W	Vehicle power	powerVEHICLE	mx1
	degree	Vehicle pitch	pitchVEHICLEdeg	mx1

APPENDIX

I. Comparison with Flight Test Data

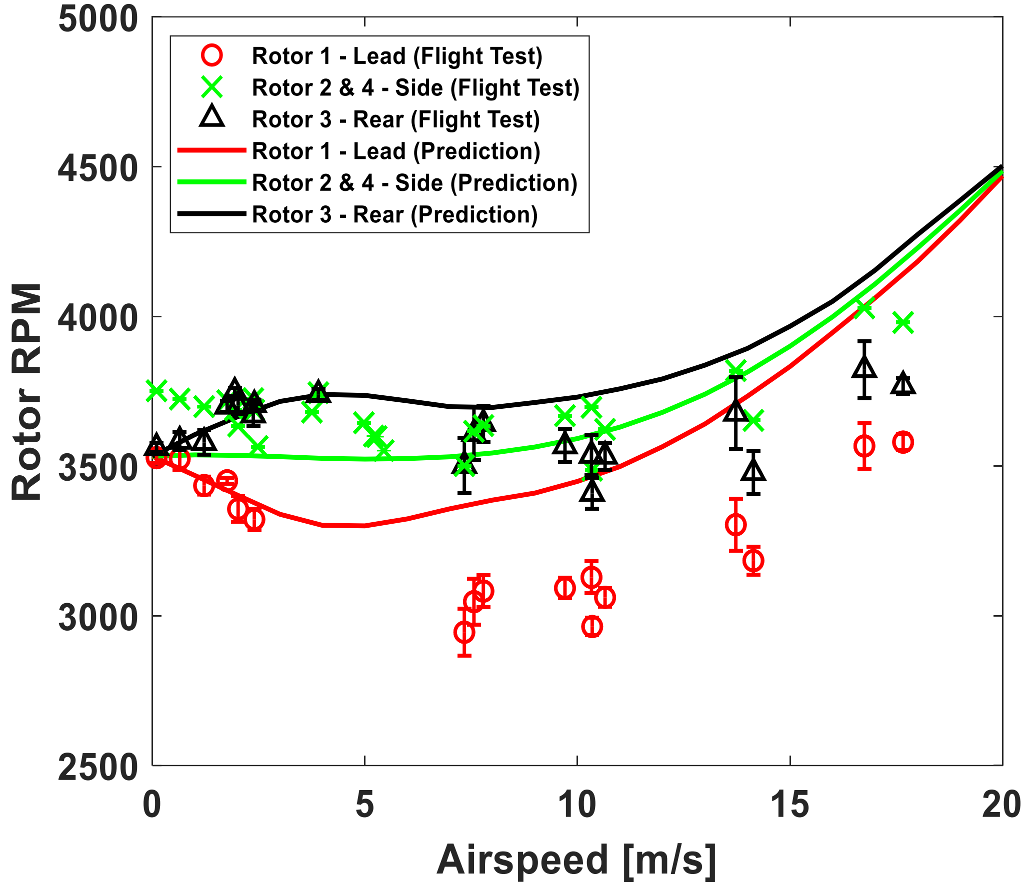


Figure 18: Rotor speed comparison between flight test and prediction data of the SkyRanger with Hawk 15” propeller

