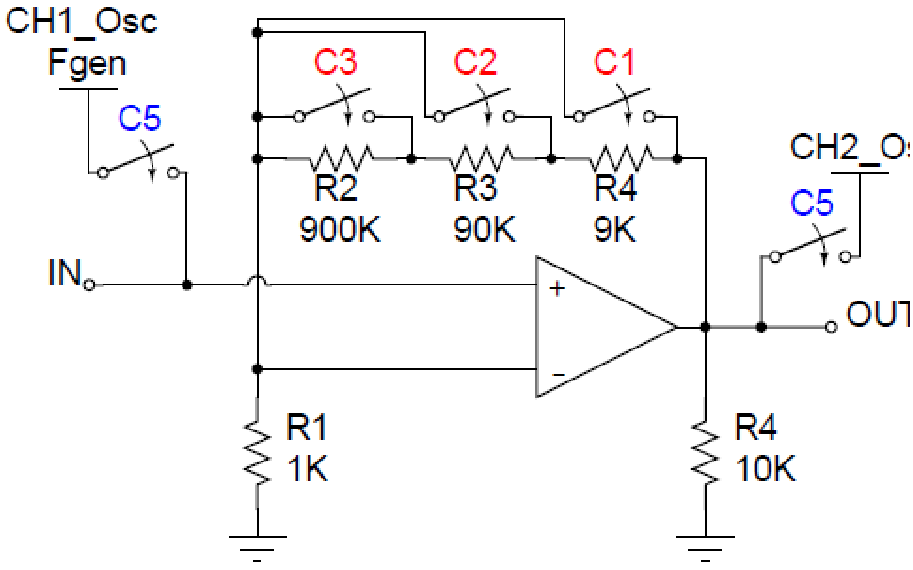
https://www.markdownguide.org/cheat-sheet/ (https://www.markdownguide.org/cheat-sh

Testarea Automată a Circuitelor

--- Îndrumar de Laborator ---

Lucrarea nr. 6 - Castigul amplificatorului



6.1.1 Montajul experimental:

Montajul experimental se bazează pe amplificatorul TL081P. Schema de principiu a aces prezentată în figura 6.1. Câștigul este reglabil prin intermediul conectoarelor C1… C3, d urmeaza: C1=OFF, C2=OFF, C3=OFF: A=1000 C1=OFF, C2=OFF, C3=ON: A=100 C1= C3=OFF: A=10 C1=ON, C2=OFF, C3=OFF: A=1

6.1.2 Procedura de masurare:

Folosind o buclă de tip „for” se va varia frecventa unui semnal sinusoidal si se va monito osciloscopul amplitudinea semnalului de la iesirea respectiv intrarea amplificatorului. Ra doua amplitudini reprezinta castigul amplificatorului. Se va trasa caracteristica castig vs. amplificatorului. Pentru măsurarea câștigului în funcție de frecvență, se va folosi oscilosc Tensiunile măsurate vor fi varf la varf (Vpp). Codul sursă prezentat în continuare este pe configurația G=100.

6.2 Codul sursa:

6.2.1 Initializarea instrumentelor si a interfetelor grafice

In [ ]:
```
#using TIVM; # libraria responsabila de comunicatia cu instrumentele si
#using Plots; # libraria responsabila de generarea de grafice
#using DataFrames; #
#using CSV; # libraria responsabila de salvarea datelor in format .csv
```

In [1]:
```
find_resources()
```

```
Found RIGOL TECHNOLOGIES,DS1104Z Plus,DS1ZD231200356,00.04.04.SP4 on ad
0::0x1AB1::0x04CE::DS1ZD231200356::INSTR
Found GW.Inc,GDM-8246,FW2.01 on address: ASRL1::INSTR
Found TIVM Relays v0.1 on address: ASRL6::INSTR
Found GW.Inc,PST-3201,I180085 ,FW1.00 on address: ASRL7::INSTR
```

