Performance analysis of mini-propellers based on FlightGear

Tomáš Vogeltanz

Citation: AIP Conference Proceedings **1738**, 120019 (2016); doi: 10.1063/1.4951902
View online: http://dx.doi.org/10.1063/1.4951902
View Table of Contents: http://aip.scitation.org/toc/apc/1738/1
Published by the American Institute of Physics

Performance Analysis of mini-Propellers Based on FlightGear

Tomáš Vogeltanz

Tomas Bata University in Zlín, Faculty of Applied Informatics
Department of Informatics and Artificial Intelligence
nám. T.G. Masaryka 5555, 760 01 Zlín, CZECH REPUBLIC

Abstract. This paper presents a performance analysis of three mini-propellers based on the FlightGear flight simulator. Although a basic propeller analysis has to be performed before the use of FlightGear, for a complex and more practical performance analysis, it is advantageous to use a propeller model in cooperation with a particular aircraft model. This approach may determine whether the propeller has sufficient quality in respect of aircraft requirements. In the first section, the software used for the analysis is illustrated. Then, the parameters of the analyzed mini-propellers and the tested UAV are described. Finally, the main section shows and discusses the results of the performance analysis of the mini-propellers.

Keywords: analysis; aircraft; FlightGear; JavaProp; modelling; performance; propeller; simulation; software; UAV
PACS: 07.05.Tp

INTRODUCTION

Same as aerodynamic design, the design of propeller is very important. When the inefficient or inappropriate propeller is used, all advantages of excellent-designed aerodynamics of an unmanned aerial vehicle (UAV) may remain underutilized. [1] [2]

The propeller theory is analogous to wing theory in which the propeller blade is considered to be a lifting surface about which there is a circulation associated with the bound vorticity and a vortex sheet is continuously shed from the trailing edge [2]. More about the propeller theory and its method in design can be found in [3].

Since propeller thrust is largely dependent on the lift and propeller power is largely dependent on the drag, the efficiency of a propeller is highly influenced by the Reynolds number [4]. While propeller performance for full-scale airplanes has been well documented since the pioneering days of aviation, data on mini-propellers has been rather scarce [5].

SOFTWARE FOR PERFORMANCE ANALYSIS

In this section, software which assists in the mini-propeller performance analysis is briefly introduced. The connections between applications are illustrated in Figure 1.

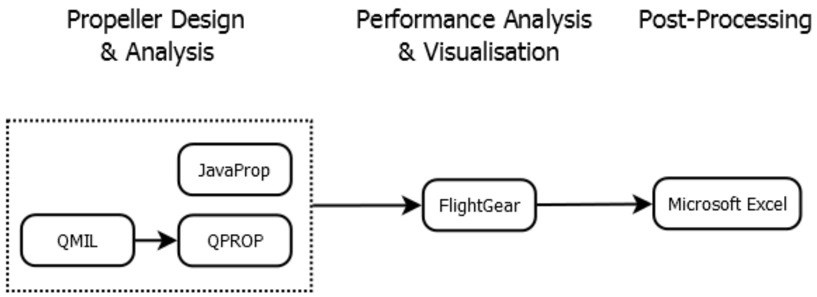
Propeller Design & Analysis
Performance Analysis & Visualisation
Post-Processing
JavaProp
QMIL
QPROP
FlightGear
Microsoft Excel

FIGURE 1. Software Connection for Performance Analysis of mini-Propellers

International Conference of Numerical Analysis and Applied Mathematics 2015 (ICNAAM 2015)
AIP Conf. Proc. 1738, 120019-1–120019-4; doi: 10.1063/1.4951902
Published by AIP Publishing. 978-0-7354-1392-4/$30.00

First of all, data which describes a mini-propeller has to be obtained. For this purpose, applications with, for example, a blade-element-method implementation can be found on the Internet [6] [7]. Furthermore, CFD software should be usable too. Another possible solution is to use measured data from research papers [8] [4].

