

Thema, Ziele: Digitale Filterung, Integer vs. Floating Point

Aufgabe: Filterung von EKG-Daten, Darstellung mit gnuplot

a) siehe Eclipse-Projekt in ./Loesung/FPFilter

b)

Funktion	Filterdatei
Hochpass	filter3.dat
Tiefpass	filter1.dat
50 Hz Notch	filter2.dat

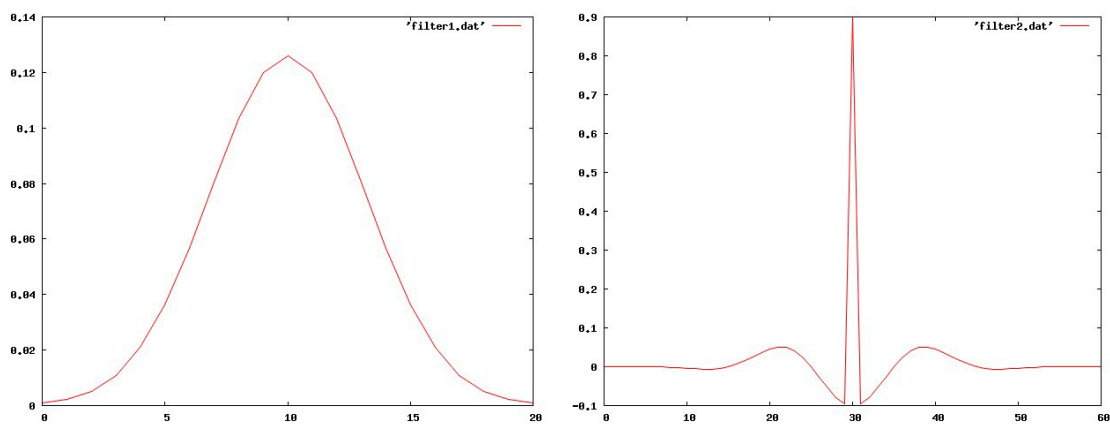


Abbildung 1: Tiefpass filter1.dat (links) und 50 Hz Notch filter2.dat (rechts)

