

2013•2014  
FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN  
*master in de industriële wetenschappen: elektronica-ICT*

## Masterproef

Ontwikkeling van een klinisch dashboard voor continue en real-time monitoring gedurende sportactiviteiten: "Big Mother is watching you"

Promotor :  
Prof. dr. ir. Luc CLAESEN

Promotor :  
dr. LARS GRIETEN

Jordie Simons

*Proefschrift ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT*

Gezamenlijke opleiding Universiteit Hasselt en KU Leuven

2013•2014

Faculteit Industriële

ingenieurswetenschappen

*master in de industriële wetenschappen: elektronica-ICT*

## Masterproef

Ontwikkeling van een klinisch dashboard voor continue en real-time monitoring gedurende sportactiviteiten: "Big Mother is watching you"

Promotor :  
Prof. dr. ir. Luc CLAESEN

Promotor :  
dr. LARS GRIETEN

Jordie Simons

*Proefschrift ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT*

## Voorwoord

Het onderwerp van deze thesis sprak mij enorm aan zowel de elektronica, de software, de signaalverwerking als het medische aspect. Omdat mijn softwarekennis eerder bescheiden was, was het zeker geen evidente keuze. Toch ben ik zeer blij dat ik de mogelijkheid heb gekregen om mijn eindwerk te verwezenlijken en om het feit dat ik veel heb bijgeleerd. Daarom wil ik het Ziekenhuis Oost-Limburg en de Universiteit van Hasselt bedanken.

Persoonlijk vond ik het fijn om in een ziekenhuisomgeving dit eindwerk uit te werken. Dit heeft zeker een positieve invloed gehad op het eindresultaat. Eerst en vooral wil ik mijn externe promotor dr. Lars Grieten bedanken. Het is door zijn enthousiasme en goede feedback dat dit eindwerk tot stand is gekomen. Tevens dank ik ook prof. Dr. Pieter Vandervoort voor zijn betrokkenheid en zijn deskundigheid over het medische aspect van het verhaal. Ook bedank ik mijn interne promotor prof. dr. ir. Luc Claesen voor zijn goede hulp.

Vervolgens zou ik Guy Jacobs van de cardiale revalidatie in Lanaken enorm willen bedanken voor het selecteren van geschikte hartpatiënten. Zonder hem was het niet mogelijk geweest om de vele metingen te doen en deel te nemen aan “Genk loopt”. Eveneens wil ik ook de enthousiaste patiënten en begeleiders bedanken: Jan Bialorucki, Schuurmans Eric, Goffin Erik, Jacobs Peter, Herinckx Marcus, Daniels Lut en Vandenborne Karolien.

Tot slot zou ik mijn familie en vrienden willen bedanken. Ze stonden klaar om mijn eindwerk na te lezen en hebben me ondersteund doorheen de uitwerking van de masterproef.



# Inhoudstabel

Voorwoord .....	I
Inhoudstabel.....	III
Figurenlijst.....	V
Lijst van tabellen.....	VII
Abstract (Nederlands) .....	IX
Abstract (Engels).....	XI
1. Inleiding.....	1
1.1. Ziekenhuis Oost-Limburg.....	1
2. Literatuurstudie.....	3
2.1. Fysiologische oorsprong van het electrocardiogram (ECG) .....	3
2.2. Registratie van een electrocardiogram .....	4
2.2.1. ECG sensorschakeling.....	4
2.2.2. Elektrodes.....	6
2.3. Mogelijke storingsbronnen op het ECG-signaal .....	8
2.3.1. Netspanningsruis.....	8
2.3.2. Basislijnverschuiving.....	9
2.3.3. Bewegingsartefacten.....	9
2.4. Digitale signaalverwerking .....	10
2.4.1. Introductie.....	10
2.4.2. Discrete Tijd Fourier Transformatie (DTFT).....	10
2.4.3. Filters .....	13
2.5. QRS piekdetectie .....	14
2.5.1. Pan-Tompkins algoritme .....	14
3. Materialen en methoden .....	15
3.1. Communicatie opstelling.....	15
3.2. ECG sensor node.....	16
3.3. MATLAB: Signaalanalyse/verwerking.....	17
3.4. Detectiealgoritmes .....	18
3.4.1. R piekdetectie.....	18
3.5. Programmeren in Java.....	20
3.5.1. Java klassen .....	20
3.6. Android applicatie .....	26

