
Table of Contents

Statique	1
Représentation graphique (plan 2D) de F en statique	2
Représentation vraie portance (axe z) et vraie trainée (axe x)	3
Polaire	4
Moyenne des signaux dynamiques	5
FFT (Fréquence d'excitation)	7
Gain apporté par le pompage	9

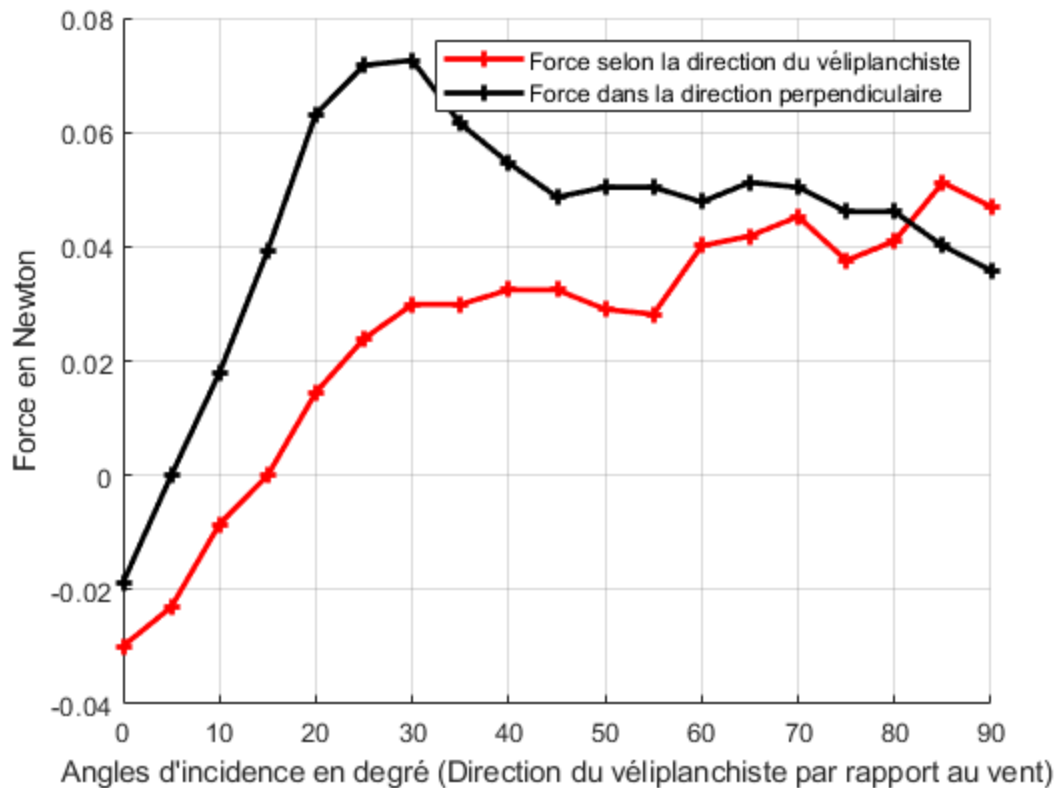
Statique

```
x_p = load("statique_portance.txt");
x_t = load("statique_trainee.txt");

m = 8.54e-4 ; %conversion en Newton
a = x_p(:,1); % angles
F_p = m*(x_p(:,3)-x_p(:,2)) ;
F_t = m*(x_t(:,3)-x_t(:,2)) ;
F = sqrt(F_p.^2+F_t.^2); % force globale, norme totale

figure(1), clf
hold on
plot(a,F_p,'+-red','LineWidth',2)
plot(a,F_t,'+-black','LineWidth',2)

legend('Force selon la direction du véliplanchiste','Force dans la
direction perpendiculaire')
xlabel("Angles d'incidence en degré (Direction du véliplanchiste par
rapport au vent)")
ylabel('Force en Newton')
grid on
F_2d = [ F_p F_t ] ;
```