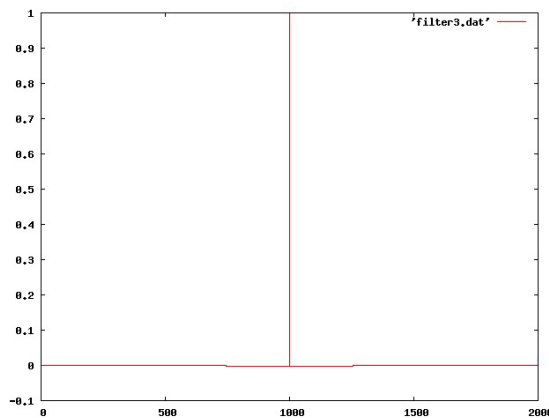


Abbildung 2: Hochpass filter3.dat

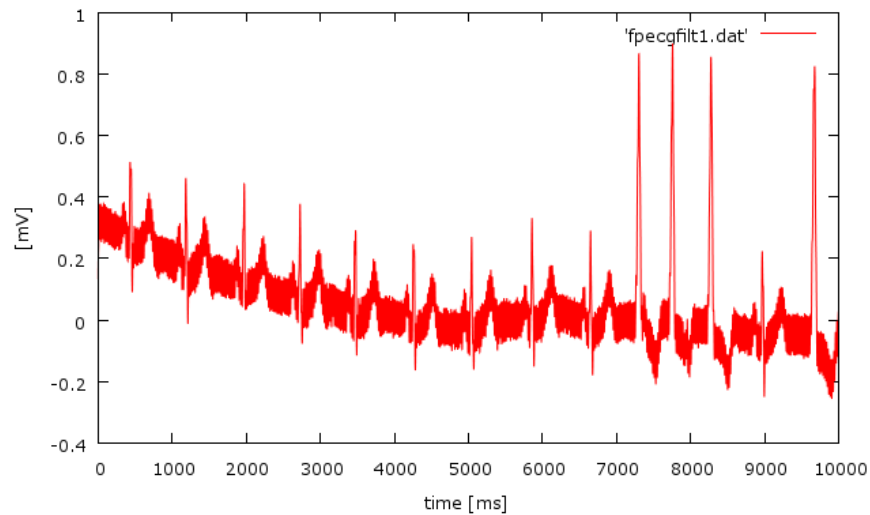


Abbildung 3: EKG-Daten nach Filterung mit Filter 1 (Tiefpass)

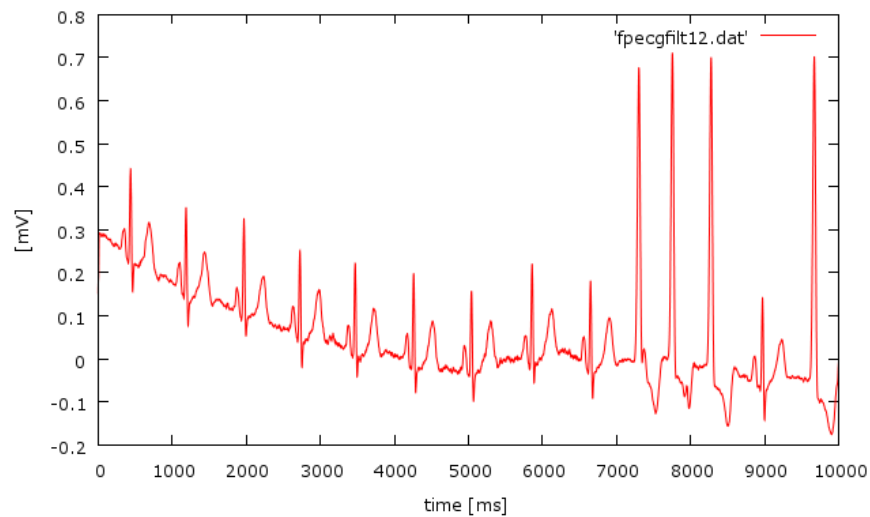


Abbildung 4: EKG-Daten nach zusätzlicher Filterung mit Filter 2 (50 Hz Notch)

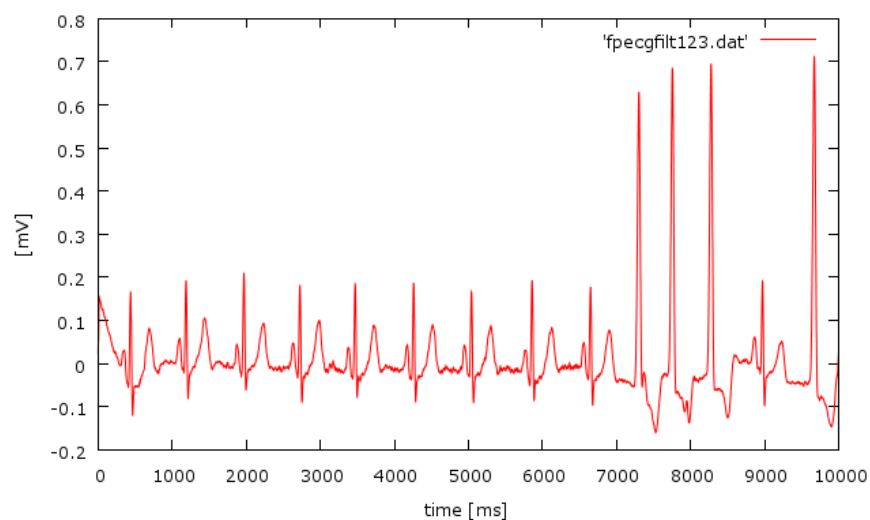


Abbildung 5: EKG-Daten nach zusätzlicher Filterung mit Filter 3 (Hochpass)

