BladeSection_AngleOfAttack.m user guide

Manganiello Serena, Massa Michele, Gianluca Porpora

Contents

1 Introduction

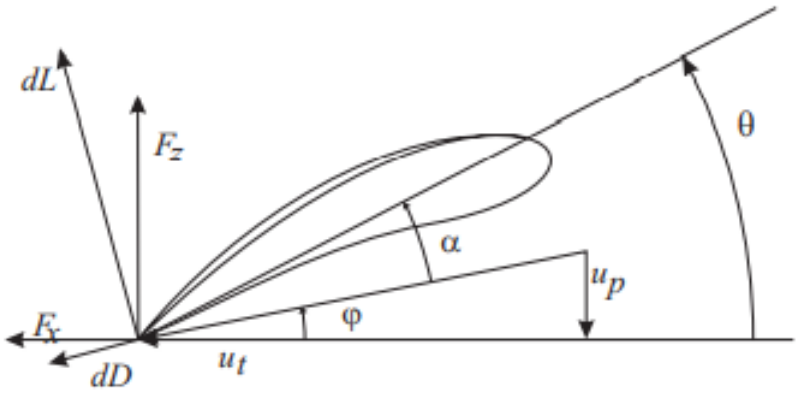

Figure 1: Aerodynamic forces and velocity triangle for a blade section element of a rotor in forward flight [1].

The function evaluates the effective angle of attack and Mach distribution of blade sections for a know helicopter. Once the articulated rotor problem is solved, the blade section angle of attack can be determined as:

$$\alpha_{eff}(\overline{r},\psi) = \theta(\overline{r},\psi) - \arctan\frac{\lambda + \overline{r}\frac{\dot{\beta}}{\Omega} + \beta\mu\cos\psi}{\overline{r} + \mu\sin\psi} \tag{1}$$

$\forall\overline{r} \in [\overline{r}_{hub}, 1], \psi \in [0, 2\pi]$, where θ is the blade pitch. In reference to figure 1, $u_P = \lambda + \overline{r}\frac{\dot{\beta}}{\Omega} + \beta\mu\cos\psi$ is the air velocity of the blade section perpendicular to the disk plane, and $u_T = \overline{r} + \mu\sin\psi$ is the air velocity of the blade section tangent to the disk plane.
Effective Mach distribution can be computed as:

$$M_{eff}(\overline{r},\psi) = \frac{\Omega R\sqrt{(\lambda + \overline{r}\frac{\dot{\beta}}{\Omega} + \beta\mu\cos\psi)^2 + (\overline{r} + \mu\sin\psi)^2}}{a_\infty} \tag{2}$$

Fuction also shows the stalled regions of the disk plane and warns user by an on-screen warning message if at least one blade section is stalling. Beware that only 2D static stall is computed and that no dynamic stall calculations are implemented.

2 Inputs and outputs

Function accepts the following inputs:

- $\overline{r}$: non-dimensional coordinate along the rotor blade.
- ψ : azimuth angle of the blade.
- μ : rotor advance ratio computed as $\frac{V_\infty \cos \alpha_{TPP}}{\Omega R}$.
- α_{TPP} : angle beetween V_∞ and the plane described by blade tips [deg].
- λ : rotor inflow ratio.
- β : blade flap angle [rad].
- $\dot{\beta}$: blade flap angle derivative.
- θ : blade pitch [rad].
- H : altitude [mt].
- ΩR : tip speed [mt/sec].
- $\alpha_{stall_{up}}$: positive stall angle of attack.
- $\alpha_{stall_{low}}$: negative stall angle of attack.
- color : flag for graphics options. Commands are 'on' and 'off'.

The function has the following outputs:

- α_{eff} : blade section effective angle of attack.
- M_{eff} : Mach number distribution on rotor disk.

