CT.2306 : Signal & Systems II

Project Report:

Processing motion signals from a PTZ camera

Ulysse MERAD, Ruben LEGRANDJACQUES,
Fabrice LIN, Anwar AL-BITAR

Under the supervision of A.BAHLOUL and I.AYAJI

December 12th, 2023

École d'ingénieurs du numérique

Abstract

Sum-up of the project

Contents

1 Data Visualization

1) *Loading data*

We first use *load('data-proj.mat')* to load variables from data file. Then using *whos* we can list all variables. We have their name and size. Here is the result :

```
  >> whos
Name          Size                   Bytes  Class

omega         1x20001               160008  double
t             1x20001               160008  double
```

Listing 1: Loaded variables

2) *Plotting the data*

We can not use the signal as it is. Graphically it is impossible to analyze. Either it is too noisy or the window is too large in order to see enough details of the signal. This is a continous (analog) signal. Electronic control devices requires digital signals.

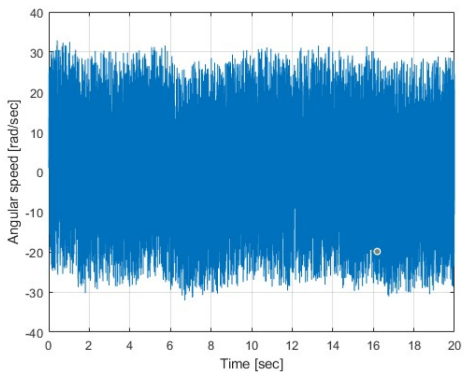

Figure 1: Angular speed as a function of time

```
%% Plot of angular speed
fig=1
figure(fig)
plot()
grid on
xlabel('Time [sec]')
ylabel('Angular speed [rad/sec
   ]')
```

Listing 2: Code for Figure 2

2 Analog Filtering

3) *Sampling period T_{e_1}*

We deduce the Sampling period T_{e_1} with the subtraction of two consecutive values of t. Also, graphically we observe the run between 2 straight line in the signal if we zoom. $\mathbf{T_{e_1} = 1.10^{-3}sec}$

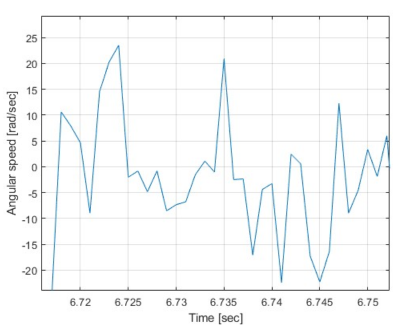

Figure 2: Same signal with zoom

```
%% Sampling period
Te1=t(5)-t(4)

>>Te1 =

1.0000e-03
```

Listing 3: Code for Figure 2

4) *DFT*

Using code from workshop 5 and Help Project, we obtain this plot.

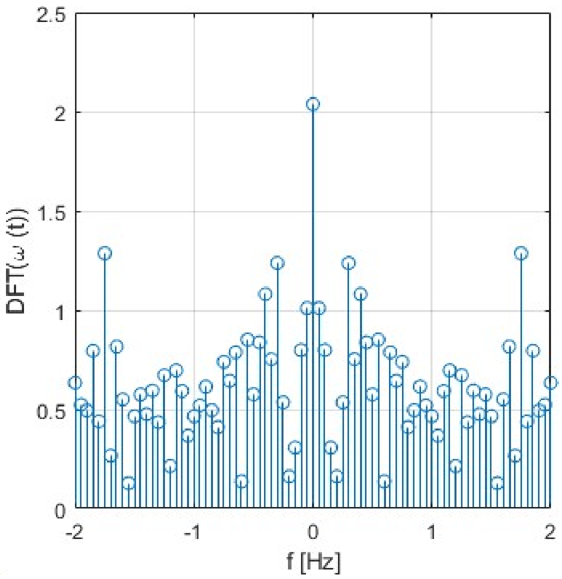

Figure 3: DFT plot of $\omega(t)$

```
Te2= 0.05;%sampling period
Fe2=1/Te2;%sampling frequency
Tf=t(end);%singal duration
N=Tf/Te2;%number of samples
%(N=Duration/sampling period)
%frequency vector
f1=-Fe2*(N/2-1)/N:Fe2/N:0;
f2=Fe2/N:Fe2/N:(N/2)*Fe2/N;
f=[f2,f1];
O= zeros(1,N);
for m=1:N
  for k=1:N
    O(m)=O(m)+omega(k)*exp(-j*2*
        pi*m*k/N);
  end
end
figure(fig+1);
stem(f,abs(O)/N)
grid on
xlim([-2,2])
xlabel('f [Hz]')
ylabel('DFT(\omega (t))')
```

