Documentation of Matlab function: autogiro performances.m

Authors:
Filippo Borriello
Alessandro Furno

Contents

1 Autogiro performances **2**
1.1 Analytical procedure 2

2 Code listing **2**

3 Test case **7**
3.1 PCA-2 7
3.2 MagniM24 9

1 Autogiro performances

This function is an implementation of the procedure shown in the references (par. 8.10), particularised for autogiros and gyroplanes. For a given aircraft, at least basic data such as weight, information on the rotor, aerodynamics of the profile are to be known, advance ratio and intake ratio are to be set by the user. In order to achieve the autorotation conditions, the function solve the equations that are in the references using the Matlab Symbolic Toolbox. In particular, given the intake ratio it is possible to solve them in respect to the collective pitch θ, and then calculate the thrust coefficient T_c, angular velocity Ω, angle of attack α and the freestream velocity V_∞.

1.1 Analytical procedure

The procedure that is applied is shown as follow [1] (it is also available in the references).

- Assign the intake ratio λ
- Set a flight conditions matrix, basically assigning also μ.
- The equation $Pc = \lambda T_c + Q_{c0} + \mu H_{c0} - \mu H_c = 0$ [3], which represents the autorotation condition, once solved in respect to the collective pitch θ, since $\theta = \theta(\lambda)$ it is possible to evaluate all the necessary coefficient to estimate the performances especially T_c, H_c, α and V_∞, those of most interest.
- It is now possible to calculate the fundamental unknown of the autorotation, Ω.
- Knowing this quantities it is easy to calculate the total required power $P_n = P_i + P_{rotor} + P_{fus}$.

2 Code listing

In the following pages it is presented the complete function that is needed to evaluate the performances of an autogiro in forward flight. Should be clear that this problem can only be solved only for some flight condition of the autogiro, in particular the user should pay attention to set the right value for μ, because too low value could lead to confusion in interpreting the results. It can be used to solve some test case that follows, achieving curves as required power, angle of attack, collective pitch and others . The function needs 12 input value that are:

- W, weight $[N]$.

- ρ, air density $[kg/m^3]$.
- c, chord length $[m]$.
- Cl_α.
- R, rotor radius $[m]$.
- θ_{tw}, blade twist $[deg]$.
- N, blade numbers.
- f_d, equivalent wet area $[m^2]$.
- A, rotor area $[m^2]$.
- C_{D_0}, drag coefficient.
- γ, Lock number.
- λ, intake ratio.
- μ, advantace ratio.
- P_D, aviable power.

and returns 7 output values:

- P_n, required power $[kW]$.
- α, angle of attack $[deg]$.
- ω, angular velocity $[1/s]$.
- θ_0, collective pitch $[deg]$.
- V_0, freestream velocity $[m/s]$.
- P_i, induced power $[kW]$.
- rs, rateo di salita $[m/s]$.

```
function [pnec, alpha, omega,THETA_0F,v0, pi, rs ]= AUTOGIRO_FUN(w,ro,co,cla,r,tetatw,n,fd,A,cd0,lock,lambda,mu,Pmax)
sigma=n*co/(3.14*r);
syms teta0 omega [1 50]
for i=1:50
    % thrust coefficient
Tc(i) = 0.5*sigma*cla*(teta0(i)/3*(1+3/2*mu(i)^2)+tetatw/4*(1+mu(i)^2)-lambda(i)/2);

 % parasitic torque coefficient
Qc0(i) = sigma*cd0/8*(1+mu(i)^2);

% flapping coefficients expressions
b_0(i) = lock*(teta0(i)/8*(1+mu(i)^2)+tetatw/10*(1+5/6*mu(i)^2)-lambda(i)/6);
b_1c(i) = -2*mu(i)*(4/3*teta0(i)+tetatw-lambda(i))/(1-0.5*mu(i)^2);
b_1s(i) = -4/3*mu(i)*b_0(i)/(1+0.5*mu(i)^2);

 % induced rotor drag coefficient
H_ci(i) = 0.5*sigma*cla*(teta0(i)*(-1/3*b_1c(i)+0.5*mu(i)*lambda(i))...
+tetatw*(-1/4*b_1c(i)+1/4*mu(i)*lambda(i))+3/4*lambda(i)*b_1c(i)...
+1/6*b_0(i)*b_1s(i)+1/4*mu(i)*(b_0(i)^2+b_1c(i)^2));

% parasitic rotor drag coefficient
H_C0(i) = sigma*cd0*mu(i)/4;

%from the definitiion of autorotaion
Pc(i) = lambda(i)*Tc(i)+Qc0(i)-mu(i)*H_ci(i);

% we choose the second solotuion
THETA_0(i,:) = vpasolve(Pc(i)==0,teta0(i));

THETA_0F(i)=THETA_0(i,2);

% It is possible now to find Tc=Tc(mu),Hc=Hc(mu),alfa=alfa(mu)
```

```

TC(i) = 0.5*sigma*cla*(THETA_0F(i)/3*(1+3/2*mu(i)^2)...
+tetatw/4*(1+mu(i)^2) -lambda(i)/2);

B_0(i) = lock*(THETA_0F(i)/8*(1+mu(i)^2)+tetatw/10*(1+5/6*mu(i)^2)...
-lambda(i)/6);

B_1c(i)= -2*mu(i)*(4/3*THETA_0F(i)+tetatw-lambda(i))/(1-0.5*mu(i)^2);

B_1s_lambda(i) = -4/3*mu(i)*B_0(i)/(1+0.5*mu(i)^2);

H_Ci(i)= 0.5*sigma*cla*(THETA_0F(i)*(-1/3*B_1c(i)...
+0.5*mu(i)*lambda(i))+tetatw*(-1/4*B_1c(i)+1/4*mu(i)*lambda(i))...
+3/4*lambda(i)*B_1c(i)+1/6*B_0(i)*B_1s_lambda(i)...
+1/4*mu(i)*(B_0(i)^2+B_1c(i)^2));

 % induced intake ratio
lambda_i(i) = 1.2*TC(i)/(2*sqrt(mu(i)^2+lambda(i)^2));

 % angle of attack
alpha(i) = atan(1/mu(i)*(lambda(i)-TC(i)/(2*sqrt(mu(i)^2+lambda(i)^2))));

 % angular velocity of the rotor
omega(i) =sqrt(w/(cos(alpha(i))*TC(i)*ro*3.14*r^4));

end

%Evaluation of the output
for i=1:50
lambda_i(i)=vpa(lambda_i(i));
 % Thrust coefficient
TC(i)=vpa(TC(i));
% [rad/s] Angular velocity
omega(i)=vpa(omega(i));
% Rotor drag coefficient
H_Ci(i)=vpa(H_Ci(i));
```

```
H_C0(i)=vpa(H_C0(i));
HC(i)=(H_Ci(i)+H_C0(i));
% Collective pitch
THETA_0F(i)=vpa(THETA_0F(i))/0.0174;
alpha(i)=vpa(alpha(i))/0.0174;
% freestream velocity
v0(i)=mu(i)*omega(i)*r;
 % induced power coefficient
pi(i)=TC(i)*lambda_i(i)*ro*omega(i)^3*r^5*3.14;
% fuselage parastic coefficient
pfus(i)=0.5*fd*v0(i)^3*A;
prot(i)=mu(i)*(H_Ci(i)+H_C0(i))*ro*3.14*r^5*omega(i)^3;
% total required power
pnec(i)=((pi(i)+prot(i))+pfus(i))/0.7;
vprot(i)=mu(i)*tan(alpha(i)*0.01744);
% climb rate
rs(i)=(Pmax-pnec(i))/w*3.28*60;
end
end
```