In [35]:
```
#Ctrl+Enter =>  rulare cod
# dmm_handle = connect!("ASRL1::INSTR")
# psu_handle = connect!("ASRL5::INSTR")
scope_handle = connect!("USB0::0x1AB1::0x04CE::DS1ZD231200356::INSTR")
fgen_handle = scope_handle
relays_handle = connect!("ASRL6::INSTR")
relays = TIVM.Relays(relays_handle);

# dmm = TIVM.GDM8246(dmm_handle);
# psu = TIVM.PST3201(psu_handle);
fgen = TIVM.DS1000Z_FGEN(fgen_handle);#scope has integrated fgen
scope = TIVM.DS1000Z(scope_handle);
```

In [2]:
```
# Panouri frontale pentru instrumente
#@async start_gui(psu_handle = psu_handle, dmm_handle = dmm_handle, fge
@async start_gui(fgen = fgen, scope = scope);
```

6.2.2 Connfigurarea instrumentelor

In [36]:
```
# Generatorul de semnal
TIVM.write(fgen_handle, ":SOURce1:OUTPut1:STATe 0") #disable to reset
sleep(1)
set_wfm(fgen, "C1", "sinusoid")
TIVM.write(fgen_handle, ":SOURce1:OUTPut1:IMPedance FIFTy") #set output
#sleep(1)
set_freq(fgen, "C1", 1000)
#sleep(1)
set_amplit(fgen, "C1", 0.1)
#sleep(1)
set_offs(fgen, "C1", 0)
#sleep(1)
set_duty(fgen, "C1", 50)
TIVM.write(fgen_handle, ":SOURce1:OUTPut1:STATe 1")#turn output on
sleep(1)
```

In [37]:
```
# Osciloscopul
#autoset is hidden :P
#ch coupling - WIP
set_ch_position(scope, "CH1", 0)
set_ch_position(scope, "CH2", 0)
#ch probe - do we need this?
set_vertical_scale(scope, "CH1", 0.1)
set_vertical_scale(scope, "CH2", 5)
set_horizontal_scale(scope, 0.001)
set_trig_ch(scope, "CH1")
set_trig_mode(scope, "NORMAL") # "NORMAL"/"AUTO"
set_trig_level(scope, 0.005)
# do we need trigger coupling or slope?
# Masuratori - se pot configura maxim 5 pt ambele canale impreuna
set_meas(scope, "MEAS1", "CH1", "FREQuency")
set_meas(scope, "MEAS2", "CH1", "PK2pk")
set_meas(scope, "MEAS3", "CH2", "FREQuency")
set_meas(scope, "MEAS4", "CH2", "PK2pk")
set_meas(scope, "MEAS5", "CH1", "PERIod")
```

In [38]:
```
set_state(relays,"C1","off")
set_state(relays,"C2","off")
set_state(relays,"C3","off")
set_state(relays,"C4","off")
set_state(relays,"C5","off")
set_state(relays,"C6","off")
set_state(relays,"C7","off")
set_state(relays,"C8","off")
set_state(relays,"C9","off")
set_state(relays,"C10","off")
set_state(relays,"C11","off")
```

6.2.3 Definirea stimulilor si a variabilelor auxiliare

In [39]:
```
freq = exp10.(1:0.25:7) # create a log vector of frequencies (incepem d
fgen_amplit = 5
dc_gain = 1 # setat din relee
dc_gain_db = 0
```