Représentation graphique (plan 2D) de F en statique

```

F_cart = zeros(19,2) ; %rep des forces dans le plan cartésien
a_r = a*pi/180 ;
for i=1:19
    F_cart(i,1) = -F_2d(i,1)*cos(a_r(i)) + F_2d(i,2)*sin(a_r(i));
    F_cart(i,2) = F_2d(i,2)*cos(a_r(i)) + F_2d(i,1)*sin(a_r(i));
end

O = zeros(19,2) ;
figure(2), clf
hold on
leg = {};

for i=1:2:19

    quiver(O(i,1),O(i,2),F_cart(i,1),F_cart(i,2),1, 'LineWidth',2,'Color',
[0,1-i*0.03,i*0.03])
    leg = cat(2,leg , sprintf(num2str(a(i))));
end

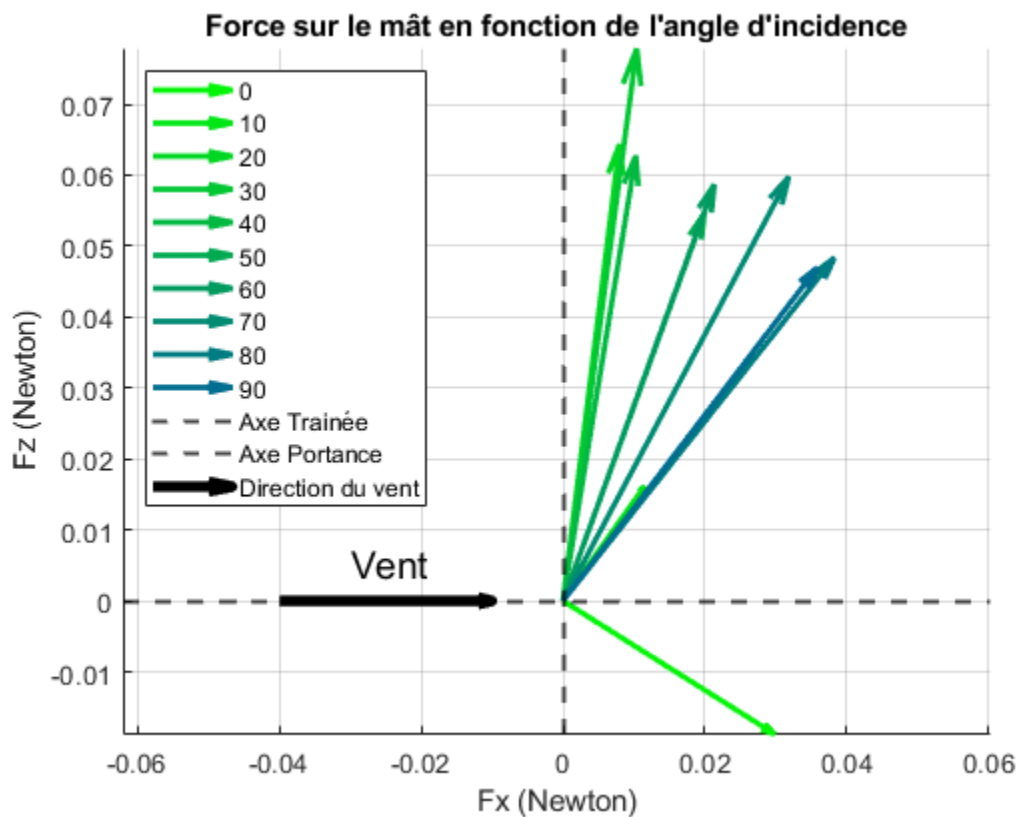
```

```

axis equal
xlabel('Fx (Newton)')
ylabel('Fz (Newton)')
xline(0, '--black', 'Linewidth',1.5)
yline(0, '--black', 'Linewidth',1.5)
quiver(-0.04,0,0.03,0,1, '-black', 'LineWidth',4)
text(-0.03,0.005, 'Vent', 'Color',[0 0 0], 'FontSize',14)

leg = cat(2,leg , {'Axe Trainée', 'Axe Portance', 'Direction du vent'});
legend(leg, 'Location', 'NorthWest')
grid on
title("Force sur le mât en fonction de l'angle d'incidence")

