**Aplicatie Java pentru preprocesarea semnalelor achiziționate**

Introducere:

Abordare teoretica

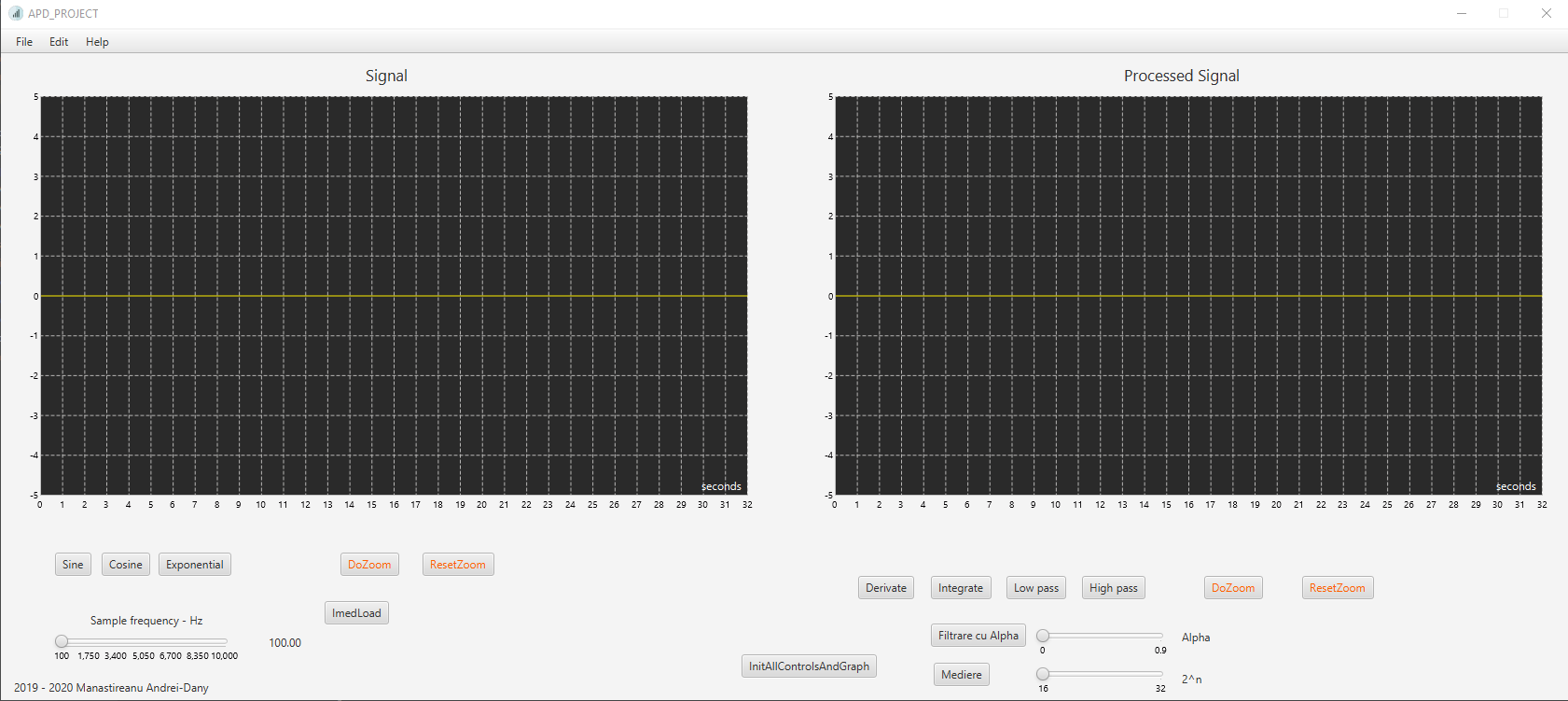
– implementareea algoritmilor de procesare a unui semnal din curs

Mediu de lucru:

- Intellij IDEA 2019 **JAVA FX**

Rezultate asteptate:

- Aplicare ferestre de timp, implementare filtre de netezire, derivare, integrare.