Out[39]:
```
0
```

6.2.4 Bucla de masurare

In [18]:

```
#dc_gain = 100 # setat din relee
#dc_gain_db = 40
#fgen_amplit = 0.05
in_freq1 = []
in_amplit_meas1 = []
out_amplit_meas1 = []
gain1 = []
gain1_db = []
crt_meas_amplit_out = 0
crt_meas_amplit_in = 0
bandwidth = 0

set_state(relays,"C5","on")
set_state(relays,"C1","off")
set_state(relays,"C2","off")
set_state(relays,"C3","on")

set_vertical_scale(scope, "CH1", fgen_amplit/2)
set_vertical_scale(scope, "CH2", dc_gain*fgen_amplit) #facem asta ca sa
set_amplit(fgen, "C1", fgen_amplit)
for crt_freq in freq
    set_freq(fgen, "C1", crt_freq)
    tbase = 1/crt_freq
    set_horizontal_scale(scope, tbase)
    sleep(2) # fgen and osc are slow to respond

    crt_meas_amplit_in = get_meas_data(scope, "MEAS2")
    crt_meas_amplit_out = get_meas_data(scope, "MEAS4")

    crt_gain = crt_meas_amplit_out/crt_meas_amplit_in
    crt_gain_db = 20*log10(crt_gain)

    if (crt_gain_db >= dc_gain_db - 3)
        bandwidth = crt_freq
    end

    # store crt stimuli value
    push!(in_freq1, crt_freq)

    # store crt measurement value
    push!(in_amplit_meas1, crt_meas_amplit_in)
    push!(out_amplit_meas1,  crt_meas_amplit_out)
    push!(gain1, crt_gain)
    push!(gain1_db, crt_gain_db)

    # print info to console
    @info "crt_freq=$crt_freq, crt_meas_amplit_in=$crt_meas_amplit_in,
    @info "bandwidth=$bandwidth"
end
```

```
┌ Info: crt_freq=10.0, crt_meas_amplit_in=9.9e37, crt_meas_amplit_out=9
_gain=1.0, crt_gain_db=0.0
└ @ Main In[18]:44
┌ Info: bandwidth=0
└ @ Main In[18]:45
┌ Info: crt_freq=17.78279410038923, crt_meas_amplit_in=0.104, crt_meas_
t=11.2,crt_gain=107.6923076923077, crt_gain_db=40.643693667428025
└ @ Main In[18]:44
┌ Info: bandwidth=17.78279410038923
└ @ Main In[18]:45
┌ Info: crt_freq=31.622776601683793, crt_meas_amplit_in=0.105, crt_meas
ut=11.2,crt_gain=106.66666666666666, crt_gain_db=40.56057447200487
└ @ Main In[18]:44
┌ Info: bandwidth=31.622776601683793
└ @ Main In[18]:45
┌ Info: crt_freq=56.23413251903491, crt_meas_amplit_in=0.104, crt_meas_
t=11.2,crt_gain=107.6923076923077, crt_gain_db=40.643693667428025
└ @ Main In[18]:44
┌ Info: bandwidth=56.23413251903491
```

In [19]:
```
bandwidth1= bandwidth
```

Out[19]: 17782.794100389227

In [20]:
```
AB1 = dc_gain*bandwidth1
```

Out[20]: 1.7782794100389227e6

In [26]:
```
#dc_gain = 10 # setat din relee
#dc_gain_db = 20
#fgen_amplit = 0.5
in_freq2 = []
in_amplit_meas2 = []
out_amplit_meas2 = []
gain2 = []
gain2_db = []
crt_meas_amplit_out = 0
crt_meas_amplit_in = 0
bandwidth = 0

set_state(relays,"C5","on")
set_state(relays,"C1","off")
set_state(relays,"C2","on")
set_state(relays,"C3","on")

set_vertical_scale(scope, "CH1", fgen_amplit/2)
set_vertical_scale(scope, "CH2", dc_gain*fgen_amplit) #facem asta ca sa
set_amplit(fgen, "C1", fgen_amplit)
for crt_freq in freq
    set_freq(fgen, "C1", crt_freq)
    tbase = 1/crt_freq
    set_horizontal_scale(scope, tbase)
    sleep(2) # fgen and osc are slow to respond

    crt_meas_amplit_in = get_meas_data(scope, "MEAS2")
    crt_meas_amplit_out = get_meas_data(scope, "MEAS4")

    crt_gain = crt_meas_amplit_out/crt_meas_amplit_in
    crt_gain_db = 20*log10(crt_gain)

    if (crt_gain_db >= dc_gain_db - 3)
        bandwidth = crt_freq
    end

    # store crt stimuli value
    push!(in_freq2, crt_freq)

    # store crt measurement value
    push!(in_amplit_meas2, crt_meas_amplit_in)
    push!(out_amplit_meas2,  crt_meas_amplit_out)
    push!(gain2, crt_gain)
    push!(gain2_db, crt_gain_db)

    # print info to console
    @info "crt_freq=$crt_freq, crt_meas_amplit_in=$crt_meas_amplit_in,
    @info "bandwidth=$bandwidth"
end
```

```
┌ Info: crt_freq=1.7782794100389227e6, crt_meas_amplit_in=1.0, crt_meas
ut=1.8,crt_gain=1.8, crt_gain_db=5.105450102066121
└ @ Main In[26]:47
┌ Info: bandwidth=177827.94100389228
└ @ Main In[26]:48
┌ Info: crt_freq=3.162277660168379e6, crt_meas_amplit_in=1.0, crt_meas_
t=0.8,crt_gain=0.8, crt_gain_db=-1.938200260161128
└ @ Main In[26]:47
┌ Info: bandwidth=177827.94100389228
└ @ Main In[26]:48
┌ Info: crt_freq=5.623413251903491e6, crt_meas_amplit_in=0.97, crt_meas
ut=0.6,crt_gain=0.6185567010309279, crt_gain_db=-4.172409677652024
└ @ Main In[26]:47
┌ Info: bandwidth=177827.94100389228
└ @ Main In[26]:48
┌ Info: crt_freq=1.0e7, crt_meas_amplit_in=0.87, crt_meas_amplit_out=0.
n=0.4597701149425288, crt_gain_db=-6.749185225813122
└ @ Main In[26]:47
┌ Info: bandwidth=177827.94100389228
```