3 Function description

Blade section angle of attack and Mach number distributions on disk rotor are evaluated from the equations (1),(2).

```
% 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.
Meff = zeros(length(r_segn),length(psi));       % Effective Mach
% Sound velocity calculation
[~,sound_speed,~,~] = atmosisa(h);
% Compute and distinguish stalled stations and inverse flow region
```

```
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);
    Meff(:,i) = ((sqrt(u_P(:,i).^2 + u_T(:,i).^2)*tip_speed)/sound_speed);
end
```

In order to better visualize the blade section angle of attack map, the regions of the blade are distinguished in stalled or inverse flow and non-stalled, as the stall angles of the blade section are provided in input.
Then positive stalling sections and computed and if there is at least one stalled profile, a warning message is showns. Finally max values for angle of attack and Mach number are evaluated.

```
inv_stall = zeros(length(r_segn),length(psi)); % inverse flow region and stalled stations
alpha_eff = NaN(length(r_segn),length(psi));   % non-stalled region of the rotor blade.
stall = NaN(length(r_segn),length(psi));       % positive stall region
for i = 1:length(r_segn)
    for j = 1:length(psi)
        if (alpha_e(i,j) - convang(alpha_stall_up,'deg','rad') > 0 && ...
                alpha_e(i,j) - convang(alpha_stall_up,'deg','rad') < 1)
            stall(i,j) = alpha_e(i,j);
        elseif (alpha_e(i,j) - convang(alpha_stall_up,'deg','rad') > 0 || ...
                alpha_e(i,j) - convang(alpha_stall_lo,'deg','rad') < 0)
            inv_stall(i,j) = alpha_e(i,j);
        else
            alpha_eff(i,j) = alpha_e(i,j);
        end
    end
end

% Compute first stalled sections and shows on screen warning
[ir,ic] = find((alpha_e(:,:) - convang(alpha_stall_up,'deg','rad') > 0) & ...
    (alpha_e(:,:) - convang(alpha_stall_up,'deg','rad') < convang(0.5,'deg','rad')));
if (~isempty(ir))
    warning('Rotor is stalling!')
end
% Compute max values for angles and Mach number
alpha_max = round(convang(max(alpha_eff,[],'all'),'rad','deg'),2);
Meff_max = round(max(Meff,[],'all'),2);
```

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

```
% Inizializing graphics' variables
mu = mu/cos(convang(alpha_tpp,'deg','rad')); % Dimensionless flight
    speed
% 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);
y = r_2d.*sin(psi_2d);
```