Descrierea implementarii aplicatiei:

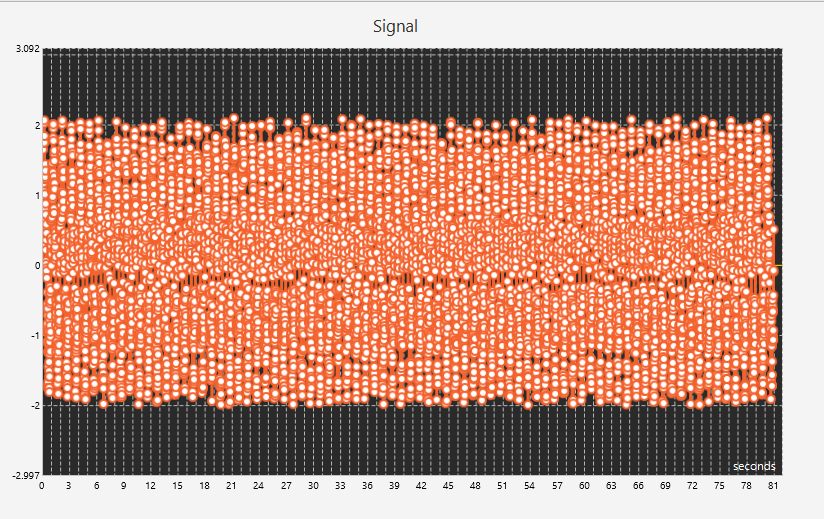
User interface

* Java FX mi-a permis o programare asemanatoare cu programarea in Android Studio, folosind pentru generarea fisierului fxml aplicatia SceneBuilder
  + Fisierul MainPanel.fxml imi retine controalele cu fiecare configuratie(pozitie, culoare, callback)
  + Informatiile sunt afisate pe un LineChart

Backend

* Am 3 fisiere java:
  + Main -> de unde fac load la fisierul MainController
  + MainController -> definita fiecare functie pentru fiecare control
  + SingnalGraph -> clasa ce are in componenta procesarea efectiva a semnalului

Implementarea functilor:



Load Signal:

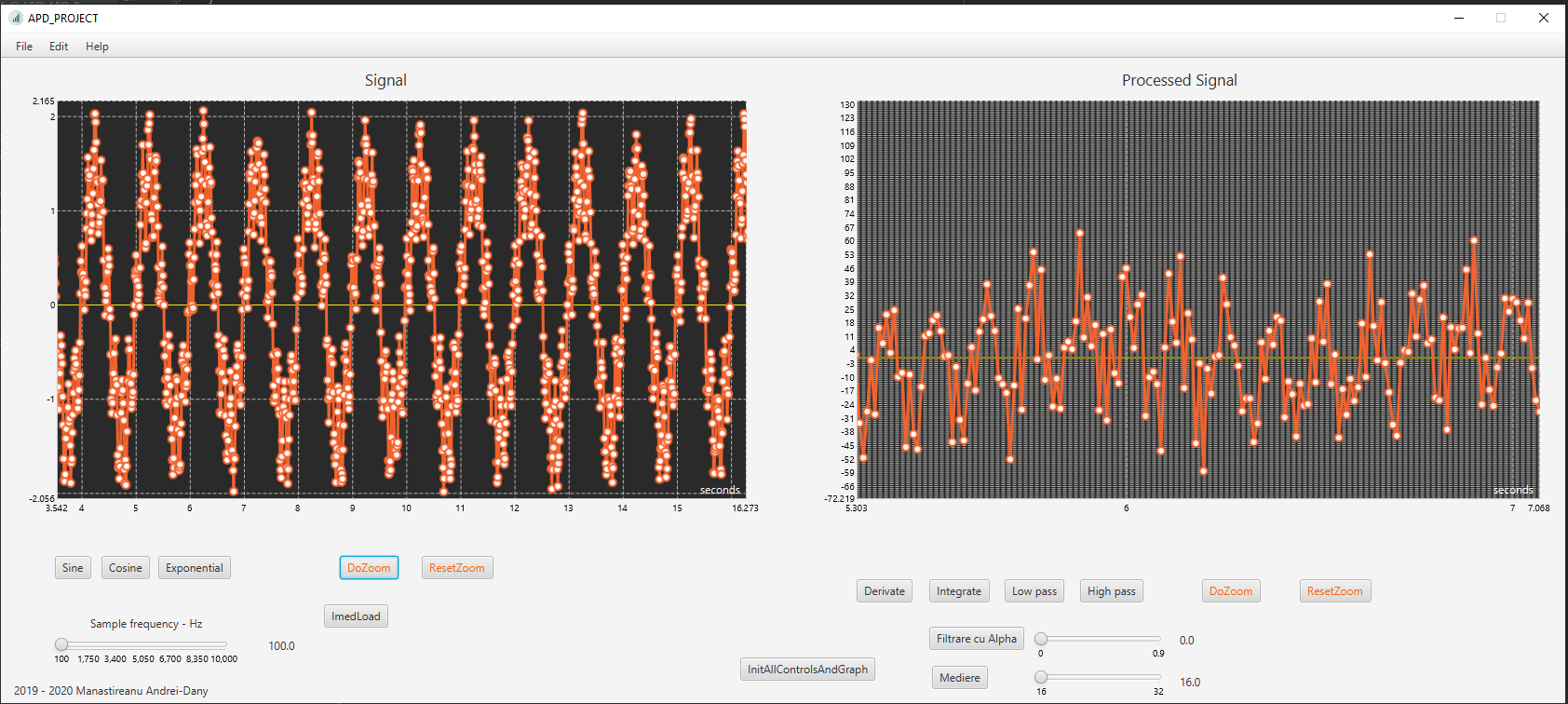
- fisierul este citit valoare cu valoare si in acest timp adaugate aceste valori intr-o lista