3.6.1.	ECG-sensor van imec .....	26
3.6.2.	BLE met Android .....	27
4.	Resultaten en Discussie .....	29
4.1.	MATLAB: Signaal analyse/verwerking .....	29
4.1.1.	Signaal analyse .....	29
4.1.2.	Signaal verwerking.....	30
4.2.	Detectiealgoritmes .....	36
4.2.1.	R-top detectie .....	36
4.3.	Java programma .....	39
4.3.1.	Werking monitorings applicatie .....	39
4.3.2.	Controle van de filterwerking.....	41
4.4.	Testen bij cardiale revalidatie .....	42
4.5.	Validatie van applicatie .....	43
4.6.	Test-loop en “Genk Loopt” .....	44
4.7.	Uitbreiding a.d.h.v. Android App.....	46
4.7.1.	Ontvangstprobleem.....	46
4.7.2.	Algemene werking van de EcgMonitor App .....	47
4.7.3.	EcgMontor Applicatie .....	48
5.	Conclusie .....	49
5.1.	Applicatie.....	49
5.2.	Werking .....	49
6.	Literatuurlijst .....	51

## Figurenlijst

Figuur 1: ECG-signal [13] .....	3
Figuur 2: Hart stroom verloop [14] .....	3
Figuur 3: Instrumentatieversterker .....	5
Figuur 4: Ag/AgCl elektroden .....	6
Figuur 5 : Driehoek van Einthoven   verschillende ECG-signalen afhankelijk van de plaats van de elektroden [12].....	7
Figuur 6: ECG-signaal met 50Hz storing (eigen fig.) .....	8
Figuur 7: ECG-signaal met basislijnverschuiving en 50Hz storing (eigen fig.) .....	9
Figuur 8: ECG-signaal met bewegingsartefact (eigen fig.) .....	9
Figuur 9: FIR-filter en IIR-filter .....	11
Figuur 10: Pan-Tompkins flowchart .....	14
Figuur 11: Algemeen verloop van de thesis .....	15
Figuur 12: Opstelling van het communicatie experiment.....	15
Figuur 13: Eenvoudige piekdetectie .....	18
Figuur 14: detectie van een stijgende flank .....	19
Figuur 15: Objectenstructuur/grafisch overzicht van het programma .....	20
Figuur 16: Gegevenspakketten afkomstig v.d. ECG noden .....	22
Figuur 17: ArrayBlockingQueue.....	23
Figuur 18: ECG-sensor van imec op BLE .....	26
Figuur 19: BLE GATT profiel [11] .....	27
Figuur 20: Frequentie inhoud van een gemeten ECG-signaal.....	29
Figuur 21: Bandoorlaatfilter (groen = hoogdoorlaatfilter, blauw = laagdoorlaatfilter) .....	30
Figuur 22: Hoogdoorlaatfilter 2de orde Butterworth, afsnijfrequentie op 1Hz.....	31
Figuur 23: Effect van de laagdoorlaatfilter.....	33
Figuur 24: ECG-signaal (= bovenste grafiek) en ACC-signaal (= onderste grafiek) .....	35
Figuur 25: Detectie op het ECG-signaal in rust.....	36
Figuur 26: Detectie algoritme op een persoon die aan het lopen is.....	37
Figuur 27: Pan-Tompkins detectie algoritme tijdens een minimale inspanning.....	37
Figuur 28: Detectie van QRS-complex tijdens actieve inspanning .....	38
Figuur 29: Startscherm .....	39
Figuur 30: Controlepaneel.....	40
Figuur 31: Impuls responsie .....	41
Figuur 32: Stap responsie.....	42
Figuur 33: Bland-Altman tussen gevalideerd toestel en applicatie.....	43
Figuur 34: Monitoring van het team van patiënten/begeleiders tijdens "Genk Loopt" .....	44
Figuur 35: Golfkar met patiënten erachter .....	44
Figuur 36: Monitorapplicatie tijdens Genk loopt in actie .....	45
Figuur 37: Pan-Tompkins van het slecht ECG-signaal tijdens "Genk Loopt" .....	45
Figuur 38: Problemen weergave ECG-signaal .....	46
Figuur 39: Functies EcgMonitor APP .....	47
Figuur 40: Algemene werking van BLE .....	47
Figuur 41: EcgMonitor App in actie.....	48





