Propeller Performance Optimization

Additional Notes

The aerodynamic figure of merit (FOM) for propeller design is

$$\text{FOM} = \frac{C\,T^{3/2}}{n D Q}$$

where T is the thrust (N)

D is prop diameter (m)

n is rotational speed (10^3 RPS)

Q is the torque (N m)

C is a constant.

The quantities D, n, T, Q are measured. T and Q are found experimentally as unknown functions of D, n and α, the propeller angle of attack. That is, we find experimental data vectors $(D_k, \alpha_k, n_k, T_k, Q_k)$.

within limits: $D_{min} \le D_k \le D_{max}$, $\alpha_{min} \le \alpha_k \le \alpha_{max}$, $n_{min} \le n_k \le n_{max}$.

The Goal: Determine the parameters D, α, n that produce maximum FOM for a given thrust T_0. That is, find $D^*(T_0)$, $\alpha^*(T_0)$, $n^*(T_0)$.

The optimization problem starts as the following.

$$\max \frac{T^{3/2}}{nDQ}$$

$$\text{s.t.} \quad T = T_0$$
$$D_{min} \le D \le D_{max}$$
$$\alpha_{min} \le \alpha_k \le \alpha_{max}$$
$$n_{min} \le n_k \le n_{max}$$

However, T and Q are functions.

$$\max_{D,\alpha,n} \frac{T(D,\alpha,n)^{3/2}}{nDQ(D,\alpha,n)}$$

$$\text{s.t.} \quad T(D,\alpha,n) = T_0$$
$$D_{min} \le D \le D_{max}$$
$$\alpha_{min} \le \alpha \le \alpha_{max}$$
$$n_{min} \le \alpha \le n_{max}$$

Then using the fixed thrust criterion:

$$\boxed{\begin{aligned} \min_{D,\alpha,n} \quad & nDQ(D,\alpha,n) \\ \text{s.t.} \quad & T(D,\alpha,n) = T_0 \\ & D_{min} \le D \le D_{max} \\ & \alpha_{min} \le \alpha \le \alpha_{max} \\ & n_{min} \le \alpha \le n_{max} \end{aligned}}$$

The functions $T(D,\alpha,n)$ and $Q(D,\alpha,n)$ are determined from the data as

$$a^* = \underset{a}{\operatorname{argmin}} \quad \sum_{k=1}^{m} \left(T(D_k,\alpha_k,n_k) - T_0\right)^2 ,$$

where $T(D,\alpha,n) = \sum_{j=1}^{p} a_j \, \phi_j(D,\alpha,n)$,

and $\phi_j(D,\alpha,n)$ are nonparametric functions.

As a simple example, suppose T is affine in D, α, n. Then we choose $\phi_1 = 1$, $\phi_2 = D$, $\phi_3 = \alpha$, $\phi_4 = n$.

$$T = a_1 + a_2 D + a_3 \alpha + a_4 n$$

and we expect the data to satisfy $T_k \approx a_1 + a_2 D_k + a_3 \alpha_k + a_4 n_k$ for each k. We then have the optimization problem

$$\min \quad f(a) = \sum_{k=1}^{m} \left(a_1 + a_2 D_k + a_3 \alpha_k + a_4 n_k - T_k \right)^2$$

We have an analytic solution: $\nabla_a f(a) = 0$:

$$\sum_{k=1}^{m} 2\left(a_1 + a_2 D_k + a_3 \alpha_k + a_4 n_k - T_k \right) \begin{bmatrix} 1 \\ D_k \\ \alpha_k \\ n_k \end{bmatrix} = 0$$

or,

First:

$D \leftarrow D / \max(D)$

$\alpha \leftarrow \alpha / \max(\alpha)$

$n \leftarrow n / \max(n)$

$$
\begin{bmatrix}
\sum_{k=1}^{m} 1 & \sum_{k=1}^{m} D_k & \sum_{k=1}^{m} \alpha_k & \sum_{k=1}^{m} n_k \\
\sum_{k=1}^{m} D_k & \sum_{k=1}^{m} D_k^2 & \sum_{k=1}^{m} \alpha_k D_k & \sum_{k=1}^{m} D_k n_k \\
\sum_{k=1}^{m} \alpha_k & \sum_{k=1}^{m} \alpha_k D_k & \sum_{k=1}^{m} \alpha_k^2 & \sum_{k=1}^{n} \alpha_k n_k \\
\sum_{k=1}^{m} n_k & \sum_{k=1}^{m} D_k n_k & \sum_{k=1}^{m} \alpha_k n_k & \sum_{k=1}^{m} n_k^2
\end{bmatrix}
\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}
=
\begin{bmatrix}
\sum_{k=1}^{m} T_k \\
\sum_{k=1}^{m} T_k D_k \\
\sum_{k=1}^{m} T_k \alpha_k \\
\sum_{k=1}^{m} T_k n_k
\end{bmatrix}
$$

$$A_T \quad a \quad = \quad B_T$$

$$\boxed{a = A_T^{-1} B_T}$$

Similarly $\quad Q = \sum_{j=1}^{p} b_j \phi_j(D, \alpha, n) \quad , \ldots , \quad b = A_Q^{-1} B_Q$

So,

$$\min_{D,\alpha,n} \quad nDQ$$

$$\text{s.t.} \quad T = T_0$$

$$D_{min} \le D \le D_{max}$$

$$\alpha_{min} \le \alpha \le \alpha_{max}$$

$$n_{min} \le n \le n_{max}$$

where:

$$T = \sum_{j=1}^{P} a_j \phi_j(D,\alpha,n)$$

$$Q = \sum_{j=1}^{P} b_j \phi_j(D,\alpha,n)$$

TASK: Select a family of smooth functions that render a good fit to the data. (T and Q).

TASK: Solve the FOM problem using the Augmented Lagrangian method with seven equality constraints, three design variables (D,α,n) and six slack variables. let $T_0 =$ ___ .

TASK: Tabulate $FOM(T_0)$ for ___ $\le T_0 \le$ ___, along with corresponding values for D,α,n.

Order N polynomial fit in $\mathbb{R}^3$.

$$T(x,y,z) = \sum_{\substack{l+m+n \le N \\ l,m,n \ge 0}} a_{lmn}\, x^l y^m z^n$$

$N=0 \qquad T = a_{000}$

$N=1 \qquad T = a_{000} + a_{100}x + a_{010}y + a_{001}z$

$$= a_{000} + [x\ y\ z]\begin{bmatrix} a\_ \\ a\_ \\ a\_ \end{bmatrix}$$

Building an array of all possible coefficients:

```
ex = [ ];
for j = 0:N
    for k = 0:j
        ex = [ex ; [(j-k)*ones(k+1,1) (k:-1:0)' (0:1:k)' ]]
    end
end
```

terms R: $N=0 \rightarrow R=1$

$N=1 \rightarrow R = 1+3 = 4$

$N=2 \rightarrow R = 1+3+6 = 10$

$N=3 \rightarrow R = 20$

$N=4 \rightarrow R = 35 \qquad N=5 \rightarrow R = 56$

$$R = \frac{1}{6}(N+1)(N+2)(N+3)$$

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

$$T(x,y,z) = \sum_{j=1}^{R} a_j \, \phi_j(x,y,z) \qquad \text{each } \phi_j = x^l y^m z^n \text{ for some } l,m,n.$$

$$Q(x,y,z) = \sum_{j=1}^{R} b_j \, \phi_j(x,y,z)$$

x, y, z, T, Q are data vectors

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Code for finding fit coefficients a_k:

```
AT = zeros(R,R)
BT = zeros(R,0)
for r = 1:m
    temp = x.^ex(r,1) .* y.^ex(r,2) .* z.^ex(r,3)          x.^ex(r,:)'
    BT(r) = sum (Tk .* temp)
    for c = 1:m
        p = ex(r,:) + ex(c,:)
        AT(r,c) = sum ( x.^p(1) .* y.^p(2) .* z.^p(3) )
    end
end
a = AT \ BT ;
```