Listing 4: Code for Figure 3

5) *Maximal Frequency F_{max} of the signal*

I don't know yet exactly what to do maybe the maximum frequency $F_e = 2Hz$.

6) *First order low-pass analog filter of a unit gain*

We first create a unit gain low-pass filter with a cutoff frequency $f_c = 2Hz$ **using Matlab.**

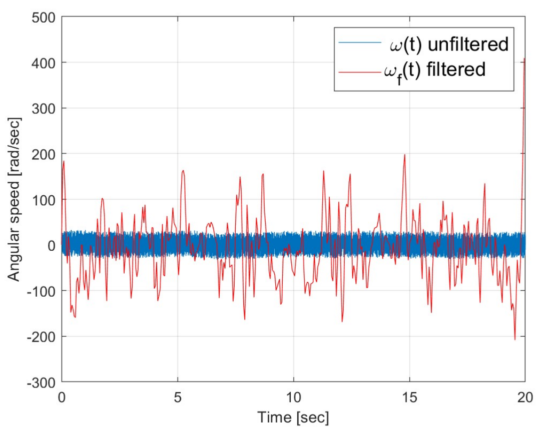

Figure 4: Plot of unfiltered and filtered $\omega(t)$ signal

```
%create the new time vector
t1=0:Te2:t(end)-Te2;
%% filter design
fc1=2;
H1=tf(1,[1/(2*pi*fc1)  1]);
Of=lsim(H1,O,t1);
%% plot of filtered signal
figure(1);
plot(t1,Of,'r')
grid on
legend(' \omega(t) unfiltered','
   \omega_{f}(t) filtered')
```

Listing 5: Code for Figure 4

We may notice that we observe on the red filtered signal that its amplitude is bigger. We do not know exactly why. Bode diagram show a magnnitude near 0 from low frequencies.

Now, let use **Simulink** to filter the signal $\omega(t)$.

Need to transpose the row time vector to a column one. Then create a 2 columns matrix with $\omega(t)$ and t in each. This matrix will be imported in Simulink

Listing 6: Code for Figure 3

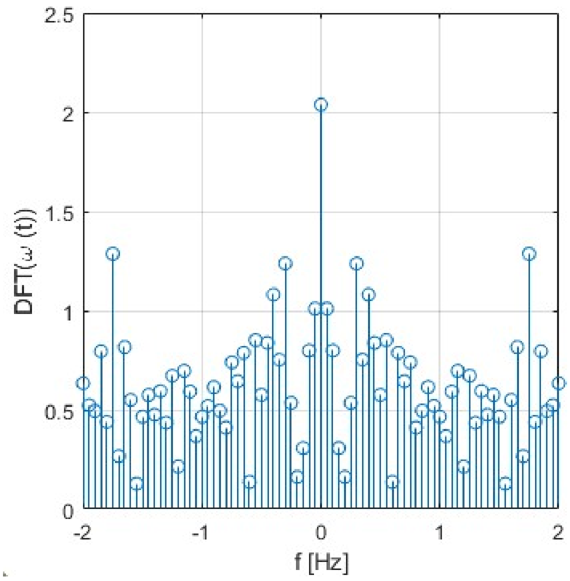

Figure 5: DFT plot of $\omega(t)$

7) *question subject*

3 Sampling

8) *question subject*

9) *question subject*

10) *question subject*

11) *question subject*

12) *question subject*

4 Angular position and acceleration

13) *question subject*

14) *question subject*

15) *question subject*

5 Digital filtering

16) *question subject*

17) *question subject*

18) *question subject*