Write signal to file:

- se ia fiecare valoare de pe graphic si se scrie intr-un fisier

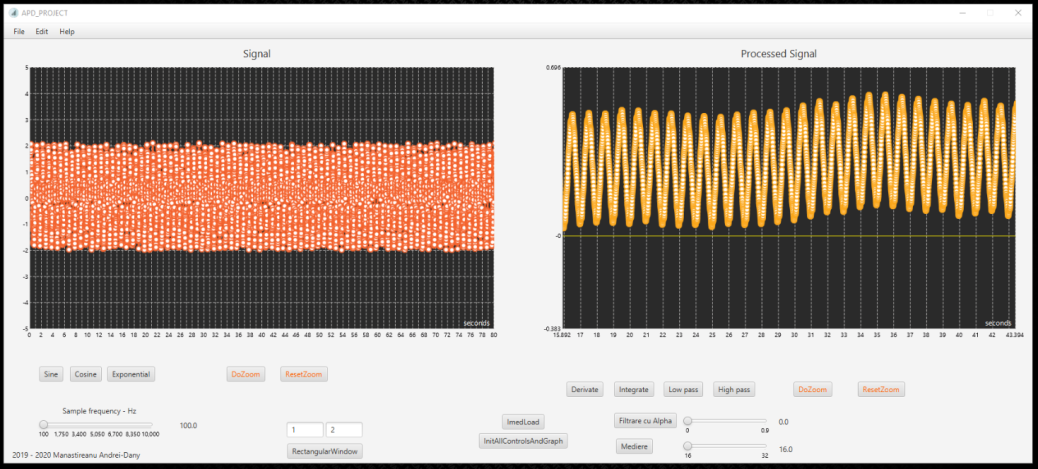
public void derivateValuesFromGraph(SignalGraph derivateSignal, Double step, Double frequency)

- calculez derivata folosind o valoare “din viitor” din care scad valoare curenta si impart totul la un ∆t.



public void integrateValuesFromGraph(SignalGraph integrateSignal, Double steps, Double frequency)