```



Représentation vraie portance (axe z) et vraie trainée (axe x)

```

U = 1.8 ; %vitesse de vent
rho = 1.3 ; % masse volumique de l'air
S = 9/15^2 ; %surface maquette 1/15
Cx = F_cart(:,1)/(1/2*rho*U^2*S) ; % Coeff trainée
Cp = F_cart(:,2)/(1/2*rho*U^2*S) ; % Coeff portance

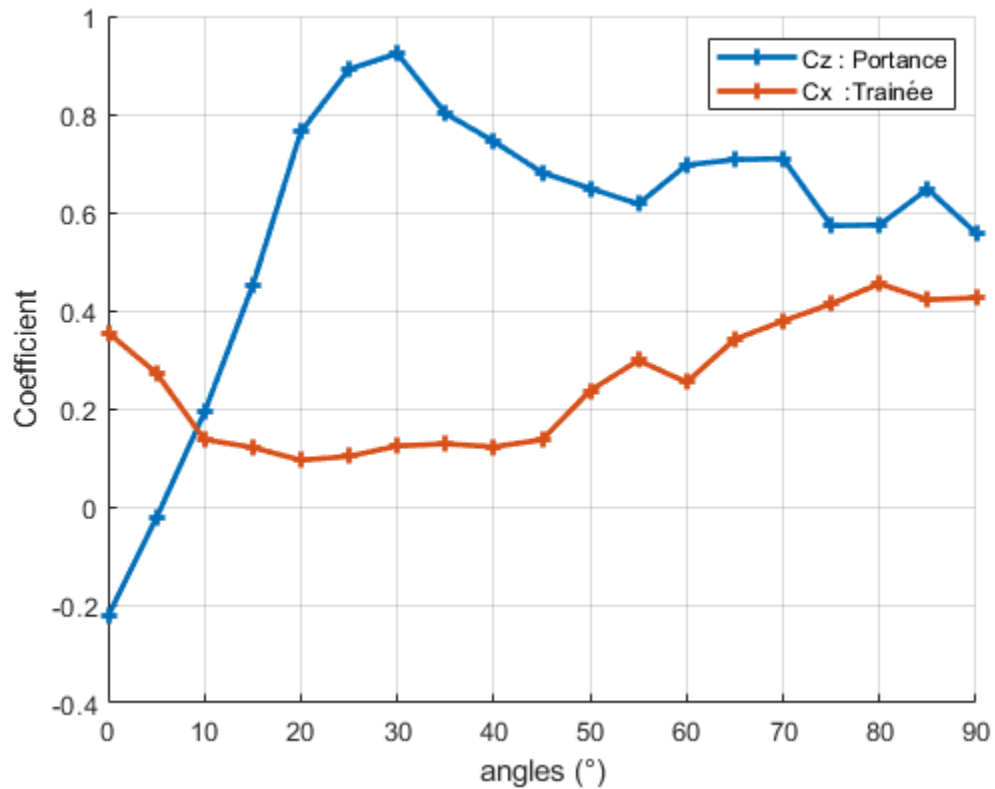
```

```
figure(3),clf
```

```

hold on
plot(a,Cp,'+-','LineWidth',2)
plot(a,Cx,'+-','LineWidth',2)
xlabel('angles (°)')
ylabel('Coefficient')
legend('Cz : Portance','Cx :Trainée')
grid on

```



Polaire

```

figure(4),clf

subplot(2,2,1)
plot(a,Cp,'+-blue','LineWidth',2)
xlabel('angles (°)')
ylabel('Cz : Portance')
grid on
title('Portance')

subplot(2,2,2)
plot(a,Cx,'+-black','LineWidth',2)
xlabel('angles (°)')
ylabel('Cx : Trainée')
grid on
title('Trainée')

