Analisi cuscinetti:

```
%% Analysis of the Localized fault
%
% G. D'Elia

clear
clc

%% Load signal
load localizzato_bk1
load localizzato_tacho

fs = str2num(fs);

x = bk1(1:2^16);
tacho = tacho(1:2^16);

L = length(x);
t = [0:L-1]/fs;
f = [0:L-1]*fs/L;

clear bk1


% Signal plot
figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 16 6])
plot(t,x,'k')
xlim([0 0.2])
xlabel('Time [s]')
ylabel('Amplitude')

% tacho plot
figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 16 6])
plot(t,tacho,'k')
xlim([0 0.2])
xlabel('Time [s]')
ylabel('Amplitude')


%% Velocity extraction
triglevel = 6;            % livello del triggher
test = tacho>triglevel;                  % restituisce 1 o 0 in caso sia vera o falsa
test = diff(test);                % restituisce 1 per il superamento in salita e -1 per il
superamento in discesa
test=(test > 0.5);                % tiene solo il superamento in salita
ind = find(test);                 % trova gli indici del superamento in salita della soglia
indup = ind+1;          %indice subito dopo il superamento della soglia
```

```
fr = fs./diff(indup);

%% Signal spectrum
X = 1/L .* fft(x);

figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 16 6])
plot(f,abs(X),'k')
xlim([0 2.5e4])
xlabel('Frequency [Hz]')
ylabel('Amplitude')

%% Signal filtering
[b,a]= butter(8,[20000 25000]/fs*2);
xf = filter(b,a,x);

%% Envelope spectrum
% analitic signal
x_env = abs(hilbert(xf));

% Spectrum of the envelope
X_env = length(x_env).*fft(x_env);

% Plot env
figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 16 6])
plot(f,abs(X_env),'k');
xlim([0 2000])
xlabel('frequency [Hz]');
ylabel('amplitude');


%% Evaluation of the residual signal
% Synchronous average
% Calcolo della media sincrona con un delta O di 1
M = fix(fs/mean(fr));

deltaangolo = 360/M;
angoli = [0:M-1]*deltaangolo;

endk = length(indup) - 1;
cicli_x = zeros(endk,length(angoli));

for k = 1:endk,
    ciclo = x(1,indup(k):indup(k+1)-1);
    tciclo = [0:length(ciclo)-1]*(360/length(ciclo));
    cicli_x(k,:) = interp1(tciclo,ciclo,angoli,'spline');
end

% vettore che contiene la media sincrona fatta su un giro
```

```
media_sinc_x = mean(cicli_x);

% Estrazione segnale residuo
xa = cicli_x';
xa = xa(:);
xa = xa';

xd = media_sinc_x';
xd = xd(:,ones(1,endk));
xd = xd(:);
xd = xd';

xr = xa - xd;

%% Ricampionamento fisso
length_xr_fix = indup(endk+1) - indup(1);
index_ciclo = 1:M;
xr_fix = zeros(1,length_xr_fix);
for indk = 1:endk,
        ciclo_a = xr(index_ciclo);
        t_index = indup(indk):indup(indk+1)-1;
        tciclo = (0:length(t_index)-1)*(360/length(t_index));
        ciclo_fix = interp1(angoli,ciclo_a,tciclo,'spline');
        xr_fix(indup(indk):indup(indk+1)-1) = ciclo_fix;
        index_ciclo = index_ciclo + M;
end


%% Cyclic modulation spectrum
fracovlp = .75;     % fraction of overlap (should be greater than or equal to .75)
Nw = 2^5;
Nfft = Nw;

Noverlap = fix(fracovlp*Nw);
R = Nw - Noverlap;
Win = hanning(Nw);
[Stf,f,t] = specgram(xr_fix,Nfft,fs,Win,Noverlap);
Lt = length(t);

Stf = abs(Stf).^2;
Saf = (fft(Stf',Lt))';
Saf = Saf(1:Nfft/2,1:fix(Lt/2));

% Plot cms
f = f(1:Nfft/2);
a = (0:fix(Lt/2)-1)*fs/(Lt*Nw*(1-fracovlp));
If = find(f>1.5e4 & f<2.5e4);

figure, mesh(a(2:end),f(If),abs(Saf(If,2:end))), colormap(gray)
xlabel(['\alpha [Hz] ; \Delta \alpha = ',num2str(diff(a(1:2))),' Hz'])
ylabel(['f [Hz] ; \Delta f = ',num2str(f(2)),' Hz'])
title('Cyclic Modulation Spectrum'),
```

Analisi ruote dentate:

```
clear
clc

load('pam3_bear_6007_runup_5Hz_sx.mat');


fs = 51200;

x = n_b_k.y_values.values;
tacho = n_tacho.y_values.values;

clear n_b_k n_tacho

L = 2^19;

x = x(2.4e6:2.4e6+L-1);
tacho = tacho(2.4e6:2.4e6+L-1);
t = (0:L-1)/fs;

figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 16 6])
plot(t,x,'k')
xlabel('Time [s]')
ylabel('Amplitude')

%% Spectrum
X = 1/L * fft(x);
f = (0:L-1)*fs/L;

figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 16 6])
plot(f,abs(X),'k')
xlim([0 2.5e4])
xlabel('Frequency [Hz]')
ylabel('Amplitude')


%% Velocity extraction
triglevel = 3;            % livello del triggher
test = tacho>triglevel;                  % restituisce 1 o 0 in caso sia vera o falsa
