

```

1 %% DBF Foamcutter for Wing
2 % This code is modified by Yuting Huang (ythuang96@gmail.com) based on
3 % Dr.Anderson's code written on Scilab.
4 % Please report all bug to the author's email address.
5 % Last updated: 8/23/2018
6
7 % This is written for DBF foamcutting, to generate G-code from wing
8 % parameters.
9 clear all; close all; clc;
10
11 %% Enter Parameters Below
12 % airfoil section file for root
13 root_filename = 'AH-79-100.dat';
14 % root chord length [in]
15 root_chord = 6;
16
17 % airfoil section file for tip
18 tip_filename = 'E216.dat';
19 % tip chord length [in]
20 tip_chord = 3;
21 % root chord has to be greater or equal to tip chord
22
23 % +1 for right wing, -1 for left wing
24 right_wing = 1;
25
26 % semi-span [in]
27 semi_span = 3;
28
29 % leading edge sweep [deg]
30 LE_sweep = 0;
31
32 % twist [deg]
33 twist = 0.0;
34
35 % g-code output file name
36 g_filename = 'HTail';
37
38 % width of CNC cutter [in]
39 cutter_width = 39;
40 % scale the cord length to accomodaate for broken trailing edge
41 % recommend using 1.2, then cut trailing edge with a blade to desired
42 % length.
43 scale_factor = 1.2;
44
45 %% -----
46 %% -----
47 %% -----
48 %% -----
49 %% Open File
50 fid = fopen(root_filename);
51 temp = textscan(fid, '%f %f', 'headerLines', 1);
52 fclose(fid);
53 root_pts = [temp{1}, temp{2}];
54

```

```

55 fid = fopen(tip_filename);
56 temp = textscan(fid, '%f %f', 'headerLines', 1);
57 fclose(fid);
58 tip_pts = [temp{1}, temp{2}];
59 clear temp fid;
60
61 %% Deconstruct Airfoil into Upper and Lower Parts
62 root_size = size(root_pts,1);
63 root_turn_point = find(root_pts(:,2)<0,1) - 1;
64 root_upper_x = root_pts(1:root_turn_point, 1);
65 root_upper_y = root_pts(1:root_turn_point, 2);
66 root_lower_x = root_pts(root_turn_point:root_size, 1);
67 root_lower_y = root_pts(root_turn_point:root_size, 2);
68
69 tip_size = size(tip_pts,1);
70 tip_turn_point = find(tip_pts(:,2)<0,1) - 1;
71 tip_upper_x = tip_pts(1:tip_turn_point, 1);
72 tip_upper_y = tip_pts(1:tip_turn_point, 2);
73 tip_lower_x = tip_pts(tip_turn_point:tip_size, 1);
74 tip_lower_y = tip_pts(tip_turn_point:tip_size, 2);
75
76 %% Interpolate
77 n = 301;
78 root_upper_yp = interp1(root_upper_x, root_upper_y, linspace(1,0,n)');
79 root_upper_xp = linspace(1,0,n)';
80 root_lower_yp = interp1(root_lower_x, root_lower_y, linspace(0,1,n)');
81 root_lower_xp = linspace(0,1,n)';
82
83 tip_upper_yp = interp1(tip_upper_x, tip_upper_y, linspace(1,0,n)');
84 tip_upper_xp = linspace(1,0,n)';
85 tip_lower_yp = interp1(tip_lower_x, tip_lower_y, linspace(0,1,n)');
86 tip_lower_xp = linspace(0,1,n)';
87
88 %% Scale to Chord Length
89 % chord length on machine
90 root_chord_ext = root_chord + ...
91     0.5*(root_chord - tip_chord)*(cutter_width - semi_span)/semi_span;
92 tip_chord_ext = root_chord - ...
93     0.5*(root_chord - tip_chord)*(cutter_width + semi_span)/semi_span;
94
95 root_upper_xp = root_chord_ext*root_upper_xp;
96 root_upper_yp = root_chord_ext*root_upper_yp;
97
98 root_lower_xp = root_chord_ext*root_lower_xp;
99 root_lower_yp = root_chord_ext*root_lower_yp;
100
101 tip_upper_xp = tip_chord_ext*tip_upper_xp;
102 tip_upper_yp = tip_chord_ext*tip_upper_yp;
103
104 tip_lower_xp = tip_chord_ext*tip_lower_xp;
105 tip_lower_yp = tip_chord_ext*tip_lower_yp;
106
107 %% Rotate Tip by Twist Angle
108 c4 = tip_chord/4;