## Lijst van tabellen

Tabel 1: Pan-Tompkins stappen .....	18
Tabel 2: Seriële instellingen voor het basisstation.....	21
Tabel 3: Services van de imec-sensor.....	27
Tabel 4: Coëfficiënten van de hoogdoorlaatfilter .....	31
Tabel 5: Coëfficiënten laagdoorlaatfilter .....	32
Tabel 6: Correlatie ECG-sigitaal met accelerometer gegevens .....	34



## **Abstract (Nederlands)**

Jordie Simons

Faculteit Industriële Ingenieurswetenschappen, Universiteit Hasselt,  
Agoralaan Gebouw H, B-3590 Diepenbeek, België

Reële tijd fysiologische parametermonitoring buiten een ziekenhuisomgeving staat nog in zijn kinderschoenen, zeker wat afstandsmonitoring betreft. Deze problematiek wordt naar voor geschoven in het kader van het evenement “Genk Loopt” in samenwerking met imec en het ZOL.

Het doel is om een applicatie te ontwikkelen, die het mogelijk maakt om het elektrocardiogram (ECG) van meerdere patiënten met een risico op hartfalen te monitoren tijdens het beoefenen van sport. Als resultaat moet er een controlepaneel zijn waarop meerdere ECG-signalen worden weergegeven en waarvan de signalen vooraf zijn gefilterd zodat interpretatie mogelijk is tijdens een inspanning. Verder moet er een accurate hartslagindicatie aanwezig zijn voor elk ECG-signaal.

Om het ECG-signaal te detecteren maakt men gebruik van een draagbaar en draadloos ECG-toestel, ontwikkeld door imec. Die zal de ECG-gegevens via RF doorsturen naar een basisstation. De ontvangen gegevenspakketten worden verwerkt voor filtering en visualisatie. De filters zijn eerst onderzocht en getest in MATLAB. Een Butterworth banddoorlaatfilter gaf het beste resultaat. Daarnaast kwam het Pan-Tompkins algoritme het beste naar voor als hartslag detectiealgoritme. Het beoogde resultaat heeft er toe geleid dat er een controlepaneel is waarop zes verschillende ECG-signalen met de bijhorende hartslag te zien zijn. Het ontwikkelde platform heeft als proof-of-concept gediend tijdens het evenement “Genk Loopt” waar zes cardiologische patiënten in reële tijd gemonitord werden.

Interne promotor: prof. dr.ir. Luc Claesen - UHasselt

Externe promotor: dr. Lars Grieten - Mobile Health Unit



## Abstract (Engels)

Jordie Simons

Faculty of Engineering Technology, Hasselt University,  
Agoralaan Gebouw H, B-3590 Diepenbeek, Belgium

Real time physiological parameter (ECG,EEG,EMG) monitoring outside a hospital environment is still in its infancy, especially when it concerns remote monitoring. This problem will be showcased in an event called “Genk Loopt”.

The goal is to develop an application, which is able to monitor the electrocardiogram (ECG) of multiple high-risk patients during exercise activities. The result has to be a dashboard where multiple ECG-signals are shown in real time. Therefore the signals are first filtered so that they still can be interpreted during exercising. Also each signal gets its own heart rhythm indicator.