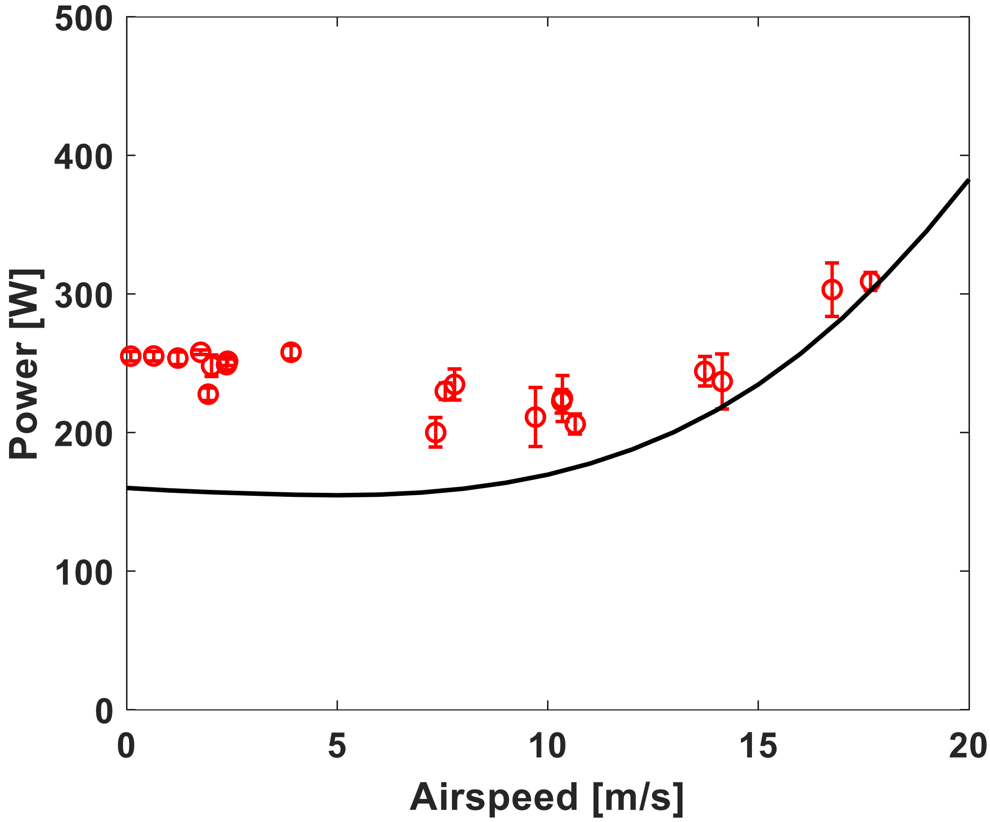


Figure 19: Vehicle power comparison between flight test and prediction data of the SkyRanger with Hawk 15" propeller

Power drawn from battery of the SkyRanger

Predicted power required by aircraft

II. Vehicle Components

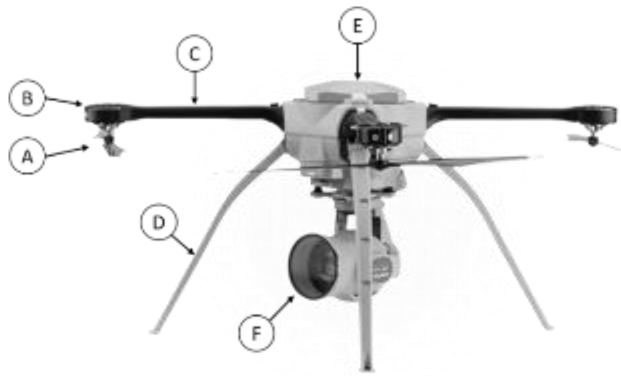


Figure 20: Standard multirotor components A) rotor, B) motor, C) rotor arm, D) landing gear (leg), E) central body, F) payload.[11]

Figure 20 shows the basic multirotor vehicle components. Table 9 also includes the number of component elements on the vehicle. The input file includes an input for number of rotors, n. The landing gear, central body, and payload have a fixed number of components in the FMPP method.

Table 9: Component labels and input number of component elements

Label	Component	No. of Elements
A	Rotor	n
B	Motor	n
C	Rotor Support Arm	n
D	Landing Gear	4
E	Central Body	1
F	Payload	1

III. Flight Orientation

Quadrotors have two main configurations, square and diamond, and anything in between. These configurations can also be referred to as "X" and "+" configurations and refer to the number of leading rotors of a quadrotor. Square, or "X" configuration, has two leading rotors and diamond, or "+" configuration, has one leading rotor.

Figure 21 shows the rotor labels for quadrotors in diamond and square configurations. In diamond configuration, rotors are labelled counter-clockwise starting with the lead rotor. Similarly, the rotors are labelled counter-clockwise in square configuration with rotor 1 labelled as the lead left rotor. The FMPP method assumes all rotors are equal distance from the fuselage centre and have equal angular spacing.