```

```

109 twist_rad = (pi/180)*twist;
110
111 tip_upper_xpr = tip_upper_xp*cos(twist_rad) ...
112     + tip_upper_yp*sin(twist_rad) + c4*(1.0-cos(twist_rad));
113 tip_upper_ypr = -tip_upper_xp*sin(twist_rad) ...
114     + tip_upper_yp*cos(twist_rad) + c4*sin(twist_rad);
115
116 tip_lower_xpr = tip_lower_xp*cos(twist_rad) ...
117     + tip_lower_yp*sin(twist_rad) + c4*(1.0-cos(twist_rad));
118 tip_lower_ypr = -tip_lower_xp*sin(twist_rad) ...
119     + tip_lower_yp*cos(twist_rad) + c4*sin(twist_rad);
120
121 %% Use Sweep Angle to Shift Tip Relative to Root
122 sweep_shift = cutter_width*tan(LE_sweep*pi/180);
123
124 tip_upper_xpr = tip_upper_xpr + sweep_shift;
125 tip_lower_xpr = tip_lower_xpr + sweep_shift;
126
127 %% Swap x-axis to Start Cut on Trailing Edge
128 root_upper_xp = root_chord_ext - root_upper_xp;
129 root_lower_xp = root_chord_ext - root_lower_xp;
130 tip_upper_xpr = root_chord_ext - tip_upper_xpr;
131 tip_lower_xpr = root_chord_ext - tip_lower_xpr;
132
133 %% Combine Upper and Lower Surfaces
134 root_x = [root_upper_xp; root_lower_xp];
135 root_y = [root_upper_yp; root_lower_yp];
136
137 tip_x = [tip_upper_xpr; tip_lower_xpr];
138 tip_y = [tip_upper_ypr; tip_lower_ypr];
139
140 %% Make a Plot of Root and Tip
141 set(0, 'defaultlinelinenewidth', 2)
142 set(0, 'defaultaxeslinelinenewidth', 1)
143 set(0, 'defaultaxesfontsize', 20)
144
145 figure(1); set(1, 'position', [0 0 1920 1080]); hold on;
146 plot(root_x, root_y);
147 plot(tip_x, tip_y);
148 legend1 = legend('Root', 'Tip');
149 set(legend1, 'interpreter', 'latex'); set(legend1, 'fontsize', 18);
150 title('Wing on FoamCutter', 'interpreter', 'latex', 'fontsize', 25);
151 xlabel('X [in]', 'interpreter', 'latex', 'fontsize', 25);
152 ylabel('Y [in]', 'interpreter', 'latex', 'fontsize', 25);
153 axis equal; grid on;
154
155 %% Write G-code File
156 root_x = scale_factor*root_x*25.4;
157 root_y = scale_factor*root_y*25.4;
158 tip_x = scale_factor*tip_x*25.4;
159 tip_y = scale_factor*tip_y*25.4;
160
161 fidw = fopen([g_filename '.txt'], 'wt');
162

```

```

163 fprintf(fidw, 'G21\n'); fprintf(fidw, 'M48\n');
164 fprintf(fidw, 'F80\n'); fprintf(fidw, 'S80\n');
165
166 fprintf(fidw, 'G1 X % 8.3f Y % 8.3f Z % 8.3f A % 8.3f\n', 0,0,0,0);
167
168 if right_wing > 0
169     fprintf(fidw, 'G1 X % 8.3f Y % 8.3f Z % 8.3f A % 8.3f\n', ...
170         root_x(1),root_y(1),tip_x(1),tip_y(1));
171     fprintf(fidw, 'G4 P5\n');
172     for i=length(root_x):-1:1
173         fprintf(fidw, 'G1 X % 8.3f Y % 8.3f Z % 8.3f A % 8.3f\n', ...
174             root_x(i),root_y(i),tip_x(i),tip_y(i));
175     end
176 else
177     fprintf(fidw, 'G1 X % 8.3f Y % 8.3f Z % 8.3f A % 8.3f\n', ...
178         tip_x(1),tip_y(1),root_x(1),root_y(1));
179     fprintf(fidw, 'G4 P5\n');
180     for i=length(root_x):-1:1
181         fprintf(fidw, 'G1 X % 8.3f Y % 8.3f Z % 8.3f A % 8.3f\n', ...
182             tip_x(i),tip_y(i),root_x(i),root_y(i));
183     end
184 end
185
186 fprintf(fidw, 'G1 X % 8.3f Y % 8.3f Z % 8.3f A % 8.3f\n', 0,0,0,0);
187 fprintf(fidw, 'M2');
188
189 fclose(fidw);
190
191

```