Assignment 3

Digital Signal processing

IIR filters

Bernd Porr, Nick Bailey

This assignment covers IIR filters.

Your task is to solve a problem which requires real-time filtering of a physical quantity.

You have the following options for the data acquisition:

1. Arduino: They work under Windows, Mac and Linux. They have a sampling rate of up to 1kHz.
2. USB-DUX sigma: Linux-only but have isolated inputs for biomedical measurements

Every team (2 members) needs to have a different application for their measurement. Think of something simple such as measuring the speed of a fan with an LDR. Or if the flowerpot is watered. Add your topic to the wiki provided on moodle as soon as possible. It needs to solve a practical real life problem and requires low, high, bandpass or bandstop filtering. Excessive use of identical topics will result in low marks.

Again, you work in teams of 2 students and one report is submitted per team.

1. Present a realtime measurement problem to be solved which requires filtering. For example noise needs to be removed or a signal needs to be detected. Marks are given for initiative, inventiveness and originality. Document the experiment with (all compulsory):
   1. photos of the setup
   2. dataflow diagrams
   3. YouTube clip(s)

   Determine the filter response(s) which are required and justify them. Generate the SOS coefficients for the filter(s) Python's high level functions such as "butter". **[25%]**.
2. Do real-time filtering with the IIR class: https://github.com/berndporr/py-iir-filter. **[25%]**.
3. Add code which checks the sampling rate of your acquisition while the main application is running and show that it can run at the specified sampling rate. Add the sampling rate info to the plot window. **[25%]**
4. Compare your filtered results with the original recordings, show both signals in a real-time demo (YouTube clip) and discuss if you have been successful. Do a critical analysis. **[25%]**

High level design commands are allowed such as "butter". Any use of "lfilter", "conv" or other high level python filter operation will result again in zero marks. Proof of real-time processing in form of a video needs to be given and the video needs to show clearly what it's about. Please add your link to the wiki and also add a readme to your zip containing all the files.

- If you use an Arduino then you must use pyFirmata2 as it provides event driven real-time callbacks.

- If you need an electrically isolated ADC for biomedical recordings then a limited number of Linux DAQ boards are available (only for Linux users).

The code needs to be again submitted as a zip file and it will be tested if it runs. Crashing code will result in low marks. As before we expect sharp figures in vector format in the report. The complete code needs to be in the appendix and also uploaded to moodle. Since this task is more difficult and you need to work in a coordinated, distributed team, there is more time to finish it.
Deadline for the report is 4th Dec 2023.