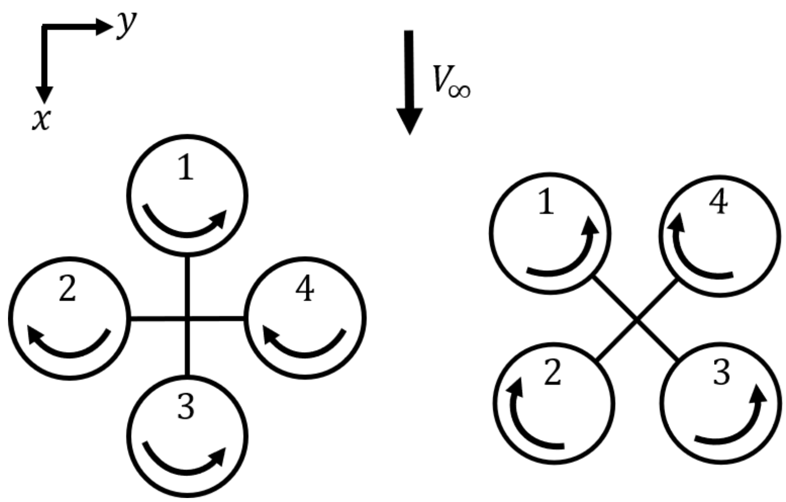


Figure 21: Diamond (left) and square (right) configurations.

IV. Force Trim

Figure 22 shows a free-body diagram of the major forces that act in the longitudinal plane of a multirotor vehicle. During steady and level flight, the loads that the rotor develops, thrust, T, and hub drag, F_x, must be in equilibrium with the vehicle weight, W, and the aerodynamic forces of the fuselage, namely parasitic and induced drag, D_{par} and D_{ind} respectively, and negative lift, L_{body}. The equations of motion are based on the set that was developed for the original multirotor vehicle performance model and expanded using the forces of the body lift. Only forces in the longitudinal plane, such as thrust, hub drag, fuselage drag, body lift, and weight are considered for force trim.

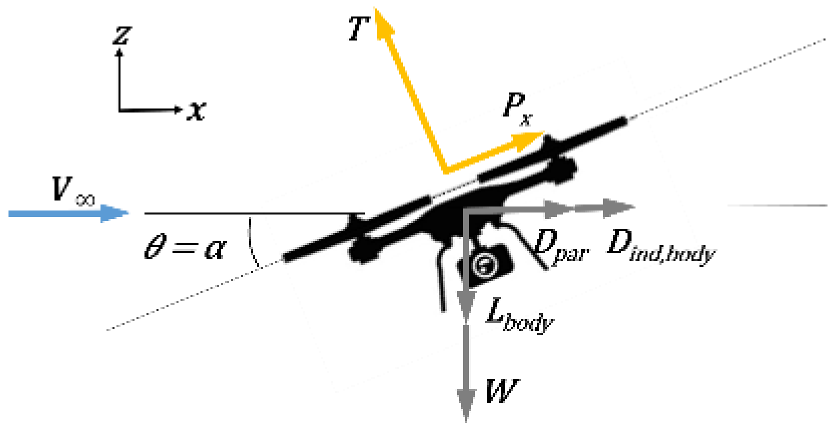


Figure 22: Free-body diagram of aerodynamic forces on a multirotor vehicle

There are two equations that are central to achieving force trim. The first is the calculation of pitch attitude in degrees and the second is the calculation of thrust. Vehicle forces, such as weight, drag, and lift, are divided by the number of rotors, n, when calculating for the thrust of one rotor.

$$\sin\theta = \frac{F_x W + F_x L_{body} + T D_{par} + T D_{ind,body}}{T^2 + {F_x}^2}$$

$$T = \sqrt{(W + L_{body} + F_x \sin\theta)^2 + (D_{par} + D_{ind,body} + F_x \cos\theta)^2}$$

Table 10 shows a list of variables used in the force trim model.

Table 10: List of variables in force trim function

Symbol	Variable	Variable Name	Array Dimension	Origin of Calculation
[redacted]				
W	Vehicle Weight	massVEHICLE*g	1x1	Input file
D_{par}	Total Parasitic Drag	dragVEHICLE	1xm	Parasitic drag function
$D_{ind,body}$	Induced Drag of Body	dragBODYinduced	1xm	Force trim function
L_{body}	Negative Lift of Body	liftBODY	1xm	Force trim function
F_x	Total Rotor Long. Hub Drag	rotorFx	1xmxn	Force trim function/Lookup table
Θ	Vehicle Pitch in Degrees	pitchVEHICLEdeg	1xm	Force trim function/Eq. 1
T	Rotor Thrust	rotorTHRUST	1xmxn	Force trim function/Eq. 2

V. Moment Trim

The moment trim model uses the total residual vehicle moment, calculated by the moment calculation function, and calculates new lead and rear rotor thrusts to resolve the total vehicle moment to zero. Figure 23 shows an example of the parasitic drag of the payload generating a negative residual pitching moment on the vehicle. Residual moments are the moments generated by the forces applied to the vehicle about the vehicle origin.

