Optimal rotor arm pivot angles for a collapsible drone.

by
Peter Burnett and Hugh Murrell

January 6, 2021

Problem Description

Consider the CAD drawing of the collapsible drone shown in figure 1 below:

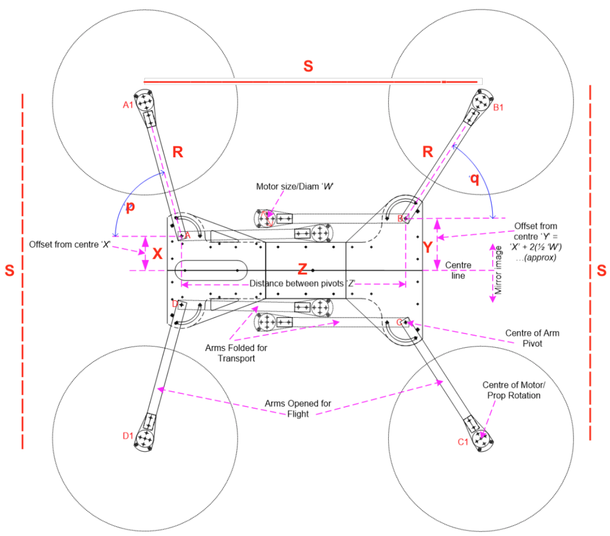

Figure 1: CAD drawing of drone with collapsible arms

In this diagram all lengths are measured in mm and the annotated lengths are design parameters for the construction of the drone:

- R the length of the rotor arm (all rotor arms have the same length)
- Z the distance between rotor arm pivot points.
- X the offset of the front pivot arm point from the horizontal axis of symmetry.
- Y the offset of the back pivot arm point from the horizontal axis of symmetry.

The problem is to determine the rotor arm angles, p and q, under the constraint that the rotor centres form a square.

Constraint Problem

Using the constraint that the rotor centres form a square with sides of some unknown length S we have:

$$\begin{aligned} S &= R\cos(p) + Z + R\cos(q) \\ S &= 2(X + R\sin(p)) \\ S &= 2(Y + R\sin(q)) \end{aligned} \tag{1}$$

The first equation in (1) is derived from an expression for the horizontal edge of length, S, whilst the last two equations are derived from expressions for the vertical front and back edges of length S.

The unknown edge length, S, can be eliminated from equations (1) yielding two non-linear equations for the angles, p and q shown in equations (2) below.

$$\begin{aligned} \cos(p) - 2\sin(p) + \cos(q) &= \frac{2X - Z}{R} \\ \sin(p) - \sin(q) &= \frac{Y - X}{R} \end{aligned} \tag{2}$$

Although it is possible to find a closed form solution for p and q the calculation requires finding the roots of a quartic polynomial which is extremely tedious as one can see from the Wikipedia entry [1].

So instead we use standard non-linear root finding technique. For example, the `scipy` python package allows us to construct the solution shown in the appendix.

To make life easier for the reader we have also coded a `javascript` solution to the problem, again using a numerical equation solver. This time we found that Martin Donk's `javascript` library, `nerdamer`, did the trick, see [2] .

To access the `javascript` solution goto:

`https://hughmurrell.github.io/DroneDesign/index.html`.

References

[1] Wikipedia Article, *Quartic function* `https://en.wikipedia.org/wiki/Quartic_function` accessed January 2021.

[2] Martin Donk, *nerdamer, Symbolic Math for Javascript*, `https://nerdamer.com/`, accessed January 2021.

Appendix

pythod code for calculating rotor arm pivot angles

```
from scipy.optimize import fsolve
from math import sin, cos, pi

def equations(vars):
    p, q = vars
    R = 225
    Z = 346
    X = 56
    Y = 84
    eq1 = cos(p) - 2*sin(p) + cos(q) - (2*X-Z)/R
    eq2 = sin(p) - sin(q) - (Y-X)/R
    return [eq1, eq2]

p, q =  fsolve(equations, (1, 1))

p = (p / pi) * 180
q = (q / pi) * 180

print(p, q)
```