In [40]:
```
#fgen_amplit = 5
#dc_gain = 1 # setat din relee
#dc_gain_db = 0
in_freq3 = []
in_amplit_meas3 = []
out_amplit_meas3 = []
gain3 = []
gain3_db = []
crt_meas_amplit_out = 0
crt_meas_amplit_in = 0
bandwidth = 0

set_state(relays,"C5","on")
set_state(relays,"C1","on")
set_state(relays,"C2","on")
set_state(relays,"C3","on")

set_vertical_scale(scope, "CH1", fgen_amplit/2)
set_vertical_scale(scope, "CH2", dc_gain*fgen_amplit) #facem asta ca sa
set_amplit(fgen, "C1", fgen_amplit)
for crt_freq in freq
    set_freq(fgen, "C1", crt_freq)
    tbase = 1/crt_freq
    set_horizontal_scale(scope, tbase)
    sleep(2) # fgen and osc are slow to respond

    crt_meas_amplit_in = get_meas_data(scope, "MEAS2")
    crt_meas_amplit_out = get_meas_data(scope, "MEAS4")

    crt_gain = crt_meas_amplit_out/crt_meas_amplit_in
    crt_gain_db = 20*log10(crt_gain)

    if (crt_gain_db >= dc_gain_db - 3)
        bandwidth = crt_freq
    end

    # store crt stimuli value
    push!(in_freq3, crt_freq)

    # store crt measurement value
    push!(in_amplit_meas3, crt_meas_amplit_in)
    push!(out_amplit_meas3,  crt_meas_amplit_out)
    push!(gain3, crt_gain)
    push!(gain3_db, crt_gain_db)

    # print info to console
    @info "crt_freq=$crt_freq, crt_meas_amplit_in=$crt_meas_amplit_in,
    @info "bandwidth=$bandwidth"
end
```

```
┌ Info: crt_freq=1.7782794100389227e6, crt_meas_amplit_in=5.1, crt_meas
ut=3.2,crt_gain=0.627450980392157, crt_gain_db=-4.048403955560607
└ @ Main In[40]:47
┌ Info: bandwidth=1.0e6
└ @ Main In[40]:48
┌ Info: crt_freq=3.162277660168379e6, crt_meas_amplit_in=5.1, crt_meas_
t=1.6,crt_gain=0.3137254901960785, crt_gain_db=-10.06900386884023
└ @ Main In[40]:47
┌ Info: bandwidth=1.0e6
└ @ Main In[40]:48
┌ Info: crt_freq=5.623413251903491e6, crt_meas_amplit_in=4.9, crt_meas_
t=1.0,crt_gain=0.2040816326530612, crt_gain_db=-13.803921600570275
└ @ Main In[40]:47
┌ Info: bandwidth=1.0e6
└ @ Main In[40]:48
┌ Info: crt_freq=1.0e7, crt_meas_amplit_in=4.5, crt_meas_amplit_out=0.8
=0.17777777777777778,   crt_gain_db=-15.002450535668002
└ @ Main In[40]:47
┌ Info: bandwidth=1.0e6
```