Results are plotted on a polar plot grid and depends on the color flag. If color is 'on', colormap of angle of attack and Mach are shown. If color is 'off' then results are plotted by black isolines. If the rotor is stalling, angle of attack isolines or colormap are plotted and stalled region is described by markers.
Results are referred to the parameter $\mu = \frac{V_\infty}{\Omega R}$.

```
switch color
    case 'on'
        ifig = 1; % Indexing for figure
        % Mach colormap
        figure(ifig)
        set(figure(ifig),'Color','w');
        clf
        h = polar(x,y);
        hold on;
        fill(x_hub,y_hub,'k'); hold on; axis equal; axis off; grid
    off
        str = {'M_{e}(r,\psi)',['\mu = ' num2str(mu)], ...
            ['M_{e_{max}} = ' num2str(Meff_max)]};
        text(r_segn(end-10),r_segn(end-2),str,'Color','k','FontSize
    ',10);
        contourf(x,y,Meff',40);
        axis equal; grid on; colormap jet;
        cb = colorbar('westoutside');
        position = cb.Position;
        cb.Position(1) = 0.55*position(1);
        cb.Position(2) = 1.7*position(2);
        cb.Position(3) = 0.5*position(3);
        cb.Position(4) = 0.8*position(4);
        cb.Label.String = ('M_e');
        cb.Label.FontSize = (10);
        set(h,'Visible','off')
        axis off; axis image
        view([90 90])
        ifig = ifig + 1;
        if (isempty(ir))
            % Alpha colormap
            figure(ifig)
            set(figure(ifig),'Color','w');
            clf
            h = polar(x,y);
            hold on;
            fill(x_hub,y_hub,'w'); hold on; axis equal; axis off;
    grid off
            str = {'\alpha_{e}(r,\psi)',['\mu = ' num2str(mu)], ...
```

```
                ['\alpha_{e_{max}} = ' num2str(alpha_max) '  ']};
            text(r_segn(end-10),r_segn(end-2),str,'Color','k','FontSize',10);
            contourf(x,y,convang(alpha_eff','rad','deg'),40);
            axis equal; grid on;
            colormap jet;
            cb = colorbar('westoutside');
            position = cb.Position;
            cb.Position(1) = 0.55*position(1);
            cb.Position(2) = 1.7*position(2);
            cb.Position(3) = 0.5*position(3);
            cb.Position(4) = 0.8*position(4);
            cb.Label.String = ('\alpha_e(  )');
            cb.Label.FontSize = (10);
            set(h,'Visible','off')
            axis off; axis image
            view([90 90])
        else
            % Stalled sections with alpha color contour
            figure(ifig)
            set(figure(ifig),'Color','w');
            clf
            polar(psi(ic),r_segn(ir),'k*');
            hold on
            fill(x_hub,y_hub,'k'); hold on; axis equal; axis off; grid off
            str = {'\alpha_{e}(r,\psi)','* Stall',['\mu = ' num2str(mu)]};
            text(r_segn(end-10),r_segn(end-2),str,'Color','k','FontSize',10);
            contourf(x,y,convang(alpha_eff','rad','deg'),40);
            axis equal; grid on; axis off; axis image; colormap jet;
            cb = colorbar('westoutside');
            position = cb.Position;
            cb.Position(1) = 0.55*position(1);
            cb.Position(2) = 1.7*position(2);
            cb.Position(3) = 0.5*position(3);
            cb.Position(4) = 0.8*position(4);
            cb.Label.String = ('\alpha_e(  )');
            cb.Label.FontSize = (10);
            view([90 90])
        end
     case 'off'
        ifig = 1; % Indexing for figure
        % Mach Map
        figure(ifig)
        set(figure(ifig),'Color','w');
        clf
        h = polar(x,y);
        hold on;
        fill(x_hub,y_hub,'k'); hold on; axis equal; axis off; grid off
        str = {'M_{e}(r,\psi)',['\mu = ' num2str(mu)], ...
            ['M_{e_{max}} = ' num2str(Meff_max)]};
        text(r_segn(end-10),r_segn(end-2),str,'Color','k','FontSize',10);
        contour(x,y,Meff','k','ShowText','on');
```

```
88          axis equal; grid on;
89          set(h,'Visible','off')
90          axis off; axis image
91          view([90 90])
92          ifig = ifig + 1;
93         if (isempty(ir))
94              % Alpha Map
95              figure(ifig)
96              set(figure(ifig),'Color','w');
97              clf
98              h = polar(x,y);
99              hold on;
100             fill(x_hub,y_hub,'k'); hold on; axis equal; axis off;
    grid off
101             str = {'\alpha_{e}(r,\psi)',['\mu = ' num2str(mu)], ...
102                 ['\alpha_{e_{max}} = ' num2str(alpha_max) '  ']};
103             text(r_segn(end-10),r_segn(end-2),str,'Color','k','
    FontSize',10);
104             contour(x,y,convang(alpha_eff','rad','deg'),'k','
    ShowText','on');
105             axis equal; grid on;
106             set(h,'Visible','off')
107             axis off; axis image
108             view([90 90])
109             ifig = ifig + 1;
110         else
111             % Stall sections with alpha isolines
112             figure(ifig)
113             set(figure(ifig),'Color','w');
114             clf
115             polar(psi(ic),r_segn(ir),'k*');
116             hold on
117             fill(x_hub,y_hub,'k'); hold on; axis equal; axis off;
    grid off
118             str = {'\alpha_{e}(r,\psi)','* Stall',['\mu = ' num2str
    (mu)]};
119             text(r_segn(end-10),r_segn(end-2),str,'Color','k','
    FontSize',10);
120             contour(x,y,convang(alpha_eff','rad','deg'),'k','
    ShowText','on');
121             axis equal; grid on; axis off; axis image;
122             view([90 90])
123         end
124 end
125 end
```

4 Test Case

A test case was performed in order to validate the function. The helicopter analyzed is the *Sikorsky UH-60 "Black Hawk"*, and data were withdrawn on reports [2],[3] and [4]. *Ndim_ Coeff_ Articulated_ Rotor* of *https://github.com/TAARG-Education/Eli-TAARG* was adapted to the following case to solve articulated rotor. Beware that the helicopter data needs to be modified inside of the latter function.
The modified version of *Ndim_ Coeff_ Articulated_ Rotor* is shown in the following listing:

```
function [Tc,Hc,Yc,Qc,Pc,alfa,lambda,theta_0,beta_0,beta_1c,beta_1s
    ] = Ndim_Coeff_Articulated_Rotor(V_inf,h,Lock,f,X)

%% Input data for local test
 N        = 4;
 W        = 9979.0321;
 R        = 8.17;
 c        = 0.527;
 Cl_alpha = 5.73;
 Omega    = 27.0477;
 theta_tw = convang(-18,'deg','rad');
 Cd_mean  = 0.01;
```