- c) Da die Daten als Floating Points vorliegen, werden folgende Operationen benötigt:
- FP Addition und FP Multiplikation.
 - Zur Berechnung des Index benötigt es nur Integeraddition und Integersubtraktion. Um I_f durch 2 zu teilen genügt eine Shiftoperation.
- Es ist eine typische Multiply-Accumulate (MAC) Funktion. Am besten dafür geeignet wäre ein DSP. Noch besser wäre selbstverständlich, wenn dieselbe Aufgabe auch mit Integern gelöst werden könnte.
- d) Sie führen Ihr Programm auf einem Controller mit folgenden Eigenschaften aus

Operation	Zyklen
FP Addition	8
FP Subtraktion	8
FP Multiplikation	12
FP Division	24
Integer-Addition (32 Bit)	1
Integer-Subtraktion	1
Integer-Multiplikation	2
Integer-Division	10

Bei einer Floating Point-Implementation würden für die Berechnung aller drei Filter zusammen 416.6 Mio Zyklen benötigt (siehe folgende Tabelle).

Filter	Koeffizienten (I_f)	Daten (n)	Additionen	Multiplikationen	Zyklen
filter1	21	10'000	210'000	210'000	4'200'000
filter2	61	10'000	610'000	610'000	12'200'000
filter3	2'001	10'000	20'010'000	20'010'000	400'200'000
Total			20'830'000	20'830'000	416'600'000

Wenn beispielsweise eine Zykluszeit von 10 ns angenommen wird (100 MHz Takt), dann würde die Berechnung 4.166 Sekunden dauern.

Eine Integerimplementation würde nur 62.49 Mio Zyklen benötigen (siehe folgende Tabelle). Die Berechnungszeit würde sich auf 0.6249 Sekunden reduzieren.

Filter	Koeffizienten (I_f)	Daten (n)	Additionen	Multiplikationen	Zyklen
filter1	21	10'000	210'000	210'000	630'000
filter2	61	10'000	610'000	610'000	1'830'000
filter3	2'001	10'000	20'010'000	20'010'000	60'030'000
Total			20'830'000	20'830'000	62'490'000

Nebenbei: Für eine Echtzeitanwendung müssten nicht alle 10'000 Werte aufs Mal berechnet werden. Da die Abtastung 1 ms beträgt, stünde für die Filterung von einem Wert im schlimmsten Fall 1 ms zur Verfügung. Allerdings wäre dann der Controller zu 100% mit der Filterung ausgelastet. In Echtzeit müssten je $(21 + 61 + 2001) = 2083$ Additionen und Multiplikationen innerhalb einer Millisekunde vorgenommen werden. Mit Floating Point würden dafür 41'660 Zyklen benötigt, der Controller müsste somit mit mindestens 41.66 MHz getaktet werden.

Die Integervariante würde 6'249 Zyklen dauern, ein Takt von 6.249 MHz würde genügen.

- e) Die Rohdaten sind Messwerte in Millivolt mit 6 Dezimalstellen¹. Sie liegen im Bereich ± 1 mV. Bei einer praktischen Anwendung müsste überprüft werden, ob die Messgenauigkeit wirklich 6 Stellen beträgt. Es macht nur Sinn, die genauen Stellen weiter zu bearbeiten. Wir nehmen hier an, dass 6 Stellen genau sind, die gefilterten Daten sollen ebenfalls auf 6 Stellen genau sein. Wenn sowohl Messdaten als auch Filter mit 10'000 multipliziert werden, ergibt sich eine maximale Abweichung von $5 \cdot 10^{-5}$ mV, was keine sichtbare Abweichung ergibt.

Wenn mehrere Filter hintereinander angewendet werden, muss geachtet werden, dass kein Overflow entsteht. In Abbildung 6 ist die resultierende Kurve dargestellt, wenn ausschliesslich mit Integeroperationen gerechnet und geeignet skaliert wird.

