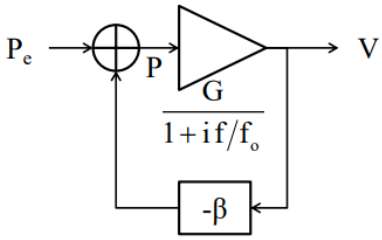
Exercise 04 - Pressure Transducer

- Gugliemo Grillo

Exercise 3

- A system consists of a pressure transducer, converting a pressure signal P into a voltage one V, with an operating range of ±1 Pa. Within this operating range, the transducer behaves like a simple low pass, with a gain of G=100 V/Pa, and a roll-off frequency of f_o= 10 Hz.
- To operate the transducer, a feedback loop sends back its output to a piezoelectric crystal, able to exert a pressure $P_{fb} = -\beta V$ on the transducer, with β=1Pa/V.



- Calculate:
  - The transfer function $P_e \rightarrow V$ of the closed loop system.
  - The maximum peak value allowed for a sinusoidal external pressure signal, in order to stay within 10% of saturation. Give the answer as a function of frequency.

Transfer function

We call P_{fb} the preassure coming from the output V and passing throught the $-\beta$ stage. the relevant equations of the system are:

$$\begin{aligned} P_{fb} &= -\beta V \\ P_e + P_{fb} &= P \\ V &= P\frac{G}{1+if/f_0} \end{aligned}$$

Solving for V/P_e we obatain:

$$h(f) = \frac{V}{P_e} = \frac{G_c}{1+G_c\beta} = \frac{G}{1+if/f_0+\beta G}$$

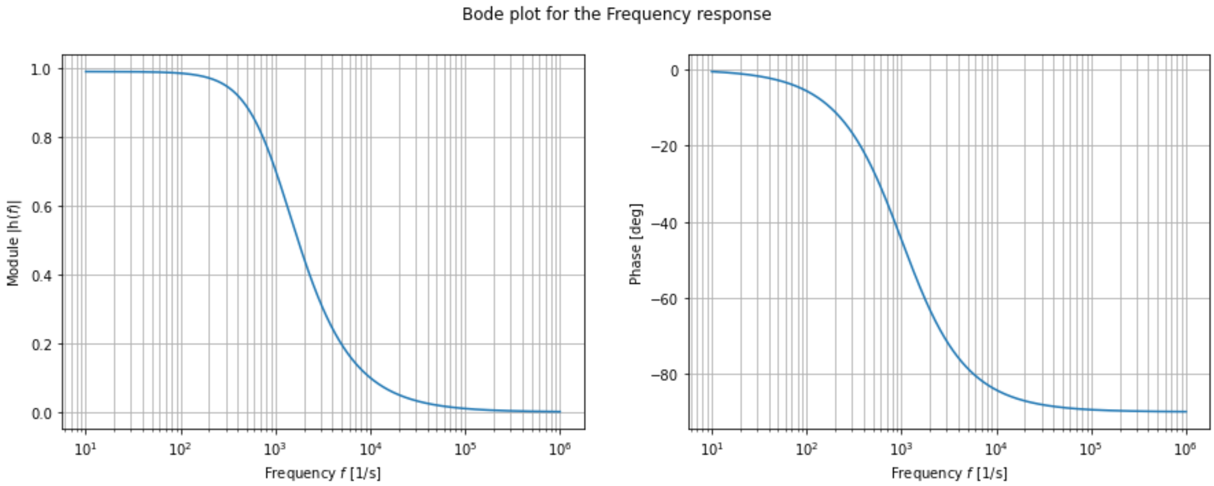
where we choose $G_c = \frac{G}{1+if/f_0}$ in order to preserve the canonical gain equation.

In [1]:

```
# packages used
from scipy import signal
import numpy as np
import matplotlib.pyplot as plt

# Given constans
OR = 1 # \pm, [Pa], operating range
G = 100 # V/Pa
F0 = 10 # Hz, roll of frequency
BETA = 1 # Pa/V

# Parameters redefinition
Gc = lambda f: G / (1+ 1j * f/F0)

# Show value
print("Gain at roll-off frequency {:.5}".format(Gc(F0)) )
```

Gain at roll-off frequency (50-50j)

In [7]:
```
def H(f):
    return G/(1+1j*f/F0 + BETA*G)

w = np.linspace(int(1e1), int(1e6), int(1e6))
h = H(w)
mag = np.abs(h)
phase = np.arctan(np.imag(h) / np.real(h))*180/np.pi

# Bode plot
fig, axis = plt.subplots(1,2, figsize=(15, 5))
fig.suptitle('Bode plot for the Frequency response')

axis[0].plot(w, mag)
axis[0].set_xscale('log')
axis[0].set_xlabel("Frequency $f$ [1/s]")
axis[0].set_ylabel("Module |h($f$)|")
axis[0].grid(True, which="both")

axis[1].plot(w, phase)
axis[1].set_xscale('log')
axis[1].set_xlabel("Frequency $f$ [1/s]")
axis[1].set_ylabel("Phase [deg]")
axis[1].grid(True, which="both")

plt.show()
```



Saturation

The operating range of the transducer is $P = \pm 1 Pa$. We will assume all the quantities to be positive. The relations between P, V and the input P_e are:

$$\begin{aligned} V &= \frac{G_c}{1+\beta G_c} P_e \\ V &= \frac{G}{1+if/f_0} P \end{aligned}$$

Which lead to:

$$P_e = P\frac{G}{1+if/f_0}\frac{1+\beta G_c}{G_c} = PG_c\frac{1+\beta G_c}{G_c} = P(1+\beta G_c)$$

We require to stay within 10% of saturation, so that

$$P_{eM} = \frac{9}{10}(1+\beta G_c)P_M = 0.9\frac{1+if/f_0+\beta G}{1+if/f_0}$$

In [9]:

```
def H(f):
    return 0.9*( 1+f*1j/F0+BETA*G )/(1+1j*f/F0)

w = np.linspace(0.05, 5000, int(1e6))
h = H(w)
mag = np.abs(h)
phase = np.arctan(np.imag(h) / np.real(h))*180/np.pi

# Bode plot
fig, axis = plt.subplots(1,2, figsize=(15, 5))
fig.suptitle('Bode plot for the Maximum input signal')

axis[0].plot(w, mag)
axis[0].set_xscale('log')
axis[0].set_xlabel("Frequency $f$ [1/s]")
axis[0].set_ylabel("Module of Maximum Input |$P_M$($f$)|")
axis[0].grid(True, which="both")

axis[1].plot(w, phase)
axis[1].set_xscale('log')
axis[1].set_xlabel("Frequency $f$ [1/s]")
axis[1].set_ylabel("Phase [deg]")
axis[1].grid(True, which="both")

plt.show()
```