Helicopter data are then defined and the inputs of *BladeSection_ AngleOfAttack* function are determined through the following procedure:

```
R = 8.17;              % Rotor radius [m/s]
c = 0.527;             % Chord [m]
A = pi*R^2;            % Disk area [m^2]
Cl_alpha = 5.73;       % Blade lift slope
Cd_mean = 0.01;        % Mean drag coefficiente
N = 4;                 % Number of blades
sigma = 0.082;         % Blade solidity
omega_tip = 220.98;    % Tip speed [m/s]
omega = omega_tip/R;   % Angular velocity [rad/s]
theta_tw = -18;        % Blade twist [deg]
W = 9979.0321;         % Weigth [kg]
W = 9.81*W;            % Weight [N]
h = 0;                 % Altitude [m]
Lock = 8.1936;         % Lock number
f = 3.376;             % Equivalent drag area [m^2]
X = 0;                 % Rate of climb [m/s]
V_inf = 55;            % Asymptotic velocity [m]

r_segn = linspace(0.1925,1,100);
psi = linspace(0,2*pi,100);
alpha_stall_up = 17;  % [deg]
alpha_stall_lo = -10; % [deg]

[Tc,Hc,Yc,Qc,Pc,alfa,lambda,theta_0,beta_0,beta_1c,beta_1s] = ...
    Ndim_Coeff_Articulated_Rotor(V_inf,h,Lock,f,X);

mu = V_inf*cos(convang(alfa,'deg','rad'))/(omega_tip);
beta = beta_0 + beta_1c*cos(psi) + beta_1s*sin(psi);
dbeta = -beta_1c*sin(psi) + beta_1s*cos(psi);
theta_tw = convang(-18,'deg','rad');
theta = theta_0 + theta_tw*r_segn;

```

```
[alpha_eff,Meff] = BladeSection_AngleOfAttack(r_segn,psi,mu,alfa,
    lambda, ...
     beta,dbeta,theta,h,omega_tip,alpha_stall_up,alpha_stall_lo,'off
    ');
```

The following figures shows the results. Figures 2 and 3 show angle-of-attack and Mach number distributions with black isolines, as the color flag option was set to be 'off'. Plotting options allows to visualize also maximum values of the angle of attack and Mach number. The results are in line with what is expected to happen from a theorical point of view. Mach number is maximum as the blade is advancing, and angles of attack are higher on the retreating side. Beware that UH-60 blade is provided with delta angle at the tip but this characteristic was ignored as articulated rotor theory is based on the rectangular plan form.
Another calculation at a much higher value of V_∞ was made in order to visualize how function represents the stalling region (Figure 3). Stall is expected to happen on the retreating side of the disk rotor.
Finally, it's possible to make a compare of the numerical results obtained, with the experimental results [5] shown in Figure 4. The above mentioned angle-of-attack distribution was computed for a different type of rotor and helicopter. Since it was not possible to withdraw correct data of the esperimental evaluation done in [5], this validation is merely qualitative .