A portable and wireless ECG device developed by imec, is used to detect the ECG signal. This device then sends the ECG data to a base station so that a computer can process it. Java is the programming language which was chosen for the implementation because of its high performance and ability to run on multiple platforms. The filters are first developed and tested in MATLAB and then translated into Java code. The best results came from a Butterworth band-pass filter. The Pan-Tompkins algorithm was proven to be the best heart rate detection algorithm. This results in a dashboard application where six ECG signals are plotted in real time with their corresponding heart rate. The developed application also was used as proof-of-concept to monitor six heart patients during the event “Genk loopt”.

Alongside, the start was made to develop a monitor platform on an Android device with the same functionalities as described above.

Internal promoter: prof. dr. ir. Luc Claesen - UHasselt

External promoter: dr. Lars Grieten - Mobile Health Unit



# 1. Inleiding

Deze master thesis bevindt zich in het domein van telemonitoring. Telemonitoring is een term die gebruikt wordt wanneer het gaat over mobiele gezondheidszorg die toelaat patiënten op afstand te volgen. Wanneer men in reële tijd gegevens verzamelt, dan spreekt men vaak van telemetrie (bv. een electrocardiogram of ECG).

In de medische wereld kan en zal telemonitoring grote voordelen opleveren. Denk maar aan het screenen van patiënten op afstand, tijdig ingrijpen in acute situaties, verbeteren van de revalidatie procedures, monitoren tijdens sportinspanning, etc. Het zorgt voor een betere diagnostisering en controle. De betere diagnosestelling is te wijten aan het feit dat de metingen over langere periodes gebeuren waardoor uiting van subklinische symptomen mogelijk aan het licht komen. Tegenwoordig gebeuren metingen veelal in het ziekenhuis zelf, dit maakt het moeilijk om correcte diagnoses te stellen doordat dit slechts een momentopname is en er geen longitudinale gegevens ter beschikking zijn.

De finaliteit van deze masterproef is om een op maat gemaakt controlepaneel te ontwikkelen, dat instaat voor de veiligheidsmonitoring van meerdere hartpatiënten. De interesse gaat hierbij vooral uit naar het ECG-signaal. Sensoren ontwikkeld door imec zullen die detecteren, waarna ze draadloos worden doorgestuurd. Een basisstation gekoppeld aan een PC zal deze signalen ontvangen, waarna de ontwikkelde software deze conditioneert. Een detectiealgoritme maakt het mogelijk de hartritmes te berekenen en ze te vergelijken met een vooraf ingestelde grenswaarde. Op het controlepaneel wordt alles in reële tijd weergegeven zodat een cardioloog de signalen kan verifiëren.

De deadline ligt op 4 mei '14 om deze monitorapplicatie te showcasen op "Genk Loopt". Zes hartpatiënten zullen deelnemen terwijl ze gescreend worden. Om dit te verwezenlijken zijn er extensieve testen uitgevoerd op zowel de applicatie zelf als de patiënten.

## 1.1. Ziekenhuis Oost-Limburg

De masterproef is uitgewerkt in het Ziekenhuis Oost-Limburg (ZOL) dit door een mooie samenwerking met de Mobile Health Unit van de UHasselt en imec uit Eindhoven. Door te werken in een ziekenhuisomgeving is het eenvoudig om de monitorapplicatie direct te testen op patiënten. Op die manier kunnen er snel fouten worden opgespoord en kunnen er meteen de nodige aanpassingen gebeuren. Daarbij komt nog dat er veel mensen met een medische achtergrond zitten die weten welk eindresultaat er gewenst is.

Veel testen zijn gebeurd bij de cardiale revalidatie in Lanaken. De patiënten die naar het revalidatiecentrum komen, hebben allerlei hartproblemen. Zo zijn er patiënten met overbruggingen alsook mensen die een hartaanval of zelfs een hartstilstand hebben gehad. Om te voorkomen dat dit zich opnieuw zou voordoen, komen ze driemaal per week sportoefeningen doen. Dit doen ze om hun algemene conditie te verbeteren. Zes van deze patiënten zijn gescreend en getest om mee deel te nemen aan de loopwedstijd van "Genk Loopt".