JavaProp is based on the blade element theory which usually works well, when the power and thrust loading of the propeller (power per disk area) is relatively small, as it is the case for most mini-propellers. However, the blade element method is limited when flow separation occurs e.g. at static conditions. Moreover, JavaProp comes with a set of airfoil polars which can represent only a limited model of the whole range of possible airfoil sections. Finally, the flow field around a propeller is complex and fully three dimensional with boundary layers, Mach number effects and local flow separation. This problem may be difficult to model accurately with the most sophisticated tools such as Navier-Stokes solvers which typically require long time to calculation. On the other hand, JavaProp can perform an analysis in fractions of a second for price of the accuracy. Nevertheless, designed propellers may be exported to IGS or DXF file in 3D, then printed by using 3D printer, and analyzed in real conditions for comparison. IGS files can be opened by, for instance, FreeCAD. [1] [6]

QPROP is an analysis program for predicting the performance of propeller-motor or windmill-generator combinations. Propeller/windmill design program QMIL generates propeller geometries for the Minimum Induced Loss (MIL) condition, or windmill geometries for the MIL or Maximum Total Power (MTP) conditions. QPROP and QMIL use an extension of the classical blade-element/vortex formulation. This method applies equally well to propellers and windmills, including that with very high disk loadings. QPROP requires a fairly detailed description of the propeller geometry and blade airfoil characteristics. [1] [7]

A basic propeller analysis is already performed by using JavaProp (or QPROP, respectively); however, for a complex and more practical performance analysis, it is advantageous to use a propeller model in cooperation with a particular aircraft model. In other words, it is appropriate to simulate an aircraft with a propeller, for example in FlightGear Flight Simulator, to obtain more specific results. This approach may determine whether the propeller has sufficient quality in respect of the aircraft requirements.

FlightGear is an open-source flight simulator for the modelling and simulation of aircraft with visualization. It is possible to choose between three primary FDMs: JSBSim, YASim, and UIUC. It is also possible to add new dynamics models or even interface to external proprietary flight dynamics models. [10] [1] [9]

Except of FDM xml configuration files (i.e. aircraft, engine and propeller configuration files), other files are required for use with FlightGear flight simulator which include an electric system file, an autopilot file, and a 3D graphical model specification file. The final required file is a file to tie the previous files together in order to perform complete simulation. [10] [1]

Finally, Microsoft Excel was used for the post-processing of results, including the creation of graphs.

SPECIFICATION OF UAV AND PROPELLERS

As indicated, the results from JavaProp and QPROP can be used for the creation of propeller (thruster) configuration file. The most important parameters are [11]: the tables of thrust and power coefficients (C_T and C_P) versus the advance ratio (J), the rotational inertia (ixx), a diameter, the number of blades (numblades), a minimum (minpitch) and maximum (maxpitch) pitch angle, and the direction of rotation (sense).

Rascal 110 UAV [12] with one Zenoah G-26A Engine (2207.27 W) was used for the performance analysis of mini-propellers. Rascal 110 UAV and its autopilot were described in [10] and [13].

Table 1 illustrates the specification of the mini-propellers used in this research: an 18x8 propeller [14] originally used in Rascal UAV, an APC 12×12E propeller measured in [8], and a propeller designed in JavaProp (named as JavaProp 18×12). Figure 2 shows APC 12×12E propeller in real (a) [15] and designed JavaProp 18×12 propeller which is displayed in FreeCAD (b).

TABLE 1. Specification of mini-Propellers

Parameter	original 18×8	APC 12×12E	JavaProp 18×12
ixx (kg·m^2)	0.00115	0.00022387	0.00113334
diameter (inch)	18	12	18
numblades (-)	2	2	3
minpitch (°)	30	23.2	15.6
maxpitch (°)	30	23.2	15.6

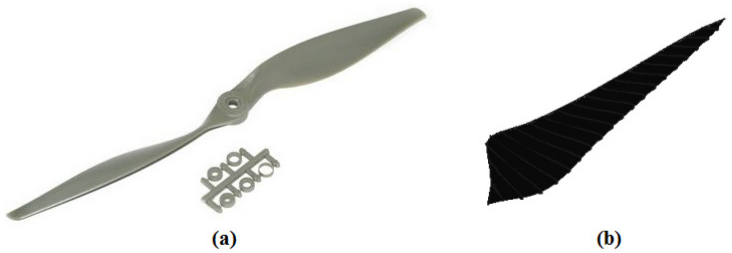

FIGURE 2. APC 12×12E Propeller (a) and One Blade of JavaProp 18×12 Propeller (b)

PERFORMANCE ANALYSIS IN FLIGHTGEAR SIMULATOR

The same simulation conditions were used for every mini-propeller test, for example, the Santorini map (ICAO id = LGSR) was chosen, the FDM rate was 100 Hz, and the pause of the simulation at its beginning was set. The last one is important because other specific options, such as commands to the autopilot and the log setting with physical information about the UAV, are accessible only in the simulation window. The UAV takes off from the default place on the ground; the place depends on the map selection - in this case, it is at the altitude of approximately 33 m.