3 Test case

3.1 PCA-2

This first test case shows the performance evaluation of an autogiro Pitcarin PCA-2. The user should add the graphics part, that it is omitted in this part, in order to see and study the curves that he needs. In particular it is necessary to show the curves of the required power over the free streamvelocity. This allow to validate the function obtaining reasonable value, especially for the maximum velocity that match with the original maximum velocity of the aircraft.

```
clear all; close all; clc;
% Test case
%% Pitcairn PCA-2 autogiro in forward flight

% input

W = 1563*9.81;                  % MTOW [ N ]
Pd = 250000;                    % Maximum aviable power [ W ]
N = 4;                          % blade numbers
R = 6.86;                       % rotorblade radius [m]
A = 3.14*R^2;                   % rotor area [m^2]
c=0.63;                         % chord length [m]
Cl_a=5.4;
tetatw=-10*0.01744;             % blade twisting (linear variation is assumed) [  ]
fd=0.01;                        % equivalent wet area [m^2]
rho = 1.225;                    % air density [kg/m^3]
cd0=0.010;                      % avarage drag coefficient
lock=8;                         % Lock's number

lambda(1:50)=-0.01;
mu=linspace(0.03,0.5,50);
%% Function output
[Pnec, alpha, omega,THETA_OF,v0,pi,rs ] = AUTOGIRO_FUN(W,rho,c,Cl_a,R,tetatw,N,fd,A,cd0,lock,lambda,mu,Pd);
```

Plotting the required power as a function of the freestream velocity. This lead to underline that the maximum velocity is about $190 km/h$ that confirm the real value stand on the specification of the autogiro.

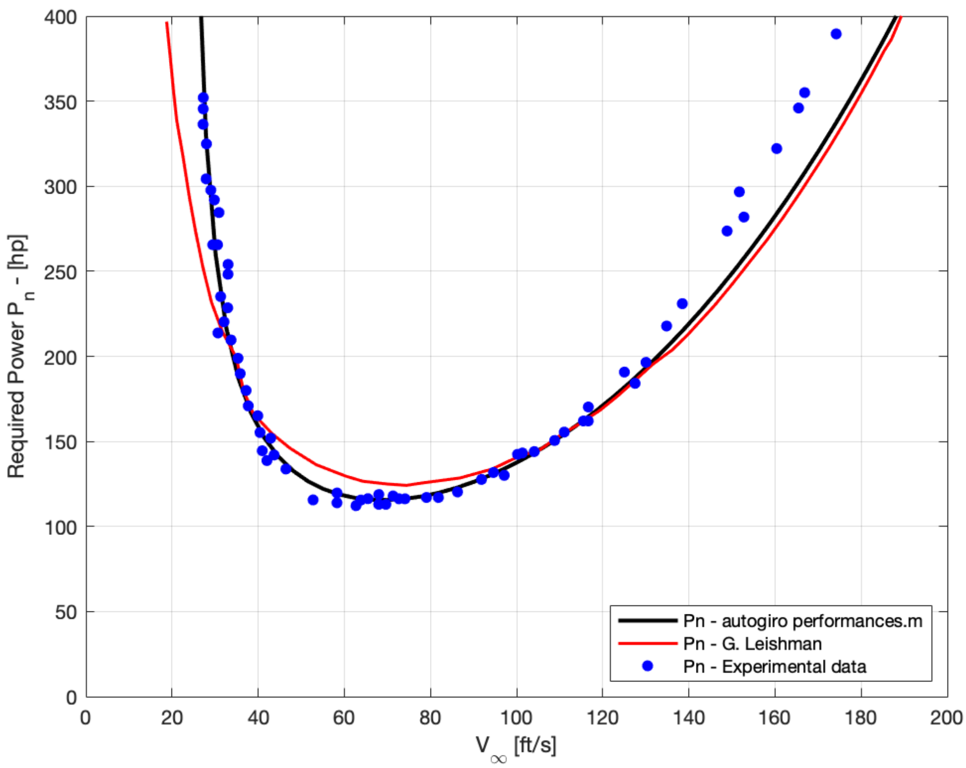

Figure 1: Validation of the function by comparing the power derived from the Matlab function with that present in the reference [2] .