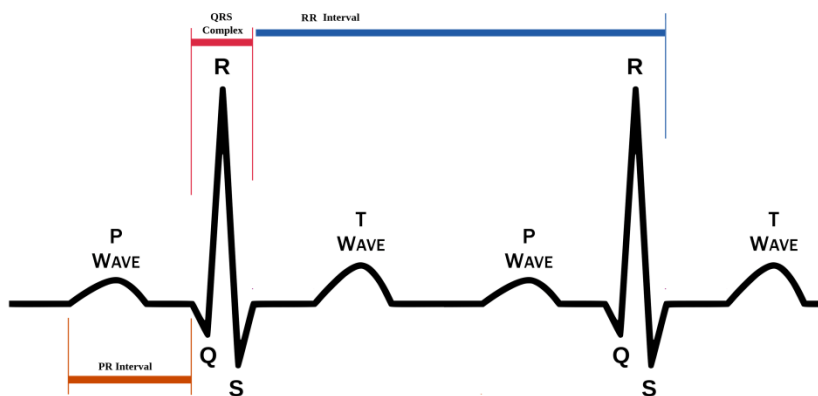




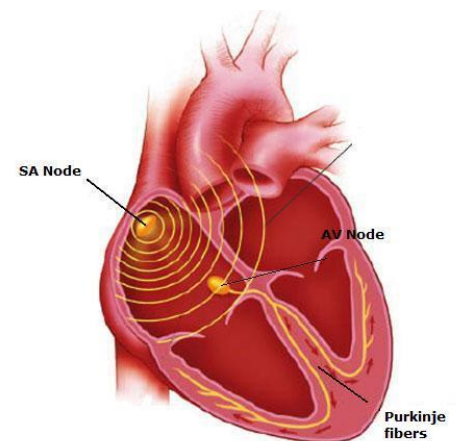
## 2. Literatuurstudie

### 2.1. Fysiologische oorsprong van het electrocardiogram (ECG)

De functie van het hart is om bloed rond te pompen doorheen ons lichaam. Zo kan het zuurstofrijk bloed onze vitale organen bereiken en het zuurstofarm bloed zal naar onze longen gaan. De reden waarom ons hart kan pompen is door een fenomeen dat depolarisatie noemt. Dit is in feite een elektrisch signaal zoals getoond in Figuur 1. Dit signaal wordt ook wel electrocardiogram (ECG) genoemd.



Figuur 1: ECG-signal [13]



Figuur 2: Hart stroom verloop [14]

Depolarisatie gebeurt wanneer een cel van polariteit verandert. Normaal gezien zijn de cellen in het hart negatief geladen. Maar op bepaalde ogenblikken maakt de sinusknoop (SA Node), wat in principe een natuurlijke pacemaker is, een elektrisch impuls aan. Deze puls is positief geladen en zal zich in de richting van de atrioventriculaire knoop (AV node) bewegen. Doordat de puls doorheen de cellen gaat zal die ervoor zorgen dat de cellen van polariteit veranderen waardoor de bovenste twee hartkamers zullen samentrekken. De hierboven beschreven actie wordt gerepresenteerd door de P-golf die te zien is in Figuur 1.

Eens het elektrisch signaal is toegekomen in de AV knoop zal die even pauzeren, zodat de twee onderste kamers van het hart zich volledig met bloed kunnen vullen (door de contractie van de voorkamers). Dit moment wordt voorgesteld door de horizontale lijn achter de P-golf. Daarna zal de puls losgelaten worden en zich verder gaan voortbewegen doorheen de bundel van Hiss naar de Purkinjevezels. Eenmaal het signaal hier is aangekomen zal deze terug omhoog bewegen via de buitenkant van het hart. Het gevolg hiervan is dat de onderste twee kamers zullen samentrekken, waardoor het bloed uit de onderste kamers wordt gestuwd. Dit hele proces is voorgesteld als het QRS-complex Figuur 1.