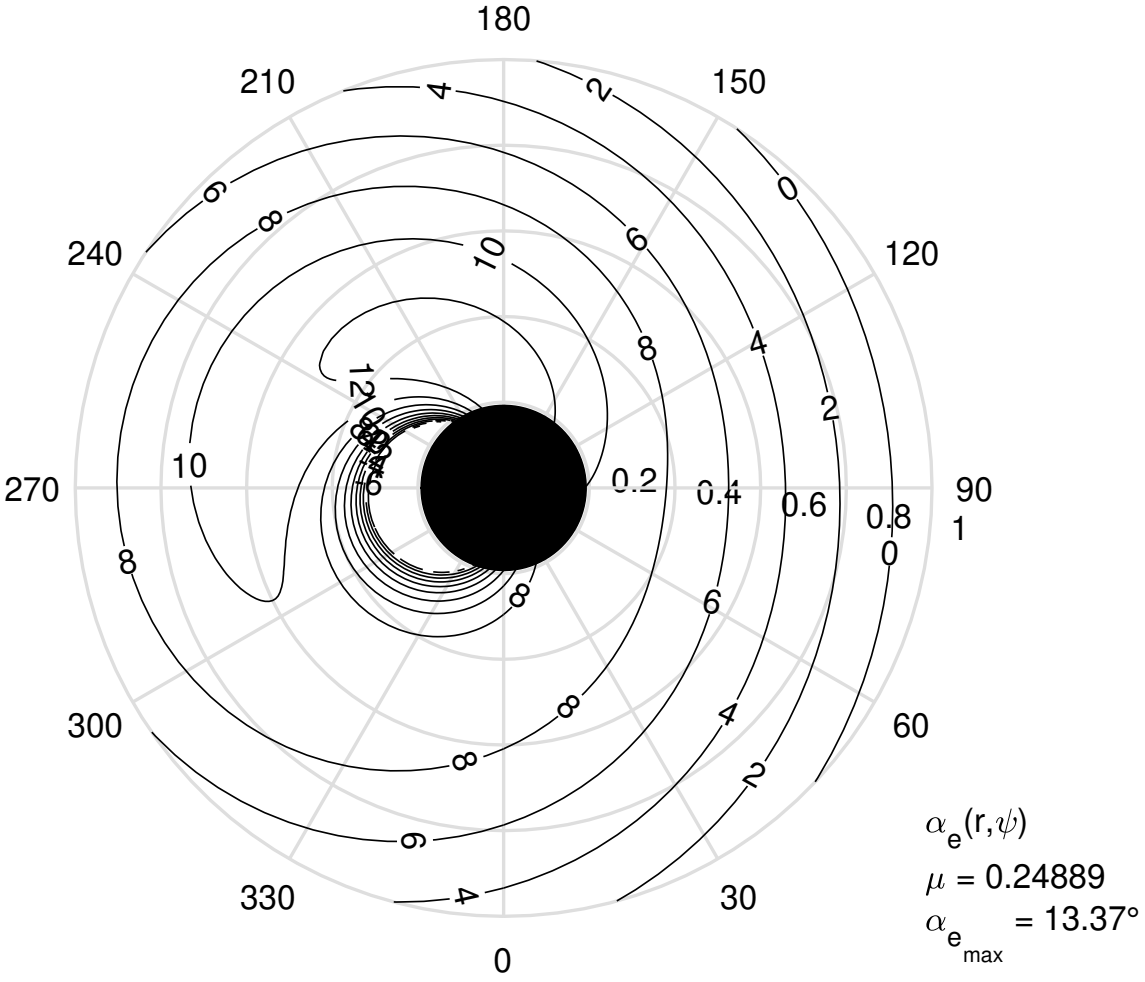

Figure 2: Effective angle-of-attack distribution on the rotor disk of the UH-60 *Black Hawk* in forward flight.