test = diff(test);                % restituisce 1 per il superamento in salita e -1 per il
superamento in discesa
test=(test > 0.5);                % tiene solo il superamento in salita
ind = find(test);                 % trova gli indici del superamento in salita della soglia
indup = ind+1;          %indice subito dopo il superamento della soglia

fr = fs./diff(indup);
```

```
%% Evaluation of the residual signal
% Synchronous average
% Calcolo della media sincrona con un delta O di 1
M = fix(fs/mean(fr));

deltaangolo = 360/M;
angoli = [0:M-1]*deltaangolo;

endk = length(indup) - 1;
cicli_x = zeros(endk,length(angoli));

for k = 1:endk,
    ciclo = x(1,indup(k):indup(k+1)-1);
    tciclo = [0:length(ciclo)-1]*(360/length(ciclo));
    cicli_x(k,:) = interp1(tciclo,ciclo,angoli,'spline');
end

% vettore che contiene la media sincrona fatta su un giro
media_sinc_x = mean(cicli_x);

figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 16 6])
plot(angoli,media_sinc_x,'k')
xlabel('angle [deg]')
ylabel('Amplitude')


%% FFT media sinc
fft_media_sinc = 1/M*fft(media_sinc_x);

figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 16 6])
stem((0:M-1),abs(fft_media_sinc),'k')
xlabel('Order')
ylabel('Amplitude')
xlim([0 60])


%% modulazione di ampiezza
%% Signal filtering
fft_media_sinc_fill = zeros(1,M);
fft_media_sinc_fill(1,14:22) = fft_media_sinc(1,14:22);
fft_media_sinc_fill(1,M+2-22:M+2-14) = fft_media_sinc(1,M+2-22:M+2-14);

media_sinc_fill = M*ifft(fft_media_sinc_fill);

am = abs(hilbert(media_sinc_fill));

figure,
set(gcf,'Units','centimeters')
```

```
set(gcf,'Position',[2 10 16 6])
plot(angoli,am,'k')
xlabel('angle [deg]')
ylabel('Amplitude')


%% Segnale residuo
% Estrazione segnale residuo
xa = cicli_x';
xa = xa(:);
xa = xa';

xd = media_sinc_x';
xd = xd(:,ones(1,endk));
xd = xd(:);
xd = xd';

xr = xa - xd;

%% Ricampionamento fisso
length_xr_fix = indup(endk+1) - indup(1);
index_ciclo = 1:M;
xr_fix = zeros(1,length_xr_fix);
for indk = 1:endk,
        ciclo_a = xr(index_ciclo);
        t_index = indup(indk):indup(indk+1)-1;
        tciclo = (0:length(t_index)-1)*(360/length(t_index));
        ciclo_fix = interp1(angoli,ciclo_a,tciclo,'spline');
        xr_fix(indup(indk):indup(indk+1)-1) = ciclo_fix;
        index_ciclo = index_ciclo + M;
end



%% CMS
fracovlp = .75;     % fraction of overlap (should be greater than or equal to .75)
Nw = 2^7;
Nfft = Nw;

Noverlap = fix(fracovlp*Nw);
R = Nw - Noverlap;
Win = hanning(Nw);
[Stf,f,t] = specgram(x,Nfft,fs,Win,Noverlap);
Lt = length(t);

Stf = abs(Stf).^2;
Saf = 1/Lt*(fft(Stf',Lt))';
Saf = Saf(1:Nfft/2,1:fix(Lt/2));

% Plot cms
f = f(1:Nfft/2);
a = (0:fix(Lt/2)-1)*fs/(Lt*Nw*(1-fracovlp));
```

```
lfm = find(f>0e4 & f<0.5e4);
lam = find(a>1 & a<80);
lfc = find(f>1.4e4 & f<1.8e4);
lac = find(a>100 & a<130);


figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 14 8])
mesh(a(lac),f(lfc),abs(Saf(lfc,lac))),
view(-26,42)
xlabel('\alpha$ [Hz]')
ylabel('f [Hz]')
title('Cyclic Modulation Spectrum'),

figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 14 8])
mesh(a(lam),f(lfm),abs(Saf(lfm,lam))),
view(-26,42)
xlabel('\alpha$ [Hz]')
ylabel('f [Hz]')
title('Cyclic Modulation Spectrum'),


%% CMS resuduo
fracovlp = .75;     % fraction of overlap (should be greater than or equal to .75)
Nw = 2^7;
Nfft = Nw;

Noverlap = fix(fracovlp*Nw);
R = Nw - Noverlap;
Win = hanning(Nw);
[Stfr,f,t] = specgram(xr_fix,Nfft,fs,Win,Noverlap);
Lt = length(t);

Stfr = abs(Stfr).^2;
Safr = 1/Lt*(fft(Stfr',Lt))';
Safr = Safr(1:Nfft/2,1:fix(Lt/2));

% Plot cms
f = f(1:Nfft/2);
a = (0:fix(Lt/2)-1)*fs/(Lt*Nw*(1-fracovlp));


lfm = find(f>0e4 & f<0.5e4);
lam = find(a>1 & a<80);
lfc = find(f>1.4e4 & f<1.8e4);
lac = find(a>100 & a<130);
```

```
figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 14 8])
mesh(a(Iac),f(Ifc),abs(Safr(Ifc,Iac))),
view(-26,42)
xlabel('\alpha$ [Hz]')
ylabel('f [Hz]')
title('Cyclic Modulation Spectrum'),

figure,
set(gcf,'Units','centimeters')
set(gcf,'Position',[2 10 14 8])
mesh(a(Iam),f(Ifm),abs(Safr(Ifm,Iam))),
view(-26,42)
xlabel('\alpha [Hz]')
ylabel('f [Hz]')
title('Cyclic Modulation Spectrum'),
```