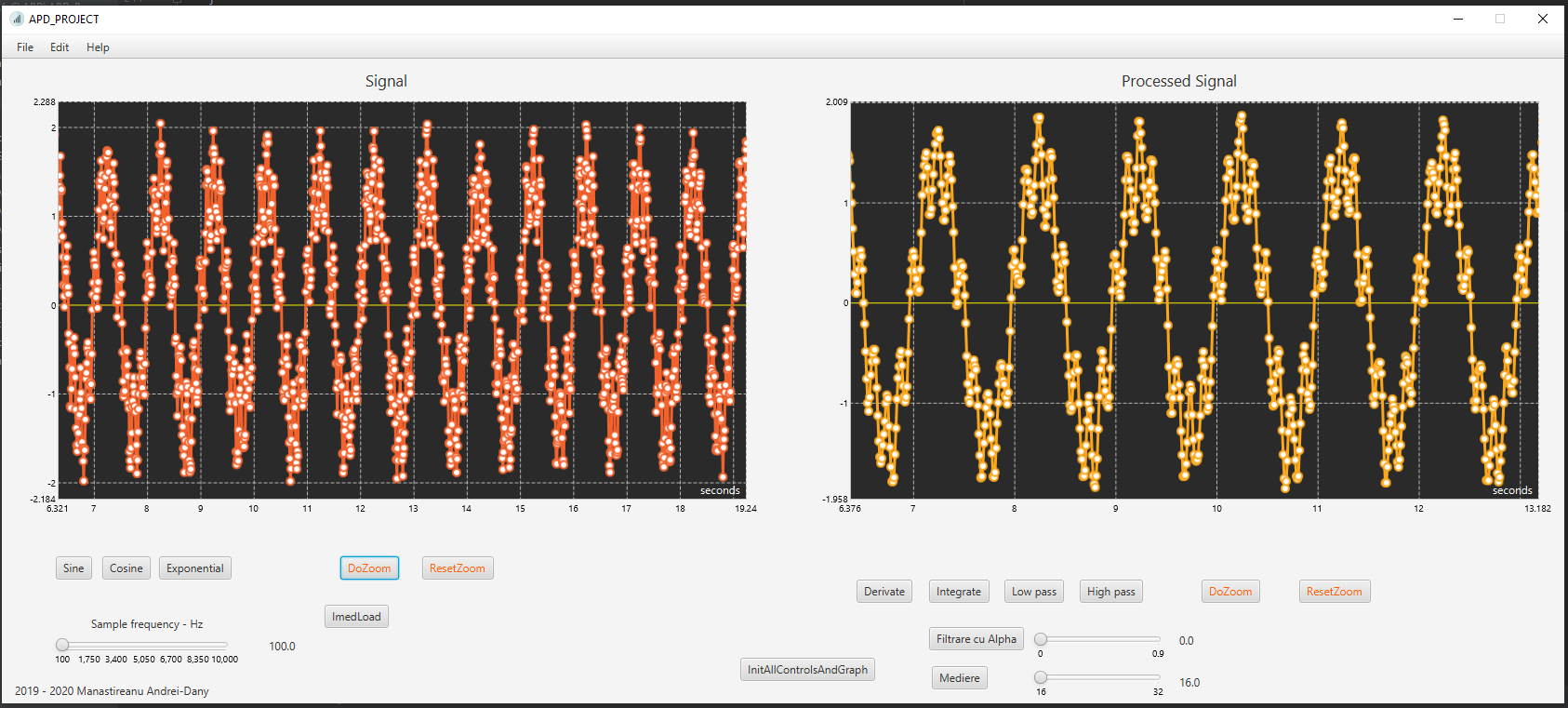
- iau primele 2 esantioane pe care le inmultesc cu un pas de integrare si retin rezultatul intr-o variabila pe care o voi folosi la urmatorul calcul.

