BladeSection_AngleOfAttack.m user guide

Manganiello Serena, Massa Michele

Contents

1 Introduction

The blade section angle of attack can be determined as:

$$\alpha_e = \theta - \frac{u_P}{u_T} \tag{1}$$

where θ is the blade pitch, u_P is the air velocity of the blade section, perpendicular to the disk plane, and u_T is the air velocity of the blade section, tangent to the disk plane. In particular, u_P and u_T can be found as:

$$u_P = \lambda + \bar{r}\frac{\dot{\beta}}{\Omega} + \beta\mu\cos\psi \tag{2}$$

$$u_T = \bar{r} + \mu\sin\psi \tag{3}$$

2 Inputs and outputs

The function has the following inputs:

- λ : rotor inflow ratio.
- $\bar{r}$: non-dimensional coordinate along the rotor blade.
- β : blade flap angle.
- $\dot{\beta}$: blade flap angle's derivative.
- μ : rotor advance ratio.
- ψ : azimuth angle of the blade.
- θ : blade pitch.
- $\alpha_{stall_{up}}$: positive stall angle of attack.
- $\alpha_{stall_{lo}}$: negative stall angle of attack.

and the following output:

- α_e : blade section angle of attack.

3 Function description

Mesh grids are created for the visualization of the hub and the blade section angle of attack during the blade rotation.

```
% Creation of the mesh grid.
[r_2d,psi_2d] = meshgrid(linspace(0,r_segn(1),...
                    length(r_segn)),psi);
x_hub = r_2d.*cos(psi_2d-pi/2);
y_hub = r_2d.*sin(psi_2d-pi/2);

[r_2d,psi_2d] = meshgrid(r_segn,psi);
x = r_2d.*cos(psi_2d-pi/2);
y = r_2d.*sin(psi_2d-pi/2);
```

Then, the blade section angle of attack is evaluated from the equation (1).

```
% Variables' initialization.
u_P = zeros(length(r_segn),length(psi));
% air velocity of the blade section, perpendicular to the disk
% plane.
u_T = zeros(length(r_segn),length(psi));
% air velocity of the blade section, tangent to the disk plane.
u_R = zeros(length(r_segn),length(psi));
% radial air velocity of blade section.
phi = zeros(length(r_segn),length(psi));
% section inflow angle.
alpha_e = zeros(length(r_segn),length(psi));
% blade section angle of attack.

for i = 1:length(psi)
    u_P(:,i) = lambda + r_segn'.*dbeta(i) + ...
    beta(i).*mu.*cos(psi(i));
    u_T(:,i) = r_segn' + mu.*sin(psi(i));
    u_R(:,i) = mu*cos(psi(i));
    phi(:,i) = u_P(:,i)./u_T(:,i);
    alpha_e(:,i) = theta' - phi(:,i);
end
```

In order to better visualize the blade section angle of attack map, the regions of the blade are distinguished in stalled and non-stalled, as the stall angles of the blade section are provided in input.

```
stall = zeros(length(r_segn),length(psi)); % stalled region of
                                           %the rotor blade.
non_stall = NaN(length(r_segn),length(psi)); % non-stalled
                                  %region of the rotor blade.
for iii = 1:length(r_segn)
    for jjj = 1:length(psi)
        if alpha_e(iii,jjj) -...
         convang(alpha_stall_up,'deg','rad') > 0  ...
         alpha_e(iii,jjj) -...
         convang(alpha_stall_lo,'deg','rad') < 0
            stall(iii,jjj) = alpha_e(iii,jjj);
        else
            non_stall(iii,jjj) = alpha_e(iii,jjj);
        end
    end
end
```

A plot is made to show the blade section angle of attack of the non-stalled region.

```
% Plot.
figure(1)
set(figure(1),'Color','w');
clf
fill(x_hub,y_hub,'k'); hold on; axis equal; axis off; grid off
text(r_segn(end-10),r_segn(end),['\mu = ',num2str(mu)],...
'Color','r','FontSize',12);
contourf(x,y,convang(non_stall','rad','deg'),20,'ShowText',...
'on'); drawnow; axis equal;
end
```

4 Test Case

A test case [1] was performed in order to validate the function. The following data were assumed.

```
%% Data.
Vinf_vec = 1:1:70; % Asymptotic velocity
Omega = 220/60*2*pi; % Angular velocity of the rotor blade
R = 8.50; % Blade radius
A = pi*R^2; % Rotor area
z0 = 2e3; % Altitude
N = 4; % Number of blades
c = 0.385; % Blade chord
sigma = N*c/(pi*R); % Rotor solidity
```

```
Cl_alpha = 2*pi; % Rate of the lift coefficient curve
theta_tw = -0.0799; % Linear twist rate
X = 0; % Climb angle
f_A = 0.015; f = f_A*A; % Equivalent drag area of helicopter
                        % fuselage and hub
Lock = 8; % Lock number
r_segn = linspace(0.15,1,100); % Nondimensional coordinate
                               % along the rotor blade
psi = linspace(0,2*pi,100); % Azimuth angle of the blade
alpha_stall_up = 20; % Positive stall angle
alpha_stall_lo = -10; % Negative stall angle
```