```

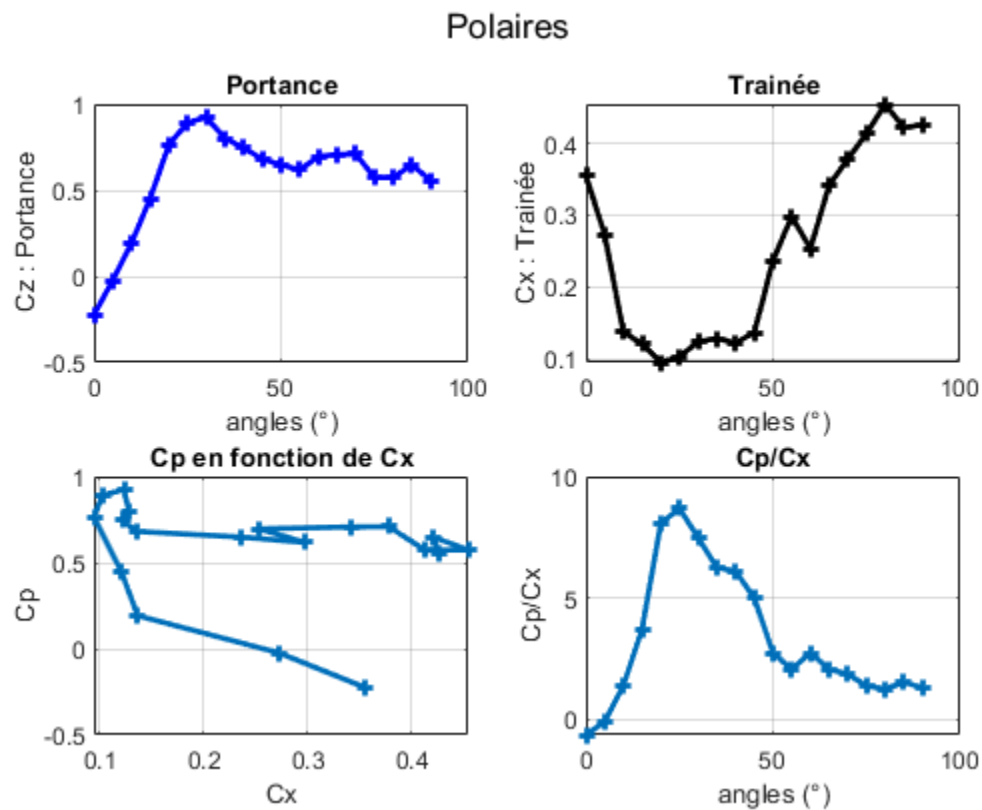
```

subplot(2,2,3)
plot(Cx,Cp, '+-', 'LineWidth', 2)
xlabel('Cx')
ylabel('Cp')
grid on
title('Cp en fonction de Cx')

subplot(2,2,4)
plot(a,Cp./Cx, '+-', 'LineWidth', 2)
xlabel('angles (°)')
ylabel('Cp/Cx')
grid on
title('Cp/Cx')

sgtitle('Polaires')

```



Moyenne des signaux dynamiques

Exemple pour le signal de fréquence 1Hz, amplitude 1 (15°) et angle 10°

```

freq = 1 ;
ampli = 1 ;
ang = 1;
f = mesure(:,freq,ampli,ang); % on charge les mesures dynamiques
f_s = mesure_stat(:,1,ampli,ang) ; % on charge les mesures statiques