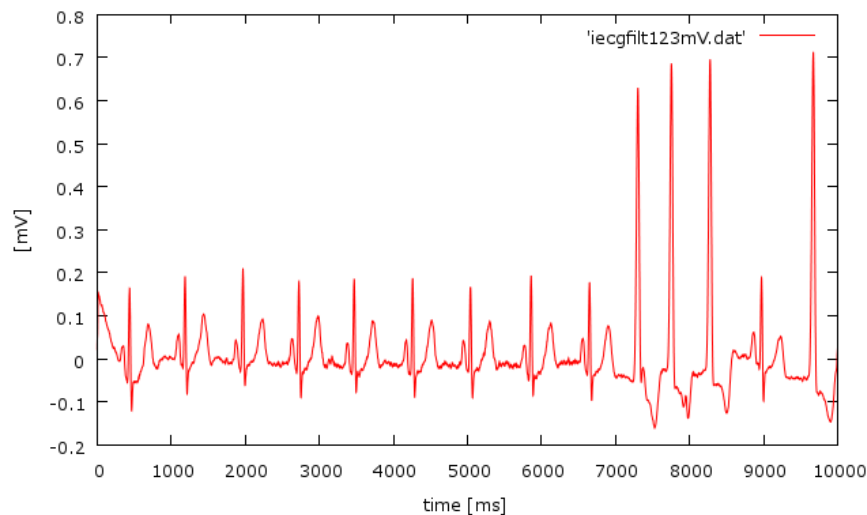


Abbildung 6: EKG-Daten nach Anwendung aller drei Filter nur mit Integeroperationen (kein Floating Point)

Die resultierende Abweichung ist mit blossen Auge nicht erkennbar. In Abbildung 7 ist die Abweichung zwischen der Integer- und der FP-Implementation hundertfach vergrössert dargestellt. Die maximale Abweichung beträgt lediglich 1.5 μ V.

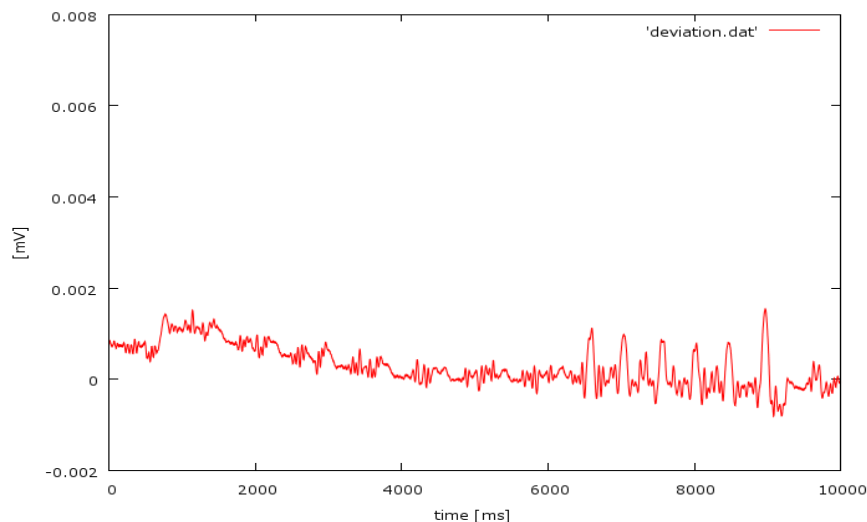


Abbildung 7: Abweichung der Integerimplementation zur Floating Point-Implementation (100 fach vergrössert)

¹ Tatsächlich sind die FP-Werte bereits gerundete Werte, denn die wirklichen Rohdaten werden die Ausgangswerte eines AD-Wandlers sein, und dieser liefert einen ganzzahligen Wert, keine Floating Points. Da es sich um einen binären Wert handelt, würde die Skalierung vernünftigerweise mit Zweierpotenzen statt mit Zehnerpotenzen erfolgen.