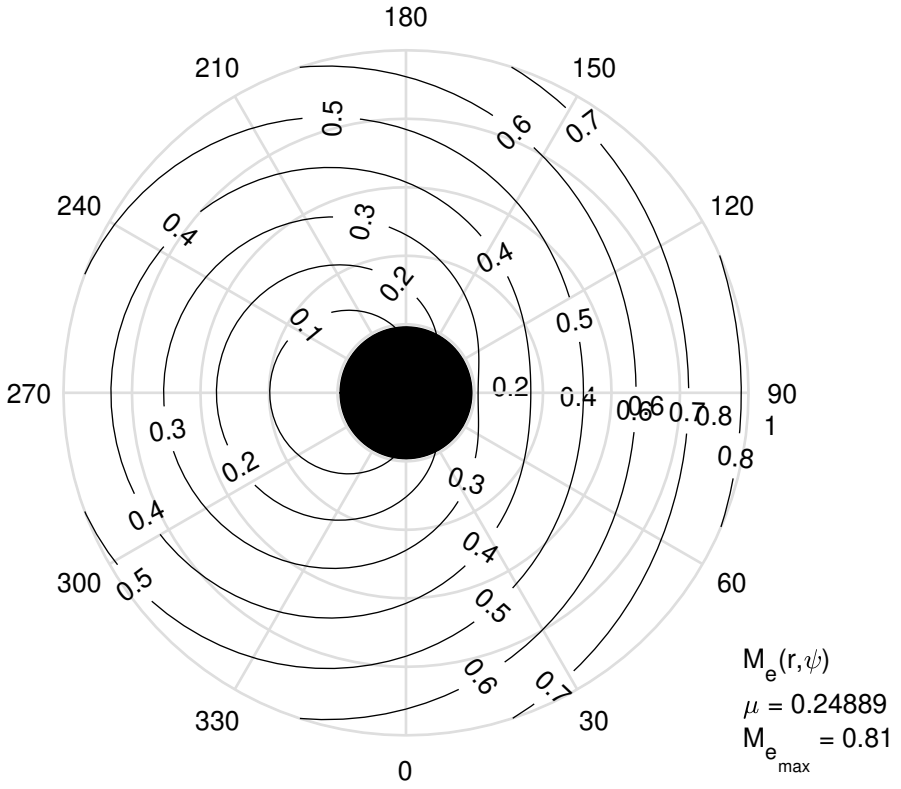

Figure 3: Effective Mach number distribution on the rotor disk of the UH-60 *Black Hawk* in forward flight.

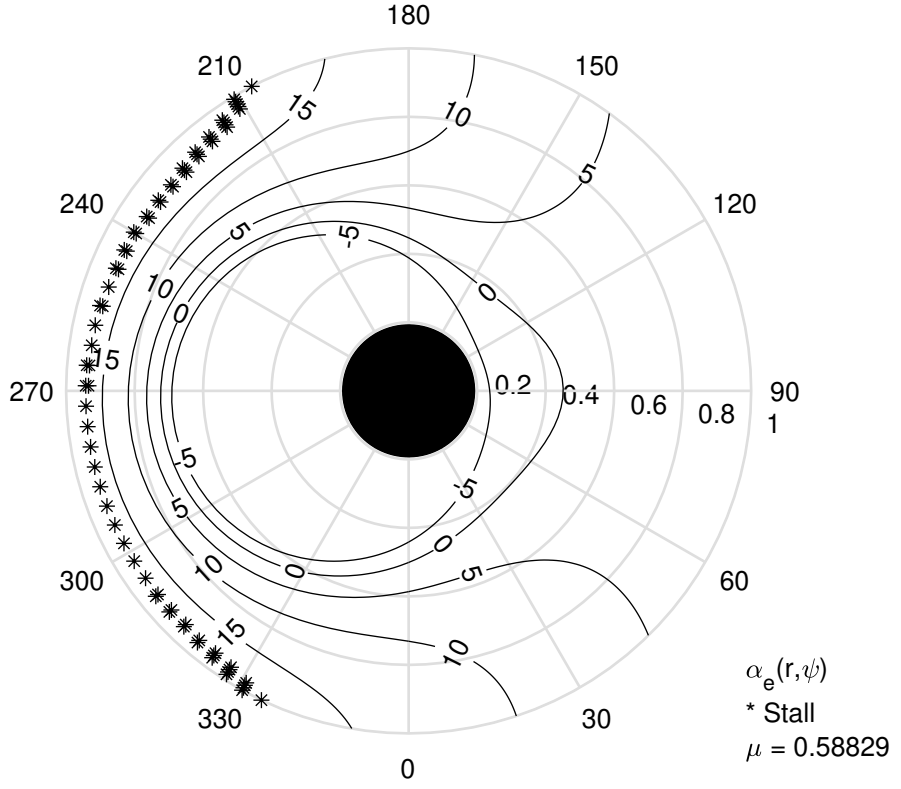

Figure 4: Effective angle-of-attack distribution on the rotor disk of the UH-60 *Black Hawk* in forward flight when rotor is stalling.