The altitude control was set to change the altitude to 914.4 meters (3000 feet) and held there; thus, the altitude change is approximately 881.4 m. The velocity control was adjusted to hold the top speed and the heading control was set to keep the heading at 0. All settings are applied immediately after the start of the simulation.

Figure 3 shows the results of the climb of the UAV with different mini-propellers and its complete stabilization after several seconds in every experiment. As can be seen, the shape of the processes is very similar; the values vary of course. Table 2 illustrates the most important results from the graphs; v_{max} means the maximum velocity, $v_{c\text{-}c\_diff}$ means the climb-cruise velocity difference, i.e. the difference between the maximum speed and the speed before the cruise flight, $t_{0\text{-}100\_km/h}$ means the time necessary for the acceleration from 0 to 100 km/h, and t_{3000_feet} means time when the target altitude (3000 feet) is achieved.

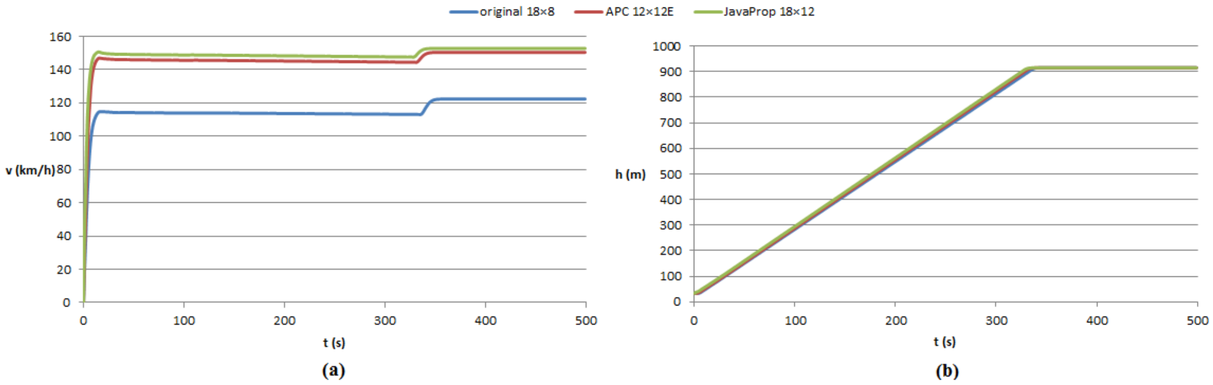

FIGURE 3. Velocity (a) and Altitude (b) Change of UAV with Different mini-Propellers

TABLE 2. Main Results of mini-Propeller Performance Analysis

Parameter	original 18×8	APC 12×12E	JavaProp 18×12
v_{max} (km/h)	122.3821393	150.3945052	152.7791515
$v_{c\text{-}c\ diff}$ (km/h)	9.3440695	6.033409	5.2714039
$t_{0\text{-}100\ km/h}$ (s)	7.693715	4.636985	3.255484
$t_{3000\ feet}$ (s)	341	337	334.5

From the results, it is obvious that the APC 12×12E and JavaProp 18×12 propellers have better performance than the original 18×8 propeller. JavaProp 18×12 definition is based on estimated values; consequently the real results may be different but the error should not be too high. In contrast the APC 12×12E propeller model was created from measured data; and thus, the results should be adequate. The accuracy of the original 18×8 model is unknown.

The maximum velocity of the APC 12×12E propeller is about 18.626 % higher than of the original 18×8 propeller; moreover, the value is even around 19.896 % higher for the JavaProp 18×12. The time for the acceleration from 0 to 100 km/h is around 39.73 % and 57.686 % lower for the APC 12×12E and JavaProp 18×12, respectively.

The time in which the climb ends is almost same for all propellers. This situation is most likely caused by the autopilot which controls the vertical speed; thus this speed cannot be higher than selected (or recommended), and as a result, the autopilot reduces the angle of attack which influences the time of a climb. This statement is also supported by the values of the climb-cruise velocity difference which should be very similar if the angle of attack is same (the forces acting on the UAV are very similar except the thrust) but they are not in our case. In summary, the solution of this problem should be the modification of the adjustment of the PID controller in the autopilot.