```

```

N_f = length(f) ;
N_f_s = length(f_s) ;
i = 1 ;
while i<N_f %on supprime les 0 des listes
    if f(i)==0
        N_f = i-2 ;
    end
    i= i+ 1;
end
i=1 ;
while i<N_f_s %on supprime les 0 des listes
    if f_s(i)==0
        N_f_s = i-2 ;
    end
    i=i+1 ;
end

f_d = f(2:N_f) ;
f_s = f_s(2:N_f_s) ;

% A METTRE EN COMMENTAIRE, si pb de zéro au minimum du signal
dynamique
% f_d = mesure(2:1500,2,2,5);
% N_f = length(f_d)+1;
% fin de la section

m_s = mean(f_s) ;
m_d = mean(f_d);
N_f = N_f -1 ;
N_f_s = N_f_s - 1 ;

m_s_t = ones(N_f_s,1)*m_s;
m_d_t = ones(N_f,1)*m_d ;

delay = 1e-3 ; % voir delay sur arduino
dt = 3/480 ;
fe = 1/dt ; %fréquence d'échantillonnage
N = length(f);
time = 0:dt:(N-1)*dt ;
Pt = 0:1:N-1 ;

figure(5), clf
fig = plot(time(1:N_f),f_d,'-red',time(1:N_f_s),f_s,'-
c',time(1:N_f),m_d_t,'red',time(1:N_f_s),m_s_t,'b');
legend('Mesure dynamique, md','Mesure statique', 'Moyenne
dynamique', 'Moyenne statique')
xlabel('temp(s)')
ylabel('Tension(Volt)')
leg = sprintf("Dynamique = %g \n Statique = %g \n Gain = %g",m_d,m_s,
m_d-m_s);
text(2,m_d+20,leg)

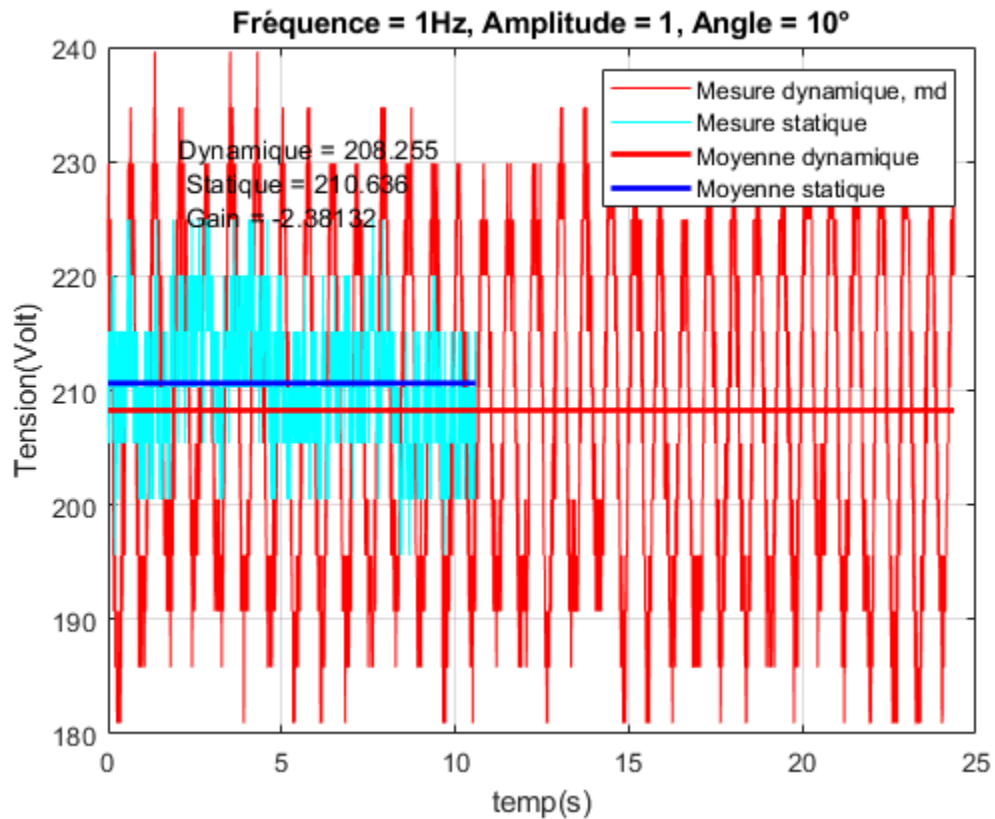
```

```

titre = sprintf("Fréquence = %gHz, Amplitude = %g, Angle = %g°",
    freq,ampli,ang*10) ;
title(titre)

fig(1).LineWidth = 0.1 ;
fig(2).LineWidth = 0.1 ;
fig(3).LineWidth = 2 ;
fig(4).LineWidth = 2 ;
grid on

```



FFT (Fréquence d'excitation)

Exemple pour le signal de fréquence 1Hz, amplitude 1 (15°) et angle 10°

```

figure (6), clf
fr = 1 ;
ampli = 1 ;
ang = 1;
N_f = 1000;
delay = 1e-3 ; % voir delay sur arduino
dt = 3/480 ;
fe = 1/dt ; %fréquence d'échantillonnage
N = N_f ;
time = 0:dt:(N-1)*dt ;
Pt = 0:1:N-1 ;