Wanneer het bloed is weggepompt, zal het hart terug repolariseren/relaxeren, wat de T-golf representeert.

## 2.2. Registratie van een elektrocardiogram

Zoals vermeldt in het hoofdstuk over de fysiologische oorsprong van het ECG, is het ECG-signaal elektrisch. Het heeft een maximum spanningswaarde van ongeveer 1 millivolt, wat zeer klein is. Daarom is het noodzakelijk om dit signaal te versterken. Veelal gebeurt dit met een instrumentatieversterker. Het is gewenst het ECG-signaal te versterken maar echter de ruis niet. Om een elektrisch signaal aan de instrumentatieversterker te krijgen zijn er elektroden nodig. In het geval van deze masterproef zal het toestel werken met behulp van twee elektroden. In dit hoofdstuk zal het gaan over de versterker, de elektrodes en de plaatsing van de elektrodes.

Opmerking: de formules staan onderaan de tekst waarop ze betrekking hebben.

### 2.2.1. ECG sensorschakeling

Een veel gebruikte schakeling voor het detecteren van een ECG-signaal is de zogenaamde “Instrumental Amplifier” (IA) of vertaald de instrumentatieversterker. Deze versterker biedt een aantal voordelen zoals: een hoge ingangsimpedantie, lage uitgangsimpedantie en een hoge Common Mode Rejection Ratio (CMMR).

#### 2.2.1.1. Common Mode Rejection Ratio (CMRR)

Een operationele versterker of “opamp” is een verschilversterker. Wanneer hetzelfde signaal op beide ingangen terecht komt zou men veronderstellen dat dit geen effect heeft op de uitgang. Dit blijkt niet zo, dit omdat de transistoren en de weerstanden van deze versterker niet exact hetzelfde zijn. Dit maakt dat er een spanningsverschil ontstaat aan de ingang en dit heeft ook zijn effect op de uitgang. Dit effect is een storing. Een parameter om aan te geven in welke maten deze storing wordt onderdrukt is CMMR.

Formule 1.3 specificeert de common mode rejection (CMR). Deze formule krijgt men door de verschilversterking te delen door de common mode versterking. Het logaritme van de CMR geeft de common mode rejection ratio (CMMR). [1]

#### Formules:

$$A = \frac{V_{out}}{V_1 - V_2} = \text{verschilversterking} \quad \{1.1\}$$

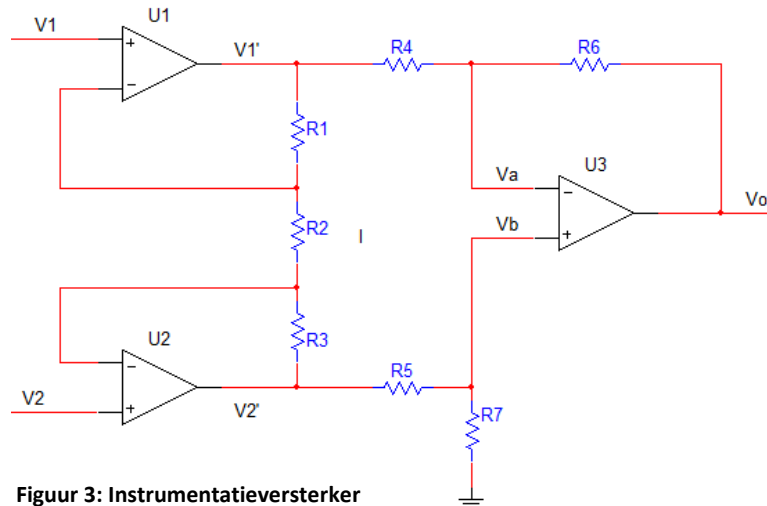
$$A_c = \frac{V_{out}}{V_c} = \text{common mode versterking} \quad \{1.2\}$$