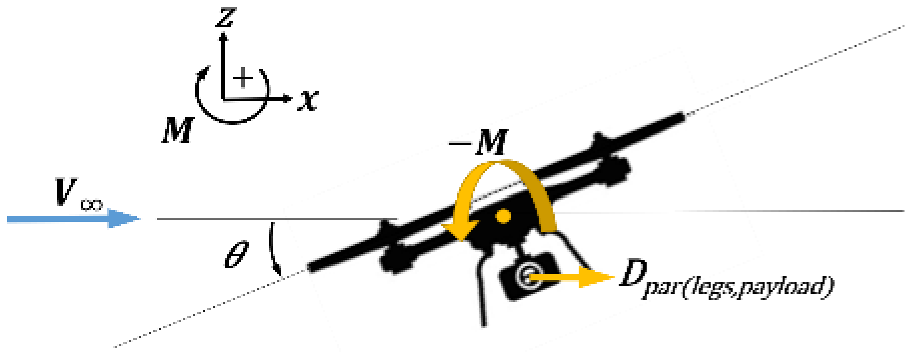


Figure 23: Free-body diagram of aerodynamic moments on a multirotor vehicle

The total vehicle moment is the sum of the moments of all vehicle forces and moments calculated as:

$$\Re_i = \sum (\vec{r}_i \times \vec{F}_i) + \sum \vec{M}_{Rotors}$$

where R_i is the cross product between the component moment arm, r_i, and the force component, F_i.

The total vehicle moment is used to calculate the change in rotor thrust of the lead and rear rotors to drive the total vehicle moment to zero.

$$T_{lead,new} = T_{lead} - \frac{\Re_{Total}}{2R}$$

$$T_{lead,new} = T_{lead} + \frac{\Re_{Total}}{2R}$$

For a two-leading rotor configuration, the total moment is divided by four to calculate the change in thrust for each rotor.

VI. Interference Velocity

Total rotor inflow velocity is the sum of the freestream, body interference velocity and the mutual interference velocity.

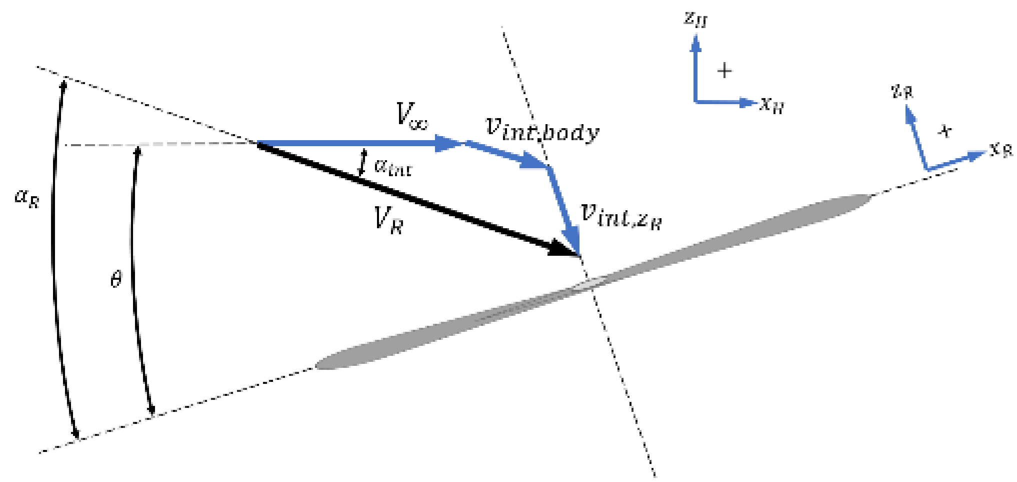


Figure 24: Mutual interference velocity applied to a rotor resulting in an increased inflow velocity and inflow angle relative to freestream velocity and angle

The resultant

$$V_R = \sqrt{(V_\infty + v_{int,body,x} - v_{int,z}\sin\theta)^2 + (v_{int,body,x} + v_{int,z}\cos\theta)^2}$$

$$\alpha_R = \theta - \alpha_{int}$$