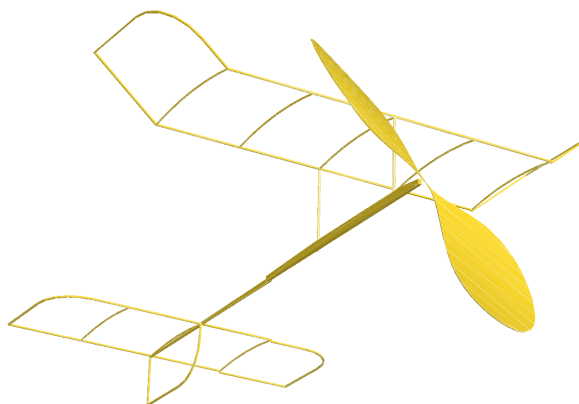# Math Magik Project
Estimating Indoor Model Flight Times

Roie R. Black

December 22, 2021

1   Introduction

In the 1976 edition of the *National Free Flight Society Symposium*, Doug McLean authored an article titled *A Method for Predicting Indoor Model Duration* [2]. Doug's method involves data from actual model flights and simplified formulas for things like the aerodynamics and efficiency of the model, propeller, and the rubber powering the airplane. In this article, we will revisit this approach and create some Python code to help predict the flight times for models we design using *OpenSCAD* [1].

1.1   About This Document

This document is produced using LaTeX, a typesetting program very popular in academia and science. All math in this document has been checked using *Python SymPy*.

2   Python Packages

We will be using two nice Python packages to support this article: —SYMPY and *Pint*. These packages are easily installed using the *Python* **pip** tool. Here is the line I used to install them on my system:

```
$ pip install sympy, pint
```

2.1   SymPy

We will be working through a bit of math in this article. I will show the equations in their "normal" form but I will be using a nice *Python* package called *SymPy* to generate the actual results. *SymPy* is a symbolic math engine, it manipulates *symbols* the same way you did back in your algebra class in high school. The really nice thing about *SymPy* is that it knows a lot about math, and it does not make mistakes (well, unless you do when you set things up!)

As a simple example of what *SymPy* can do, let's show a bit of that algebra. Suppose you were asked to expand $(x + y)^3$. Here is the *Python* code that uses *SymPy* to figure this out for you:

?? PythonTeX ??

Those **print** statements are strictly for use in this document, so things display nicely in the final *PDF file*. The result is this:

I bet you wish you had access to this tool back in your school days!

2.2   *Pint*

In engineering, we absolutely must pay attention to the units we use for physical properties. In the U.S. we commonly use Imperial units (inches or

feet, and ounces or pounds). Elsewhere it is more common to use Metric units (centimeters, or meters, and grams or kilograms). Indoor modelers seem to mix the two systems, using inches for lengths, and grams for weight. Failure to pay attention to all of this can lead to embarrassing failures, as NASA found out in one of their Mars expeditions when some subcontractors were using Metric units and everyone else was using Imperial units!

Fortunately, another *Python*— package can make tracking units and converting them as needed easy. The —PINT— package has a lot of power we can use here:

?? PythonTeX ??

Once again, we see some trickery to get the output to look nice in a document generated using LaTeX.

Here is the result:

Who came up with "slug" as a unit anyway? Let's convert that to the Metric system:

?? PythonTeX ??

Pint uses the Metric system internally, so we just asked it to show those "base" units:

This is really nice! No more digging out a calculator and googling for unit conversion factors!

Armed with these tools, let's take a look at the atmosphere we will be flying through next.

3 Air Properties

We will need a few basic properties of air for our calculations:

The density of air, which we looked ta briefly in the last section, is assumed to be constant in our flying sites (well, unless you fly in very tall buildings!)

$$\rho = 33.6 \frac{g}{ft^3} \tag{1}$$

The kinematic viscosity, a measure of the "air's resistance to motion is given next:

$$\nu = 15.88x \cdot 10^{-5} \frac{ft^2}{sec} \tag{2}$$

4 Symbols

AF	Aspect ratio of forward wing
AR	aspect ratio of rear lifting surface
b	wingspan
c_w	mean wing chord
c_t	mean stab chord
b_r	stab span
C_d	overall drag coefficient
C_{di}	overall induced drag coefficient
C_{dif}	induced drag coefficients of forward wing(s)
C_{dir}	induced drag coefficient of tail
C_{dp}	total profile drag coefficient
C_{dpf}	profile drag of forward wing(s)
C_{dpr}	profile drag coefficient of tail
C_{dw}	profile drag coefficient of bracing wires
ΔC_d	total drag of wing posts and bracing wire
C_f	mean aerodynamic chord of forward wing
C_l	overall lift coefficient
C_{lf}	lift coefficient of forward wing(s)
C_{lr}	lift coefficient of horizontal stab
dw	diameter of bracing wire
E	initial energy stored in rubber motor
F	emperical efficiency factor
g	grams
G	vertical gap between wings of a biplane or tandem
h	ceiling height in feet
H	dimensionless ceiling height
J	propeller advance ratio
K_m	rubber motor constant
l	tail moment arm
l_w	total length of bracing wire
n	propeller rotational speed (revolutions/second)
N	maximum turns for rubber motor
P	power required for level flight (thrust * speed)
Re_f	Reynolds number for forward wing(s)
Re_r	Reynolds number for tail
Re_w	Reynolds number for bracing wire
S	total horizontal projected area of model
S_f	total horizontal projected area of forward wing(s)
S_p	propeller disc area
S_r	horizontal projected area of tail
S_{wp}	area of wing posts projected in flight direction
t	flight time (duration)
T	propeller thrust force
V	flight speed

References

[1] M. Kintel. OpenSCAD - The Programmers Solid 3D CAD Modeller, 2021. URL https://www.openscad.org/.

[2] D. McLean. A method for predicting indoor model duration. *National Free Flight Association Symposium*, 1976.