```

```

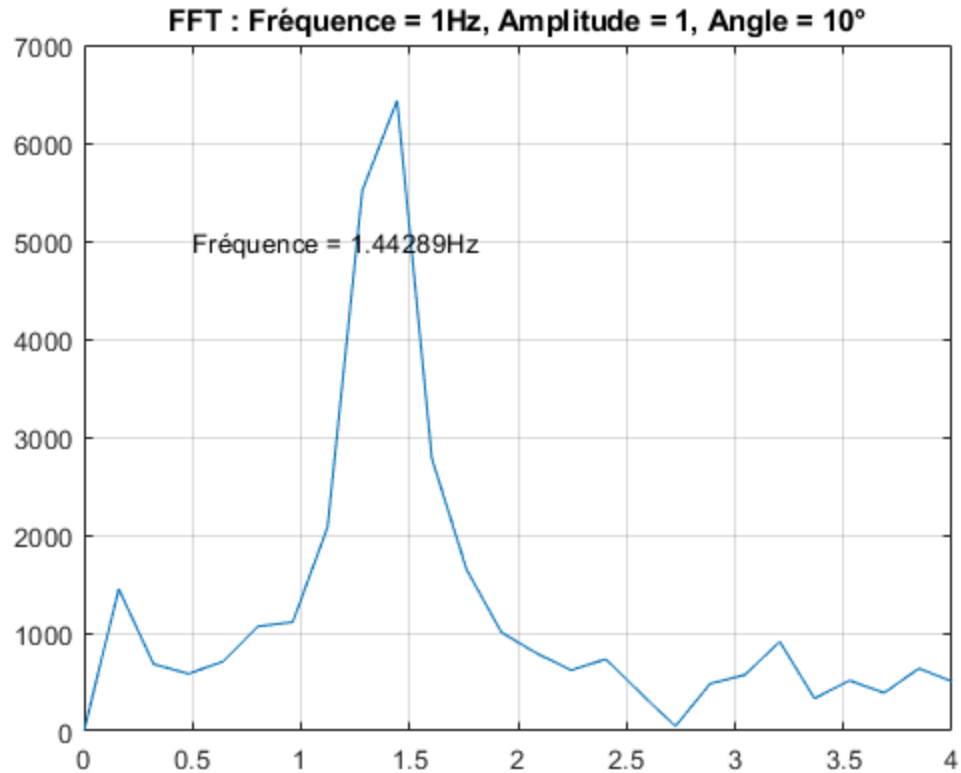
f = mesure(:,fr,ampli,ang);
f_dyn = f(2:N_f); % signal sans 0
m_d = mean(f_dyn);
N_f = N_f -1;
tps = time(1:N_f);
f_red = f_dyn-m_d; % signal réduit, remis à 0, sans offset
TF = fft(f_red);
T = tps(end);
df = 1/T ;
n_2 = floor(N_f/2);
TF = abs(TF(1:n_2));
freq = 0:df:df*(N_f-1);

plot(freq(1:n_2),TF)
xlim([0 4])
grid on
titre = sprintf("FFT : Fréquence = %gHz, Amplitude = %g, Angle = %g°",
    fr,ampli,ang*10) ;
[M, I] = max(TF) ;
f0 = freq(I) ;
leg = sprintf("Fréquence = %gHz",f0)
text(0.5,5000,leg)
title(titre)

leg =

    "Fréquence = 1.44289Hz"

```

Gain apporté par le pompage

Les valeurs entre crochet sont Dynamique(arduino) - statique (arduino)

```

m = -2.4e3 ; %coefficient étalonnage arduino, NEGATIF
% La conversion à appliquer est Réel = Arduino /m
g1_1hz = 1/m*[-2.38132 0.743072 7.8822 -23.038 -7.08417 3.2931
-7.28307 -1.6752 2.4362] ; %amplitude 1, fréquence d'excitation 1 Hz
g1_2hz = 1/m*[-4.54597 -0.664012 12.0506 -19.1624 -10.909 -8.79921
-19.6265 -6.6798 -7.823] ; % fréquence de 2 Hz
g1_3hz = 1/m*[-3.73929 2.62482 21.7503 -14.7291 -21.4342 -17.5297
-31.9205 -9.36503 -8.843] ; % 3 Hz

g2_1hz = 1/m*[3.75576 21.5967 23.8391 -11.6516 -16.2554 -15.0828
-15.279 11.9047 -4.87875] ; %amplitude 2
g2_2hz = 1/m*[7.86013 28.5536 42.3294 1.32931 -36.3513 -34.0982
-38.833 -18.5077 -19.6027] ;
g2_3hz = 1/m*[10.8972 33.0501 22.1504 2.65902 NaN NaN NaN NaN NaN] ; %
Nan car mesures non réalisables

angles = [10 20 30 40 50 60 70 80 90] ;

figure(7),clf
hold on
yline(0,'black')
plot(angles,g1_1hz./abs(Stat)*100,'+-blue')