- integrare prin metoda trapezului

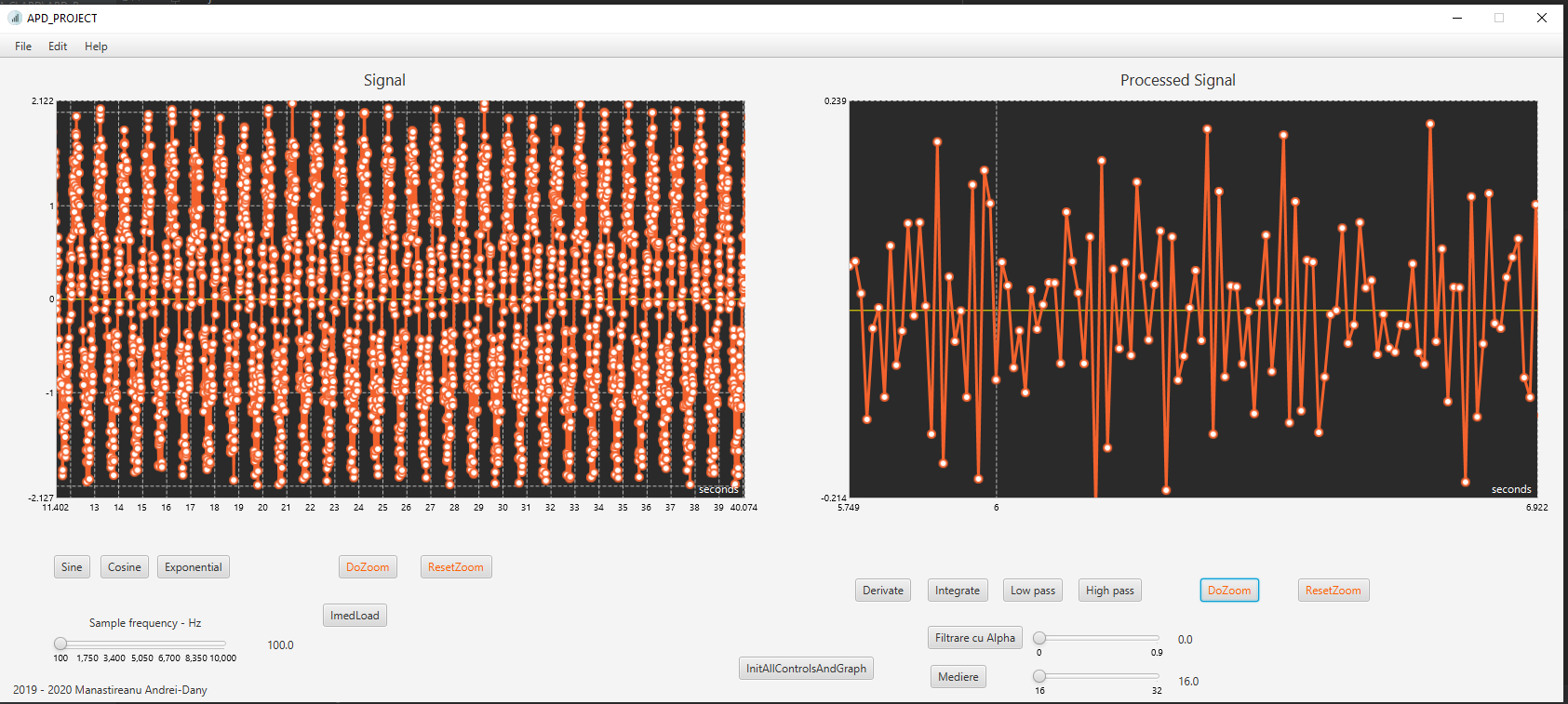
public void applyFilter(SignalGraph filteredSignal, Double steps, double[] filter, int range, double frequency)