In [29]:
```
bandwidth2=bandwidth
```

Out[29]:
```
177827.94100389228
```

In [30]:
```
AB2 = dc_gain*bandwidth2
```

Out[30]:
```
1.778279410038923e6
```

In [41]:
```
bandwidth3=bandwidth
```

Out[41]:
```
1.0e6
```

In [42]:
```
AB3 = dc_gain*bandwidth2
```

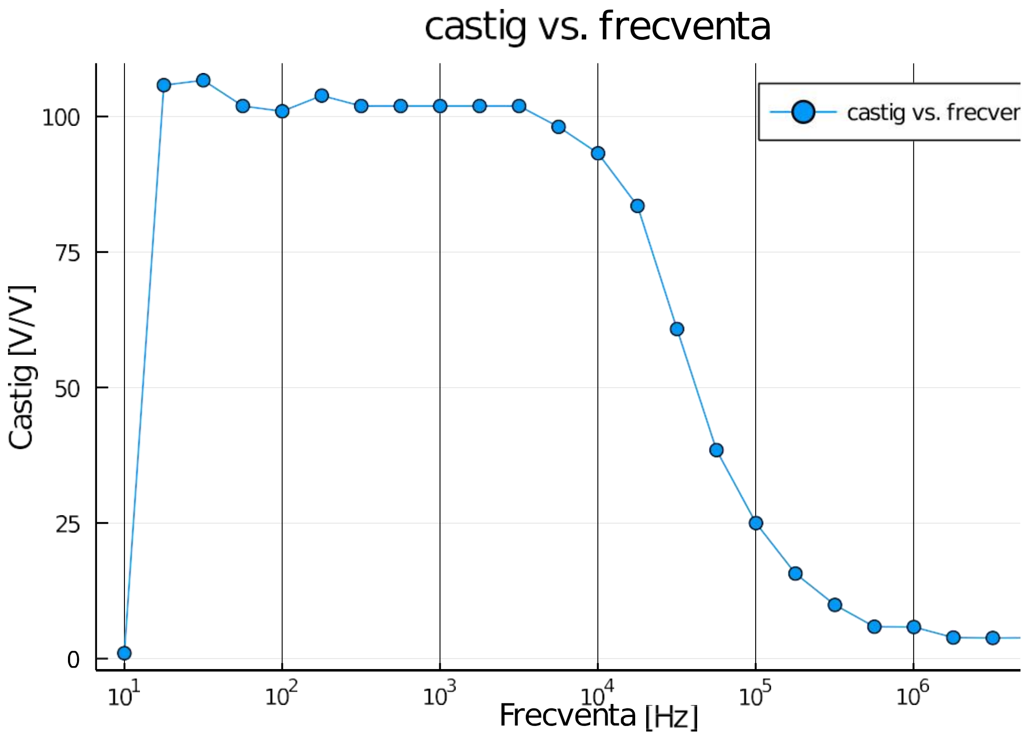
Out[42]:
```
177827.94100389228
```

6.2.5 Oprirea instrumentelor

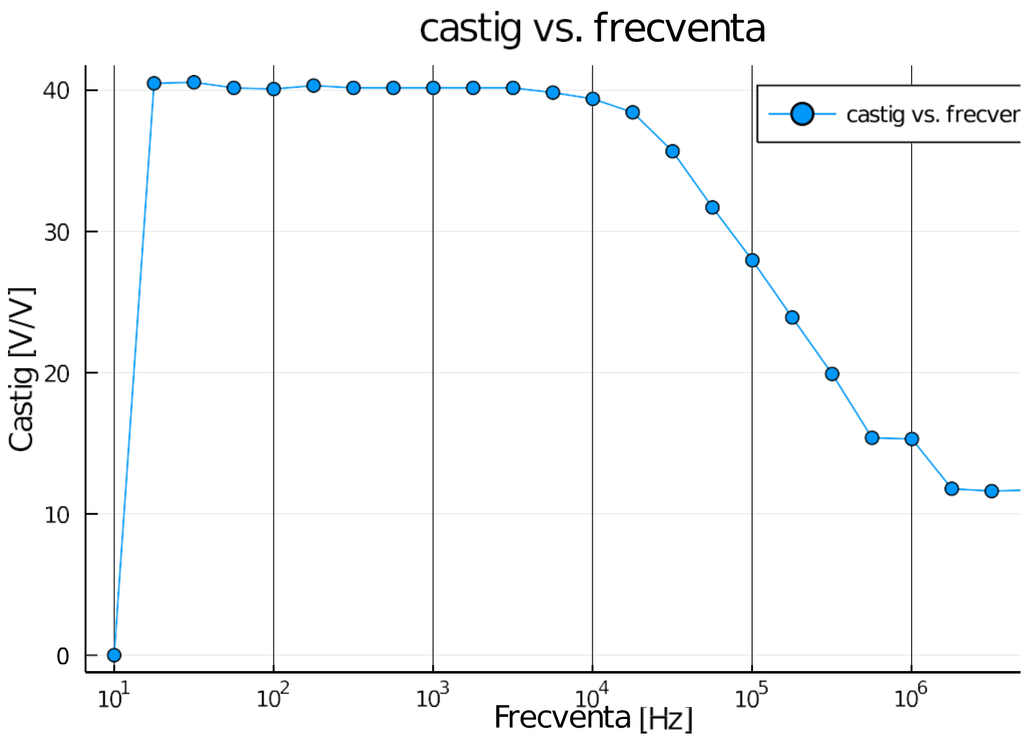