CONCLUSION

This paper has presented a performance analysis of three mini-propellers based on the FlightGear flight simulator. First of all, the software used for the analysis was illustrated. The main section of this work described and discussed the specification of the mini-propellers and their performance analysis.

From the results, it is obvious that the appropriate selection of a propeller for a specific UAV can increase the speed without the rise in the power. However, a bigger propeller may not mean bigger performance – the performance is influenced by the construction of the propeller, its structure, airfoil, and also by the physical limits of technology. For example, the original 18×8 propeller does not sufficiently utilize the potential of the UAV; the APC 12×12E and JavaProp 18×12 propellers should be better.

As noted, the time in which the climb ends is almost same for all analyzed propellers in the simulation. If there is a requirement to achieve a higher altitude faster than presented, the setting of the PID controller in the autopilot have to be modified.

ACKNOWLEDGMENTS

This work was supported by the Internal Grant Agency of Tomas Bata University in Zlín under the project No. IGA/FAI/2015/001.

REFERENCES

1. Tomáš Vogeltanz, "A Survey of Free Software for the Design, Analysis, Modelling, and Simulation of an Unmanned Aerial Vehicle", *Archives of Computational Methods in Engineering*, 2015, March 22, 66 p, DOI: 10.1007/s11831-015-9147-y.
2. Q. R. Wald, "The aerodynamics of propellers", *Progress in Aerospace Sciences*, vol. 42, no. 2, 2006, pp. 85-128, DOI: 10.1016/j.paerosci.2006.04.001.
3. Ralph D. Kimberlin, *Flight Testing of Fixed-Wing Aircraft*, Reston, VA: American Institute of Aeronautics and Astronautics, 2003, ISBN 9781563475641.
4. Robert W. Deters, Gavin K. Ananda, and Michael S. Selig, "Reynolds Number Effects on the Performance of Small-Scale Propellers", *32nd AIAA Applied Aerodynamics Conference*, Atlanta, GA, 2014, June 16-20, 43 p.
5. John B. Brandt, and Michael S. Selig, "Propeller Performance Data at Low Reynolds Numbers", *49th AIAA Aerospace Sciences Meeting*, Orlando, FL, 2011, January 4-7, 18 p.
6. Martin Hepperle, "JavaProp - Design and Analysis of Propellers", 2003, September 8, available: http://www.mh-aerotools.de/airfoils/javaprop.htm.
7. Mark Drela, "QPROP: Propeller/Windmill Analysis and Design", 2007, December 23, available: http://web.mit.edu/drela/Public/web/qprop/.
8. Monal Pankaj Merchant, "Propeller Performance Measurement for Low Reynolds Number Unmanned Aerial Vehicle Applications", MSc. Thesis, Wichita State University, December 2005.
9. FlightGear contributors, "FlightGear Flight Simulator: Features", accessed: 2015, June 18, available: http://www.flightgear.org/about/features/.
10. Tomáš Vogeltanz, and Roman Jašek, "FlightGear Application for Flight Simulation of a mini-UAV", *AIP Conference Proceedings* 1648, American Institute of Physics, Melville, NY, 2015, 550014, DOI: 10.1063/1.4912769.
11. FlightGear contributors, "JSBSim Thrusters: FGPropeller", 2014, December 12, available: http://wiki.flightgear.org/JSBSim_Thrusters#FGPropeller.
12. J. Goppert, "Rascal", 2013, January 13, available: https://github.com/PX4/HIL/tree/master/aircraft/Rascal.
13. Tomáš Vogeltanz, and Roman Jašek, "JSBSim Library for Flight Dynamics Modelling of a mini-UAV", *AIP Conference Proceedings* 1648, American Institute of Physics, Melville, NY, 2015, 550015, DOI: 10.1063/1.4912770.
14. J. Goppert, "18x8", 2013, January 13, available: https://github.com/PX4/HIL/blob/master/aircraft/Rascal/Engines/18x8.xml.
15. Astra, "APC 12 x 12E Elektro", available: http://www.astramodel.cz/images/apc/800x600/RE-LP12012E.jpg?v=0npF.