

# Aplicație laborator IV

## Teoria Sistemelor, Seria CD

Dorel, student la Politehnică pasionat de mașini, electronică și echipamente audio, a achiziționat recent de la o bunicuță din Germania o nouă **mașină**.

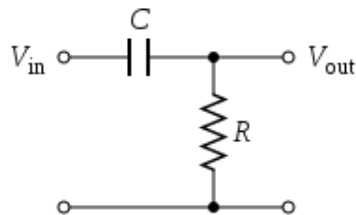
Să se realizeze următoarele sarcini:

- (i) Realizați o implementare pentru funcția

```
function y = apply_filter(tf, u, fs)
```

care primește o funcție de transfer, un semnal audio de intrare și o frecvență de eșantionare și întoarce semnalul rezultat la ieșirea sistemului cu acea funcție de transfer. (1pct)

- (ii) Dorel, audiofil fiind, se decide să testeze sistemul audio al noii mașini folosind fișierul *bass\_test.wav*. Nesațisfăcut fiind de performanță, acesta decide să filtreze semnalul audio cu un filtru high-pass cu o frecvență de cut-off de 100Hz. El știe de la electronică cum arată un astfel de filtru și că funcția de transfer este



$$H(s) = \frac{sRC}{1 + sRC}.$$

Găsiți o pereche (R, C) care să satisfacă condițiile filtrului și, folosind funcția de la punctul (i), observați rezultatul pe fișierul *bass\_test.wav*. (2pct)

- (iii) Neavând componentele necesare la îndemână, acesta decide să folosească un circuit găsit prin cutia cu electronice și aproape își distruge sistemul audio. Inspectând mai atent, acesta observă funcția de transfer

$$H(s) = \frac{s^2}{s^2 + 62.83s + 394800}.$$

Folosindu-vă de diagrama Bode și de spectrul semnalului rezultat, calculați și evidențiați parametrul greșit din design-ul acestui filtru. Propuneți și motivați o îmbunătățire.

**Atenție! Nu ascultați rezultatul!** (2pct)

- (iv) Cu sistemul audio încă intact, acesta decide să asculte melodia sa preferată. Având urechea fină, acesta observă că frecvențele între 1kHz și 5kHz se aud mai încet. Construiți, prin compunerea a două sau mai multe filtre, un filtru band-pass care să amplifice cu cel puțin 5 decibeli plaja de frecvențe și aplicați-l pe o secvență de 10s din melodia voastră preferată (în format wav, cu o frecvență de eșantionare de 48kHz, mono). În cazul în care nu puteți folosi melodia preferată, se va folosi fișierul *supernova.wav*. (2.5pct)
- (v) Pentru a-i face o farsă lui Dorel, Gigel i-a corupt întregul playlist adăugând un țuit constant peste toate melodiile. Folosind funcția *merge\_sound\_files* cu o valoare alpha recomandată de 45/50 și fișierul *beep.wav*, corupeți o secvență de 10s din melodia voastră preferată (în format wav, cu o frecvență de eșantionare de 48kHz, mono). Determinați frecvența țuitului și construiți un filtru notch pentru a o filtra. În cazul în care nu puteți folosi melodia preferată, se va folosi fișierul *supernova.wav*. (2.5pct)

---

**Indicații:**

- (i) Frecvența de eșantionare este folosită pentru a determina pasul vectorului de timp.
  - (ii) Toate sample-urile audio folosite vor avea o durată de 10s, eșantionate la 48kHz, mono și vor fi în formatul WAV.
  - (iii) Sunt puse la dispoziție funcțiile *play\_file* (pentru a citi un fișier audio și opțional a-l asculta), *play\_signal* (pentru a asculta un semnal audio), *plot\_fft* (pentru a observa spectrul semnalului audio) și *merge\_sound\_files* (pentru a suprapune două fișiere audio).
  - (iv) Explicațiile cerute la fiecare subpunct vor fi scrise sub formă de comentarii în Matlab/Octave în codul din aplicație corespunzător subpunctului respectiv.
  - (v) Pentru manipularea de fișiere audio (decupare, convertire etc.) recomandăm [Audacity](#).
  - (vi) **Pentru fiecare subpunct afișați diagrama Bode a filtrului rezultat și spectrul semnalului audio rezultat după aplicarea acestuia. În plus, pentru (iv) și (v) afișați spectrul semnalului audio înainte de aplicarea filtrului.**
-