hw3

December 18, 2024

[1]:
```
import numpy as np
```

[2]:
```
def parallel(*resistances):
    return 1 / sum(1/r for r in resistances)
```

[3]:
```
# 3-1
clm = 0.1
CGS = 20 * 10**-15 # F
CGD = 10 * 10**-15 # F
CDB = 5 * 10**-15 # F
I_D = 1 * 10**-3 # A
gm = 20 * 10**-3 # S
Rs = 1 * 10**3 # Ohm

ro = 1 / (clm * I_D)

Rout = parallel(1 / gm, ro, ro)
Av = - gm * parallel(1 / gm, ro, ro)
Cx = CGD * (1 - Av)
Cy = CGD * (1 - 1/Av)
Cin = Cx + CGS
Cout = Cy + CDB + CGS
win = 1 / (Rs * Cin)
wout = 1 / (Rout * Cout)
fin = win / (2 * np.pi)
fout = wout / (2 * np.pi)
np.format_float_scientific(fin, precision=2), np.format_float_scientific(fout,␣
 ↪precision=2)
```

[3]:
```
('3.99e+09', '7.13e+10')
```

[4]:
```
# 3-2
Rsig = 100 * 10**3 # Ohm
gm = 2 * 10**-3 # S
RL = 5 * 10**3 # Ohm
ro = 20 * 10**3 # Ohm
CL = 40 * 10**-15 # F
CGS = 30 * 10**-15 # F
```

```
CGD = 5 * 10**-15 # F

Av = - gm * parallel(ro, RL)
Rout = parallel(ro, RL)
Cx = CGD * (1 - Av)
Cy = CGD * (1 - 1/Av)
Cin = Cx + CGS
Cout = Cy + CL
fin = 1 / (2 * np.pi * Rsig * Cin)
fout = 1 / (2 * np.pi * Rout * Cout)
np.format_float_scientific(fin, precision=2), np.format_float_scientific(fout,␣
 ↪precision=2)
```

```
[4]: ('2.12e+07', '8.72e+08')
```

```
[5]: # 3-3
     # (a)
     CGS = 30 * 10**-15 # F
     CL = 40 * 10**-15 # F
     gm = 2 * 10**-3 # S
     RL = 5 * 10**3 # Ohm

     w = 1 / (RL * CL)
     np.format_float_scientific(w, precision=2)
```

```
[5]: '5.e+09'
```

```
[6]: # (b)
     db = 20 * np.log10(gm ** 2 * RL ** 2)
     w1 = 1 / (RL * (CL + CGS))
     w2 = 1 / (RL * CL)
     np.format_float_scientific(db, precision=2), np.format_float_scientific(w1,␣
      ↪precision=2), np.format_float_scientific(w2, precision=2)
```

```
[6]: ('4.e+01', '2.86e+09', '5.e+09')
```

```
[7]: # 3-4
     PCGS = 80 * 10**-15 # F
     PCGD = 70 * 10**-15 # F
     PCDB = 10 * 10**-15 # F
     NCGS = 50 * 10**-15 # F
     NCGD = 40 * 10**-15 # F
     NCDB = 10 * 10**-15 # F
     gm = 10 * 10**-3 # S
     Rs = 2 * 10**3 # Ohm

     Av = -10
```

```
Cx = NCGD * (1 - Av)
Cy = NCGD * (1 - 1/Av)
Cin = NCGS + Cx
Cout = NCDB + Cy + PCDB + PCGS
Rout = np.abs(Av / gm)
fin = 1 / (2 * np.pi * Rs * Cin)
fout = 1 / (2 * np.pi * Rout * Cout)
np.format_float_scientific(fin, precision=2), np.format_float_scientific(fout, precision=2)
```

[7]:
```
('1.62e+08', '1.11e+09')
```

[8]:
```
# 3-5
# Cn = - (1 - Av) / (Cx)
Cn = np.abs((Cx + NCGS) / (1 - np.abs(Av)))
np.format_float_scientific(Cn, precision=2)
```

[8]:
```
'5.44e-14'
```