Consider the two key functions:

[f, g] = PropObjective (x, par)

[c, h] = PropConstraints (x, par)

Objective computation (for $f = nDQ$)

```
function [f,g] = PropObj(x, p)
 Q = 0;
 m = length(p.b);
 for r = 1:m
     Q = Q + p.b(r) * prod( x.^p.pw(r,:) );
 end

 f = x(1)*x(3)*Q;

 (Also compute g)

 return
```

$$x = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad p.b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_R \end{bmatrix}$$

$$p.pw = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \\ & \vdots & \end{bmatrix} = ex$$

$$Q = \text{prod}\left(x\,.\!\wedge (p.pw)' \right) * p.b$$

— Gradient Computations —

$f = nDQ(D, \alpha, n)$

let $x_1 = D$ $x_2 = \alpha$ $x_3 = n$ then

$$f = x_1 x_3 \sum_{j=1}^{R} b_j \phi_j(x)$$

$$\frac{\partial f}{\partial x_1} = x_3 \sum_{j=1}^{R} b_j \phi_j(x) + x_1 x_3 \sum_{j=1}^{R} b_j \frac{\partial \phi_j}{\partial x_1}$$

$$\frac{\partial f}{\partial x_2} = x_1 x_3 \sum_{j=1}^{R} b_j \frac{\partial \phi_j}{\partial x_2}$$

$$\frac{\partial f}{\partial x_3} = x_1 \sum_{j=1}^{R} b_j \phi_j(x) + x_1 x_3 \sum_{j=1}^{R} b_j \frac{\partial \phi_j}{\partial x_3}$$

$$\boxed{\nabla_y f = 0_{6\times 1}}$$

$$\nabla_x f = \begin{bmatrix} x_3 Q(x) \\ 0 \\ x_1 Q(x) \end{bmatrix} + x_1 x_3 \nabla_x Q(x)$$

$$\boxed{\nabla_x f = \begin{bmatrix} x_3 \\ 0 \\ x_1 \end{bmatrix} Q(x) + x_1 x_3 \sum_{j=1}^{R} b_j \nabla_x \phi_j}$$

$$C_7(x) = T(x) - T_0 \qquad \nabla_x C_7(x) = \nabla_x T(x) = \sum_{j=1}^{R} a_j \nabla_x \phi_j(x)$$

$$\nabla_y C_7(x) = 0$$

$$C_1(x) = D_{max} - x_1 - y_1^2 = 0 \qquad \nabla C_1(x) = \begin{bmatrix} -1 & 0 & 0 & -2y_1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_2(x) = \alpha_{max} - x_2 - y_2^2 = 0 \qquad \nabla C_2(x) = \begin{bmatrix} 0 & -1 & 0 & 0 & -2y_2 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_3(x) = n_{max} - x_3 - y_3^2 = 0 \qquad \nabla C_3(x) = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & -2y_3 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_4(x) = x - D_{min} - y_4^2 = 0 \qquad \nabla C_4(x) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -2y_4 & 0 & 0 \end{bmatrix}^T$$

$$\boxed{\nabla C(x) = \left[\begin{array}{cc|c} -I_3 & I_3 & \nabla_x T(x) \\ \multicolumn{2}{c|}{-2Y} & 0_{6\times 1} \end{array}\right], \qquad \begin{aligned} \nabla_x T(x) &= \sum_{j=1}^{R} a_j \nabla_x \phi_j(x) \\ Y &= \operatorname{diag}(y) \end{aligned}}$$

Then build the Lagrangran function

```
function [F, G] = ALOBJ(x, par)
 [f, g] = PropObjective(x, par)
 [c, h] = PropConstraints(x, par)
  F = f - c' * par.lambda + (0.5)*par.mu * ( c' * c );
 G = g - h * ( par.lambda - par.mu * c )
return
```

$$F = f(x) - c^T\lambda + \frac{1}{2}\mu c^T c$$

$$\nabla F = \nabla f - \nabla c(\lambda - \mu c)$$