In [10]:
```
set_amplit(fgen, "C1", 0.02)
TIVM.write(fgen_handle, ":SOURce1:OUTPut1:STATe 0")
```

6.2.6 Generarea caracteristicilor, dupa incheierea masuratorilor

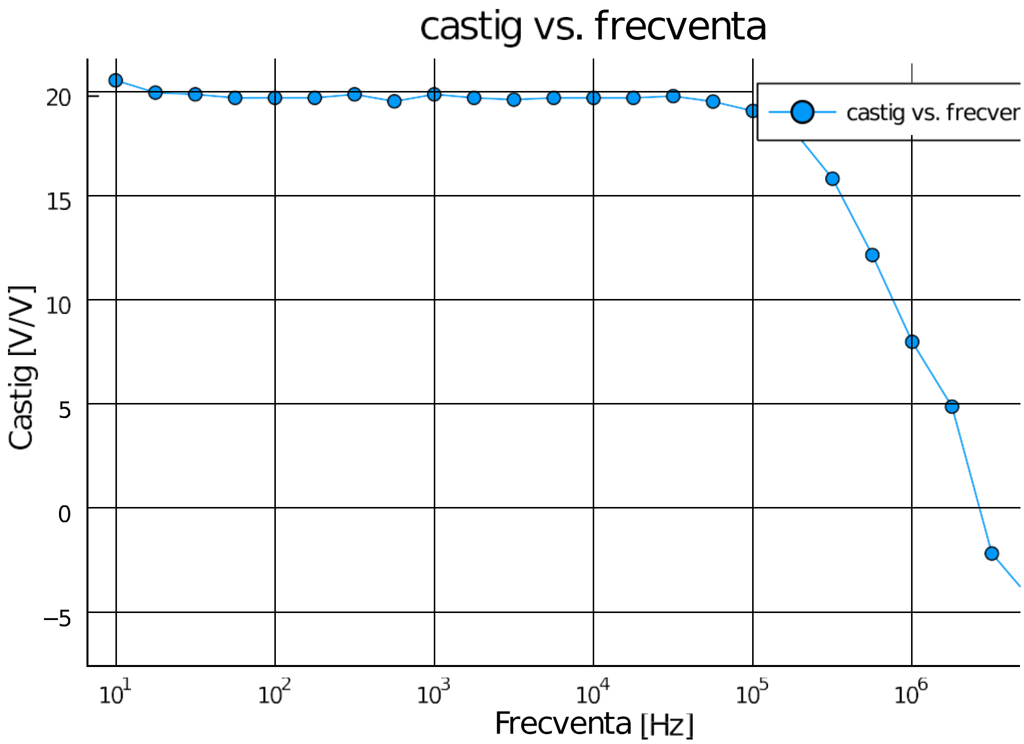
In [14]:
```
# castig vs frecventa
h1=plot(in_freq1, gain1; xaxis=:log, markershape=:circle, label="castig
title!("castig vs. frecventa");
xlabel!("Frecventa [Hz]");
ylabel!("Castig [V/V]");
display(h1)
```