- functie pentru aplicarea Low pass filter && High pass filter

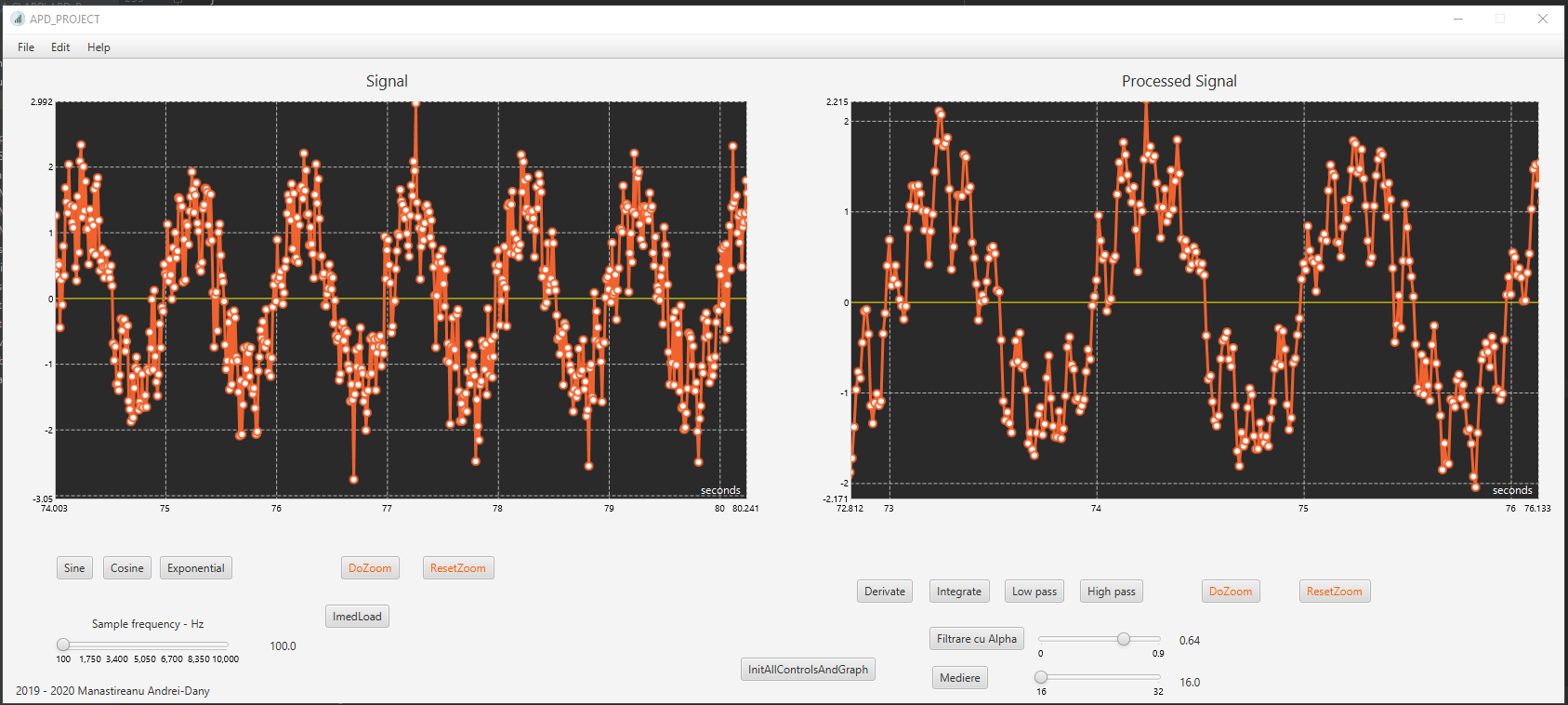
Low Pass



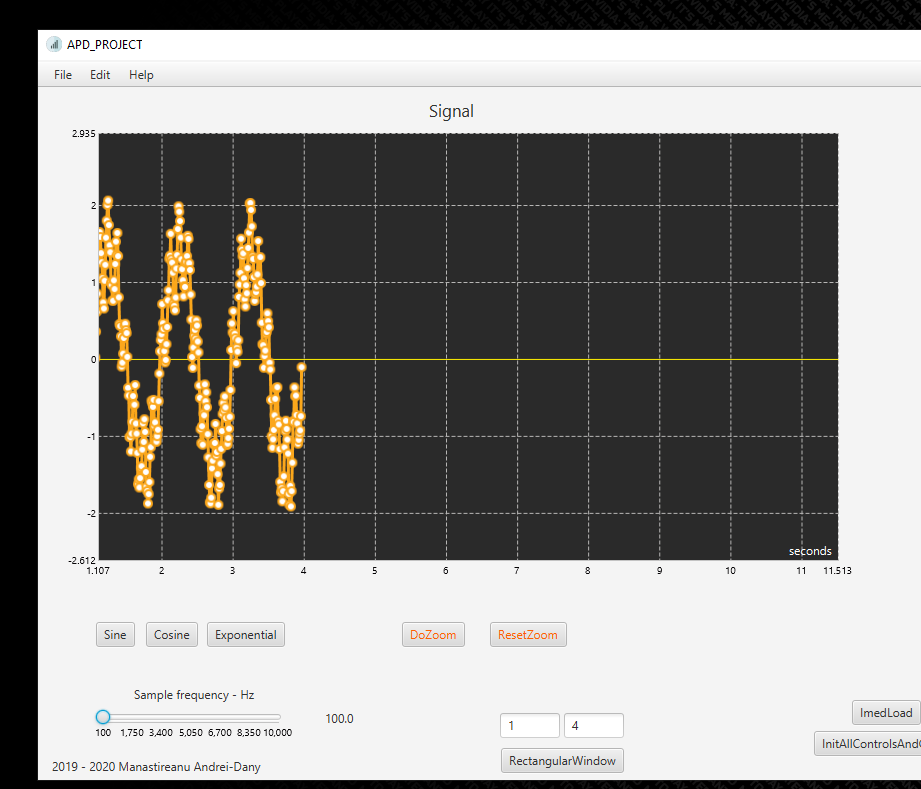
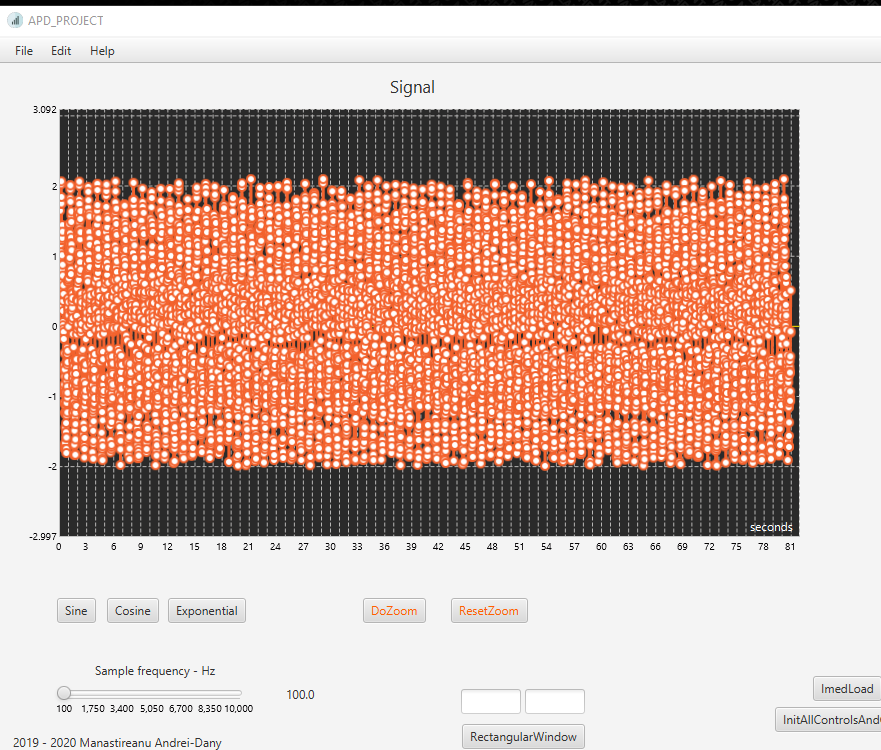
High Pass



White Noise Add (cu Matlab) si filtrare cu alpha



Fereastra de timp dreptunghiulara



Fiecare functie de procesare de semnal a fost implementata de la 0 folosind metodele numerice.

Total linii de cod: ~800+