$$CMR = \frac{A}{A_c} \quad \{1.3\}$$

$$CMMR = 20 \cdot \log_{10}(CMR) \quad \{1.4\}$$

### 2.2.1.2. Werking van de schakeling

De schakeling bestaat uit twee delen een “ingangsbuffertrap” en een “verschilversterking trap”. Ten eerste wordt de “ingangsbuffertrap” besproken. Wanneer een spanning wordt aangelegd aan ingang  $V_{1+}$ , dan zal dezelfde spanning op  $V_{1-}$  tevoorschijn komen. Hetzelfde gebeurt aan de opamp  $U_2$ . Hierdoor komt de spanning van  $V_{1+}$  en  $V_{2+}$  over weerstand  $R_2$  te staan. [1,2]



Figuur 3: Instrumentatieversterker

Indien er een spanningsverschil is aan  $V_{1+}$  en  $V_{2+}$  dan zal dit spanningsverschil ook over  $R_2$  staan. Wat als gevolg heeft dat er een stroom vloeit doorheen deze weerstand. Diezelfde stroom zal ook door weerstanden  $R_1$  en  $R_3$  lopen. Dit omdat er geen stroom kan wegvloeien via de opamps (vanwege hun zeer hoge ingangsimpedantie).

Wanneer er eenzelfde spanning ( $V_c$ ) aan de ingangen ( $V_{1+}$  en  $V_{2+}$ ) ligt dan zal er geen spanningsverschil over  $R_2$  staan. Dit betekent dat er ook geen stroom zal vloeien doorheen  $R_2$ . Indien er geen stroom vloeit zal  $V_c$  op  $V_{1'}$  en  $V_{2'}$  verschijnen. Hierdoor zal de versterking ( $A_c$ ) van de common mode signalen gelijk zijn aan één en met gevolg dat de CMR zal toenemen, zie formule 1.3. [1,2]

Ten tweede, het uitgangsgedeelte is een normale differentiaalversterker. Wat betekent dat de overgebleven common mode signalen van elkaar zullen afgetrokken worden. De formule 2.1. geeft de CMR van het hele circuit weer.  $A_1$  en  $A_{C1}$  zijn de versterkingen van de “ingangsbuffertrap” en  $A_2$  en  $A_{C2}$  zijn dat van de “verschilversterking trap”. Het is duidelijk dat hier de CMR veel beter zal zijn t.o.v. een gewone verschilversterker. [1,2]

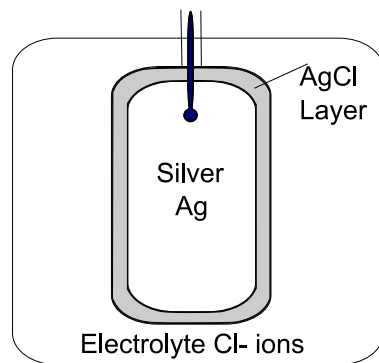
De hoge CMMR, met hoge ingangsimpedantie en lage uitgangsimpedantie, maakt dat dit een veel gebruikte schakeling is om metingen te doen, ook voor ECG-signalen. De schakeling in Figuur 3 heeft twee elektrodes nodig om een ECG-signaal te meten. Er bestaan ook toestellen met drie elektrodes. Hierbij wordt één elektrode gebruikt om de CMR nog meer te doen stijgen. [1,2]