The inputs of the function are determined through the following procedure.

```
%% Calculation.
for i = 1:length(Vinf_vec)
    Vinf = Vinf_vec(i);

    [Tc,Hc,Yc,Qc,Pc,alpha_inf,lambda] =...
    Ndim_Coeff_Articulated_Rotor(Vinf,z0,Lock,f,X);
    mu = Vinf*cos(convang(alpha_inf,'deg','rad'))/(Omega*R);

    theta0 = 3/(1+3/2*mu^2)*(2*Tc/(sigma*Cl_alpha) -...
             theta_tw/4*(1+mu^2) + lambda/2);

    beta0 =Lock*(theta0/8*(1+mu^2)+theta_tw/10*(1+5/6*mu^2)-...
            lambda/6);
    beta1c = -2*mu*(4/3*theta0+theta_tw-lambda)/(1-mu^2/2);
    beta1s = -4/3*mu*beta0/(1+mu^2/2);

    beta = beta0 + beta1c*cos(psi) + beta1s*sin(psi);
    dbeta = -beta1c*sin(psi) + beta1s*cos(psi);

    theta = theta0 + theta_tw*r_segn;

    alpha_stall_up = 20;
    alpha_stall_lo = -10;

    alpha_eff = BladeSection_AngleOfAttack...
               (lambda,r_segn,beta,dbeta,mu,psi,theta,...
                alpha_stall_up,alpha_stall_lo);

end
```

Note that the data assumed must be coherent with the data imposed in the function *Ndim_Coeff_Articulated_Rotor*.

The results are shown for $V_\infty = 20$ m/s.

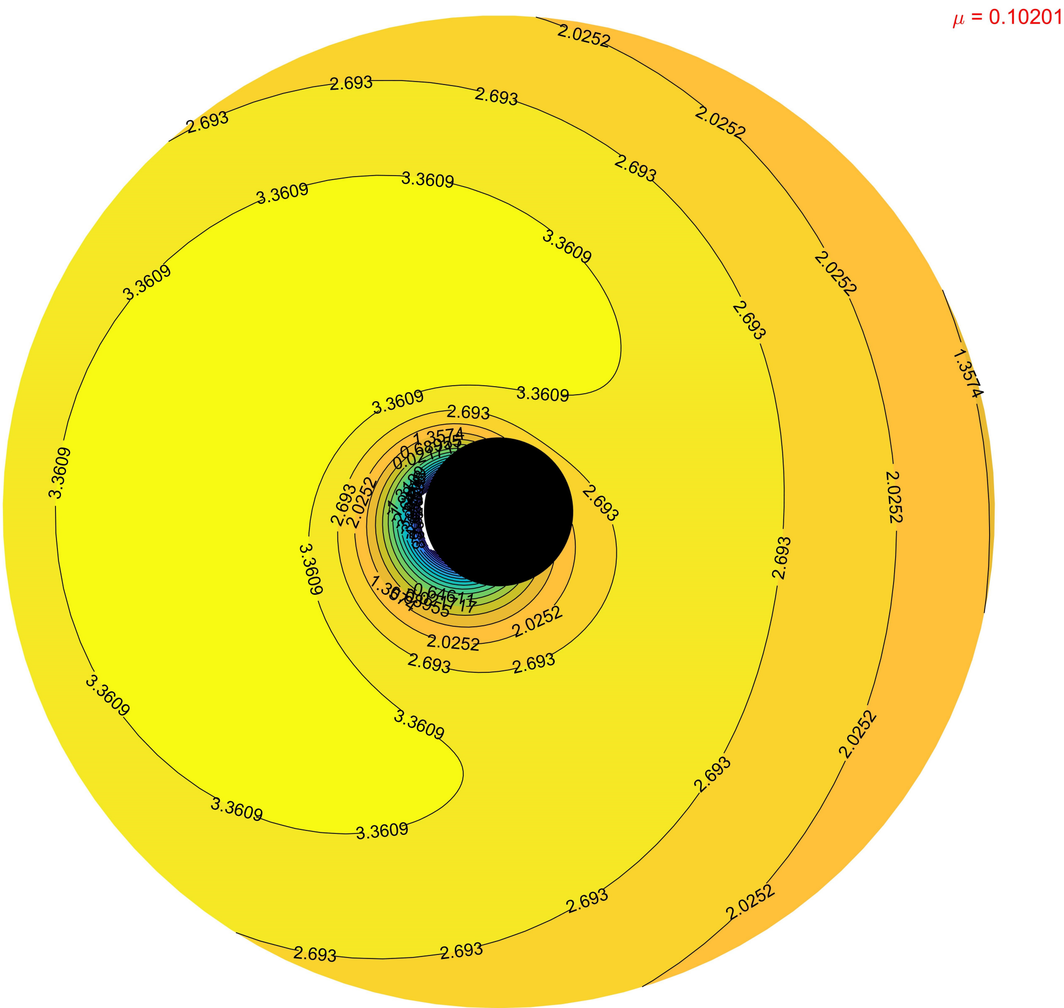

Figure 1: Evolution of the blade section angle of attack for $V_\infty = 20$ m/s.

References

[1] Lezioni di Aerodinamica dell'Ala Rotante, Tognaccini Renato, 2019, pp. 109-112

[2] Helicopter Theory, Johnson Wayne, 1980, pp. 722-726