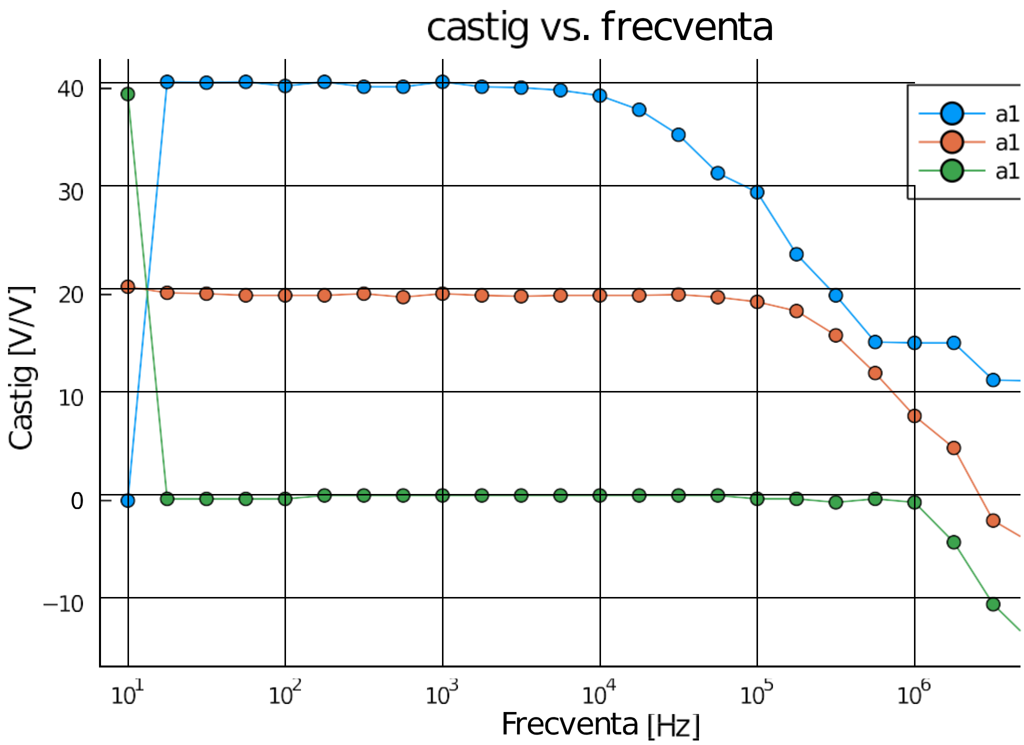
In [15]:
```
# castig DB vs frecventa
h1=plot(in_freq1, gain1_db; xaxis=:log, markershape=:circle, label="cas
title!("castig vs. frecventa");
xlabel!("Frecventa [Hz]");
ylabel!("Castig [V/V]");
display(h1)
```