```

```

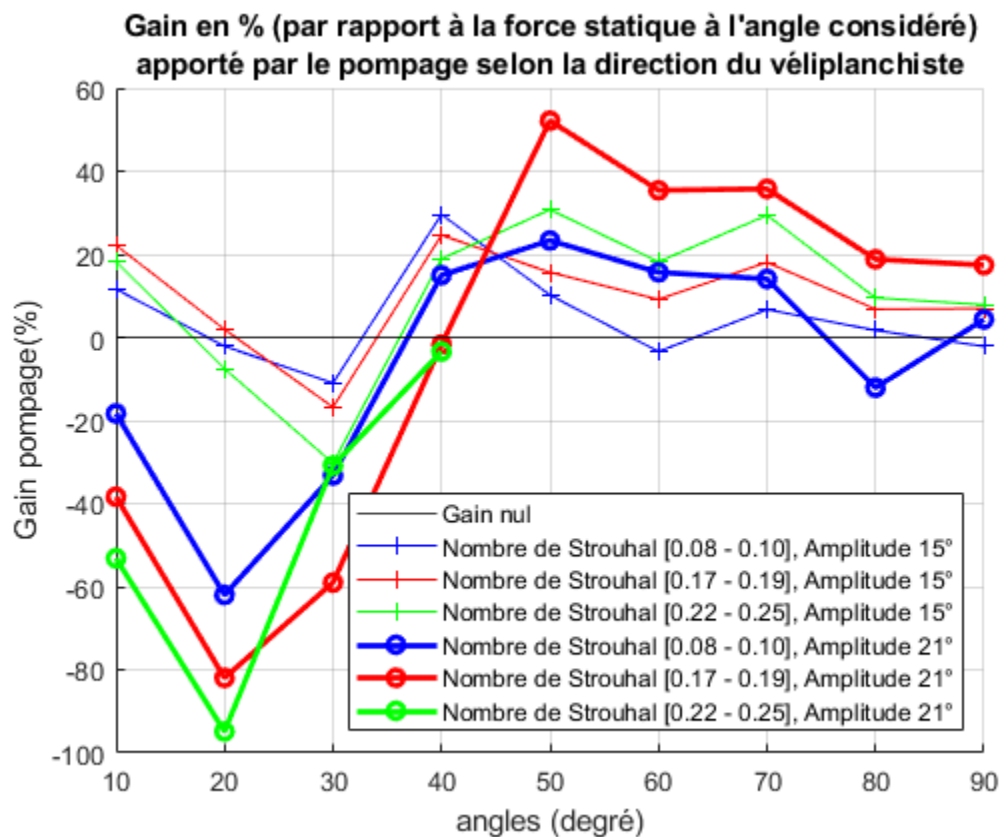
xlabel('angles (degré)')
ylabel("Gain pompage(%)")
grid on
plot(angles,g1_2hz./abs(Stat)*100,'+-r')
plot(angles,g1_3hz./abs(Stat)*100,'+-g')

plot(angles,g2_1hz./abs(Stat)*100,'o-blue','LineWidth',2)
plot(angles,g2_2hz./abs(Stat)*100,'o-r','LineWidth',2)
plot(angles,g2_3hz./abs(Stat)*100,'o-g','LineWidth',2)

legend('Gain nul','Nombre de Strouhal [0.08 - 0.10], Amplitude
15°',...
'Nombre de Strouhal [0.17 - 0.19], Amplitude 15°',...
'Nombre de Strouhal [0.22 - 0.25], Amplitude 15°',...
'Nombre de Strouhal [0.08 - 0.10], Amplitude 21°',...
'Nombre de Strouhal [0.17 - 0.19], Amplitude 21°',...
'Nombre de Strouhal [0.22 - 0.25], Amplitude 21°',...
'Location','SouthEast')

title({"Gain en % (par rapport à la force statique à l'angle
considéré)" "apporté par le pompage selon la direction du
véliplanchiste"})

```



Published with MATLAB® R2020a