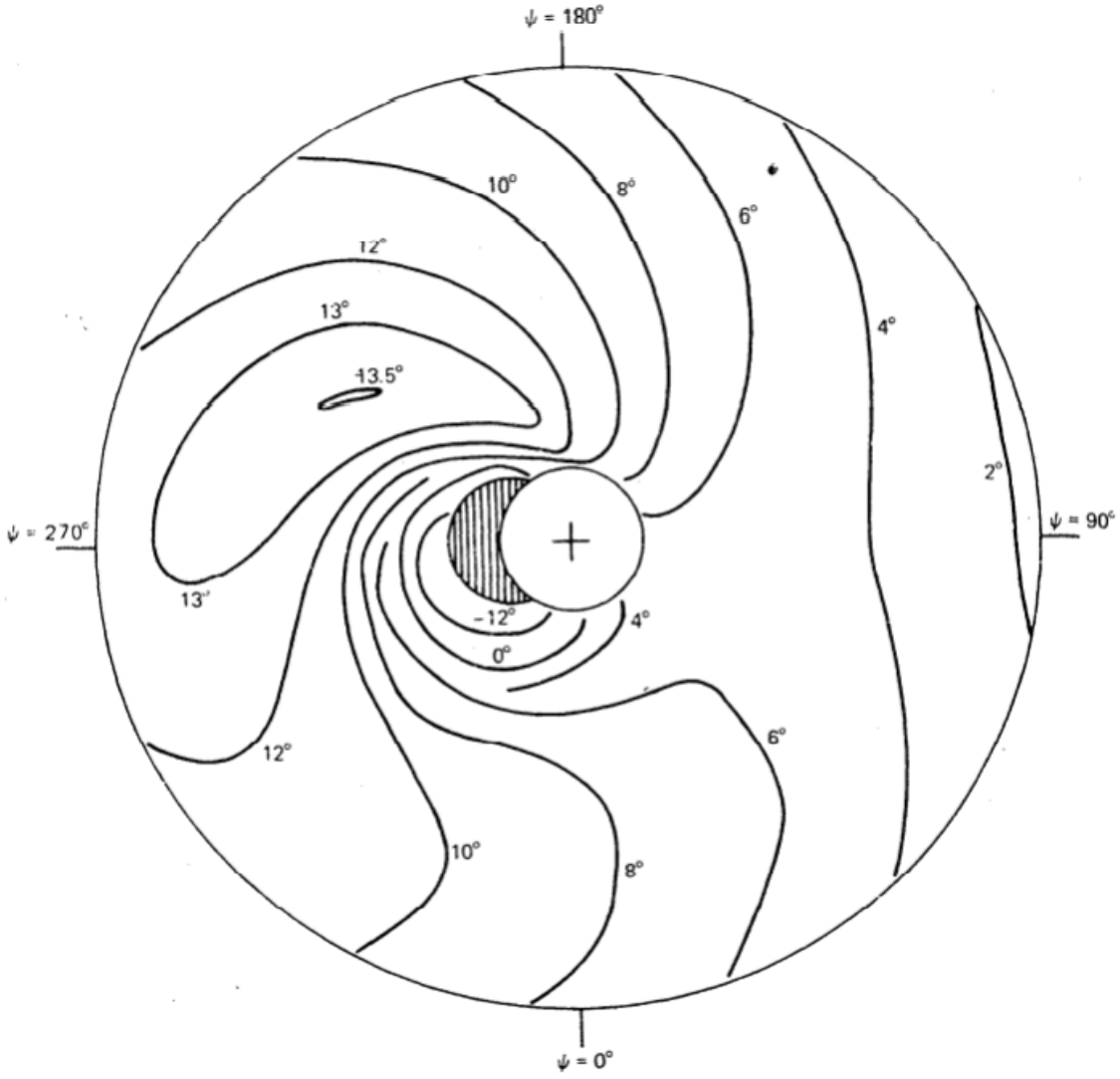

Figure 5: Experimental blade angle-of-attack distribution on rotor disk at $\mu = 0.25$, $\frac{f}{A} = 0.015$ and $\theta_{tw} = -8°$ [5].

References

[1] Tognaccini R., *Lezioni di Aerodinamica dell'Ala Rotante*. Università degli studi di Napoli Federico II, Napoli. A.A. 2022-2023, vsn 2.00.

[2] Kathryn B. Hilbert, *A Mathematical Model of the UH-60 Helicopter*. Nasa Ames Research Center, Moffett Field, California 94035.

[3] William G. Bousman, *Aerodynamic Characteristics of SC1095 and SC1094 R8 Airfoils*. Nasa Ames Research Center, Moffett Field, California 94035. December 2003.

[4] Hyeonsoo Yeo, William G. Bousman, Wayne Johnson, *Performance Analysis of a Utility Helicopter with Standard and Advanced Rotors*. Nasa Ames Research Center, Moffett Field, California 94035.

[5] Johnson Wayne, *Helicopter Theory*. Princepton University Press, Dover Publications, 1980.