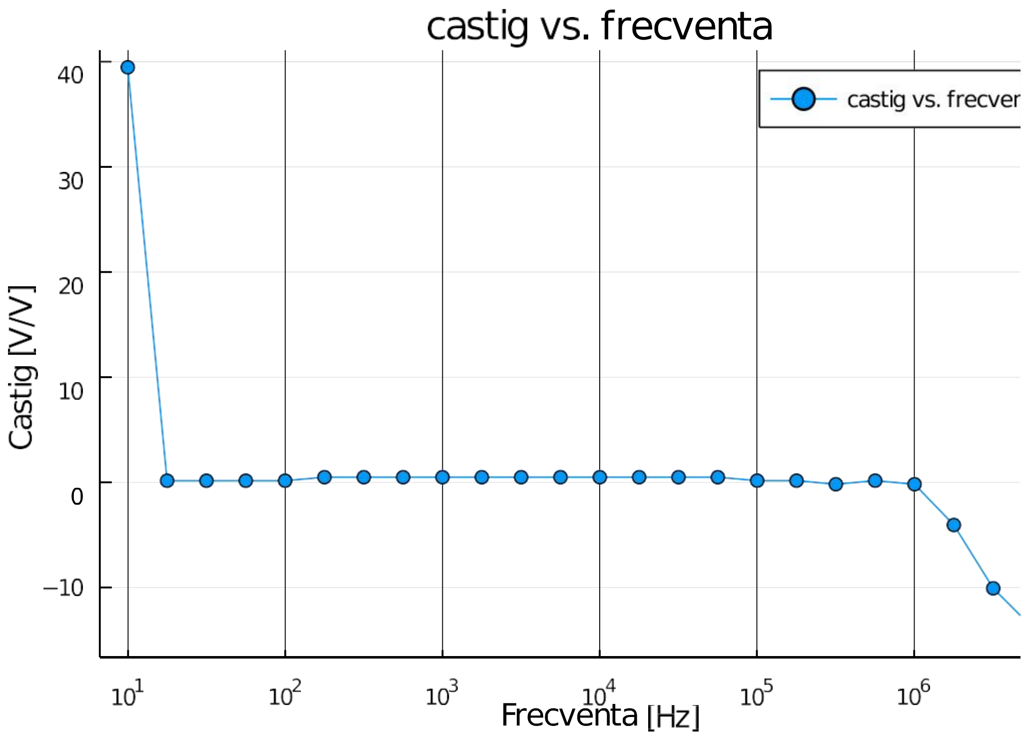
In [27]:
```
# castig DB vs frecventa
h1=plot(in_freq2, gain2_db; xaxis=:log, markershape=:circle, label="cas
title!("castig vs. frecventa");
xlabel!("Frecventa [Hz]");
ylabel!("Castig [V/V]");
display(h1)
```



In [44]:
```
# castig DB vs frecventa
h1=plot(in_freq1, gain1_db; xaxis=:log, markershape=:circle, label="a10
h1=plot!(in_freq2, gain2_db; xaxis=:log, markershape=:circle, label="a1
h1=plot!(in_freq3, gain3_db; xaxis=:log, markershape=:circle, label="a1
title!("castig vs. frecventa");
xlabel!("Frecventa [Hz]");
ylabel!("Castig [V/V]");
display(h1)
```



In [43]:
```
# castig DB vs frecventa
h1=plot(in_freq3, gain3"Frecventa [Hz]" => in_freq,_db; xaxis=:log, mar
title!("castig vs. frecventa");
xlabel!("Frecventa [Hz]");
ylabel!("Castig [V/V]");
display(h1)
```



6.2.7 Salvarea datelor in fisierul .csv

In [47]:
```
df = DataFrame(
    "Tip  amplificare"  =>  "100",
    "Frecventa [Hz]" => in_freq1,
    "Amplitudinea la intrare [V]" => in_amplit_meas1,
    "Amplitudinea la iesire [V]" => out_amplit_meas1,
    "Castig [V/V]" => gain1,
    "Castig [dB]" => gain1_db,
    )
CSV.write("0006_castig_amplif.csv", df)
```

Out[47]: "0006_castig_amplif.csv"

6.2.8 Deconectarea instrumentelor

In [10]:
```
#disconnect!(fgen_handle)
disconnect!(scope_handle)
```

Out[10]: 0

6.3 Desfasurarea lucrarii:

1. Măsurați caracteristicile câștig-frecvență pentru G=1 (Vin=3Vrms), G=10 (Vin=0.3Vr (Vin=0.03Vrms). Comparați precizia măsurătorilor efectuate cu osciloscopul și multi toate caracteristicile pe același grafic.
2. Afisati castigul in decibeli.
3. Introduceti o coloana suplimentara in CSV cu castigul in decibeli.
4. Calculati frecventa de taiere si panta amplificatorului.
5. Masurati doua din cele 4 configuratii ale amplificatorului. Introduceti datele experiem acelasi tabel..