ESLab HW7

游祖鈞 B06209040, 宋家齊 B07901187, 陳🄴平 B06208042

November 30, 2021

This is a group homework. Reports with several issues and solutions presented. The github link is at https://github.com/NOOMA-42/NTUEE-ESLab/tree/hw7

1 general approach

We use DSP program to process the data of 3D accelerators at x-direction.

1.1 Implement a low pass filter, like FIR

We use low pass filter to process the data. First, we need to copy the file that enable the functions of the sensors, BSP folder in the sensor program. Then, initialize the instance of fir "arm_fir_instance_f32" , two buffer storing the sensor value (size=320) and the filtered value, two pointers to each buffer. Finally, measure the accelerator data, put it into the buffer and use the function "arm_fir_f32" to filter the signals. We print out all the signals and put it to the filter written in python.

Cutoff frequency = 6 kHz. h = fir1(28, 6/24), where 28 is the order of the filter, 24 is the Nyquist frequency, which is sampling rate/2.

```
while(wait<1) {
    wait++;
    for (i=0;i<320;i++){
        printf("\nNew loop, LED1 should blink during sensor read\n");

        ThisThread::sleep_for(2);

        BSP_ACCELERO_AccGetXYZ(pDataXYZ);
        printf("\nACCELERO_X = %d\n", pDataXYZ[0]);
        printf("ACCELERO_Y = %d\n", pDataXYZ[1]);
        printf("ACCELERO_Z = %d\n", pDataXYZ[2]);

        ThisThread::sleep_for(2);
        b_x[i] = pDataXYZ[0]; b_y[i] = pDataXYZ[1]; b_z[i] = pDataXYZ[2];
        printf("%i\n",i);
    }

    for(i=0; i < numBlocks; i++)
    {
        arm_fir_f32(&Sx, x_sensor + (i * blockSize), x_out + (i * blockSize), blockSize);
        printf( "x_out:%f \n", *(x_out + (i * blockSize)));
    }
```

Figure 1: DSP FIR

1.2 Testing DSP result with Python

We test the data with the filter written in python. $numtaps = 29$, $cutoff\_hz = 6k$, $nyq\_rate = 24k$. signal=measured value. The result is 0.0001147, which is very small,

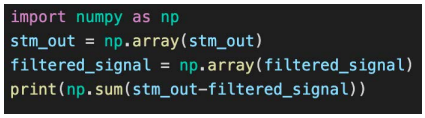

Figure 2: Python Output

showing that the STM filtered value is correct.

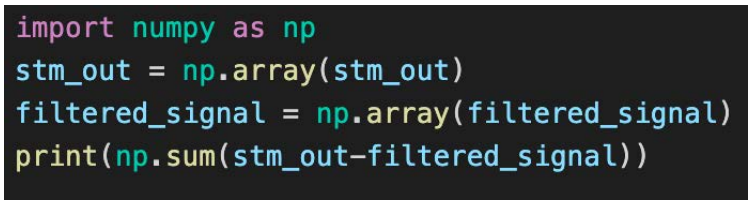

Figure 3: Python Output

2 Discussion

1. Only one direction is available. If I use two "arm_fir_32" function to filter the signals of two directions. The program can be compiled but has a problem in running stage. I ask TA about this question, but the problem still remain unsolved. The requirement of this homework is only one direction, so I give up digging into it. TA thought us how to debug problem like this, in case we meet the same problem in the term project.

2. We can design the filter' s order and coefficient depending on which cutoff frequency or sampling rate we want.