In the figure above we can see the black curve (derived from our function), superimposed on the required power found in the literature, in which it is possible to highlight that for low speeds the power coming from our function corresponds in such a good way to the experimental value, while for high speed follows the curves of the Leisman's work [2].

3.2 MagniM24

A second test case is carried out on the Magni M24.The output is shown below the code and is quite reasonable according to the data that can be find on internet. Unfortunately we were not able to find the literature to validate the function also on this aircraft.

```
clear all; close all; clc;
% Test case
%% Magni M24 autogiro in forward flight

% input

W = 550*9.81;                     % MTOW [ N ]
Pd = 85000;                       % Maximum aviable power [ W ]
N = 2;                            % blade numbers
R = 4.1;                          % rotorblade radius [m]
A = pi*R^2;                       % rotor area [m^2]
c=0.3;                            % chord length [m]
Cl_a=5.4;
tetatw=-10*0.01744;               % blade twisting (linear variation is assumed) [  ]
fd=0.01;                          % equivalent wet area [m^2]
rho = 1.225;                      % air density [kg/m^3]
cd0=0.010;                        % avarage drag coefficient
lock=8;                           % Lock's number
lambda(1:50)=-0.01;
mu=linspace(0.03,0.5,50);
%% Function output
[Pnec, alpha, omega,THETA_0F,v0,pi,rs ] = AUTOGIRO_FUN(W,rho,c,Cl_a,R,tetatw,N,fd,A,cd0,lock,lambda,mu,Pd);
```

The output is shown below the code and is quite reasonable according to the data that can be find on internet. Unfortunately we were not able to find the literature to validate the function also on this aircraft.

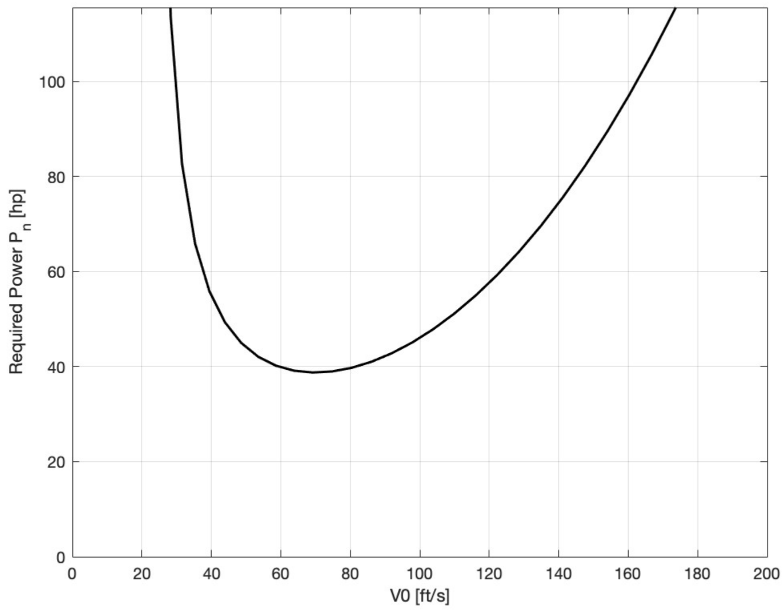

Figure 2: Prediction of power required for Magni M24

References

[1] R. Tognaccini, *Appunti di Aerodinamica dell'ala Rotante* 2022-2023.

[2] G. Leishman, *Principles of helicopter aerodynamics* , 2006, Cambridge University Press.

[3] G. Di Giorgio,*Lezioni integrative di aerodinamica dell'elicottero*, 2022-2023.

[4] H. Glauert,*A general theory of the autogyro*, 1926.