#### Formule:

$$CMR = \frac{A_1 \cdot A_2}{A_{C1} \cdot A_{C2}} \quad \{2.1\}$$

### 2.2.2. Elektrodes

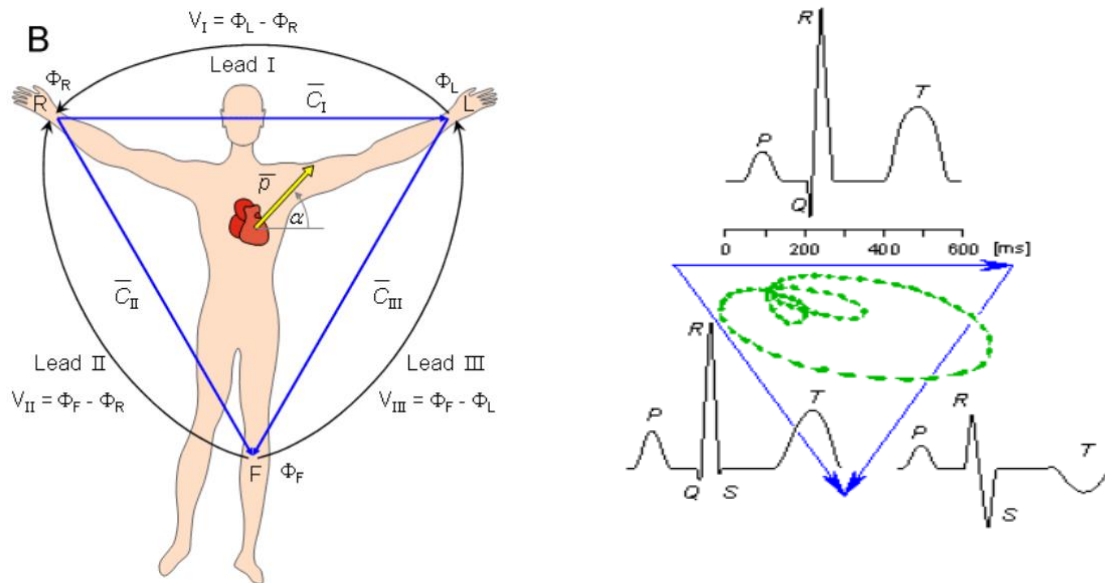
Doordat ladingen in ons lichaam worden voortgebracht door ionen dient er een ladingsoverdracht plaats te vinden tussen de ECG-monitor en het lichaam. Deze interface wordt voorzien door elektroden. De cellen in het menselijk lichaam kunnen gezien worden als een soort batterijen. Deze cellen hebben verschillende ionenconcentraties binnen en buiten hun membraam. Dit verschil in ionenconcentratie zorgt voor een potentiaalverschil. Dit potentiaalverschil wordt ook wel biopotential genoemd. [2]



Figuur 4: Ag/AgCl elektroden

Wanneer een potentiaal verstoord wordt spreken we van een actiepotentiaal. Het elektrisch signaal in ons hart zorgt ervoor dat dit gebeurt. De elektrodes kunnen dit actiepotentiaal waarnemen zodat het doorgegeven kan worden aan de instrumentatieversterker. In Figuur 4 is de samenstelling van een veel gebruikte elektrode weergegeven. Deze samenstelling is zo gekozen omdat ze weinig polariseerbaar is, wat maakt dat de bewegingsstoring minder wordt doorgegeven.

### 2.2.2.1. Driehoek van Einthoven



Figuur 5 : Driehoek van Einthoven | verschillende ECG-signalen afhankelijk van de plaats van de elektroden [12]

#### Afleidingen:

- Afleiding 1: tussen rechterarm en linkerarm
- Afleiding 2: tussen rechterarm en linkerbeen
- Afleiding 3: tussen linkerarm en linkerbeen

De plaats waar de elektroden worden bevestigd is van belang. Dit heeft namelijk een invloed op het gemeten signaal. Hierrond is een standaard uitgewerkt, namelijk de standaard van de 12-afleidingen. Deze is gebaseerd op de driehoek van Einthoven waarvan de afleidingen in Figuur 5 zijn te zien. [3]

Het ECG-signaal van het hart kan gezien worden als een elektrische vector die van hoek en amplitude verandert doorheen de tijd. De plaats van de elektroden hebben een grote invloed op welk onderdeel van het ECG-signaal het beste zichtbaar wordt. In het rechter deel van de figuur zijn de invloeden voor de drie afleidingen weergegeven. Dit is een theoretische benadering, in de praktijk is het zo dat het hart veelal een bepaalde kanteling heeft. Met als gevolg dat het ooit zoeken is om een mooi ECG-signaal te vinden.

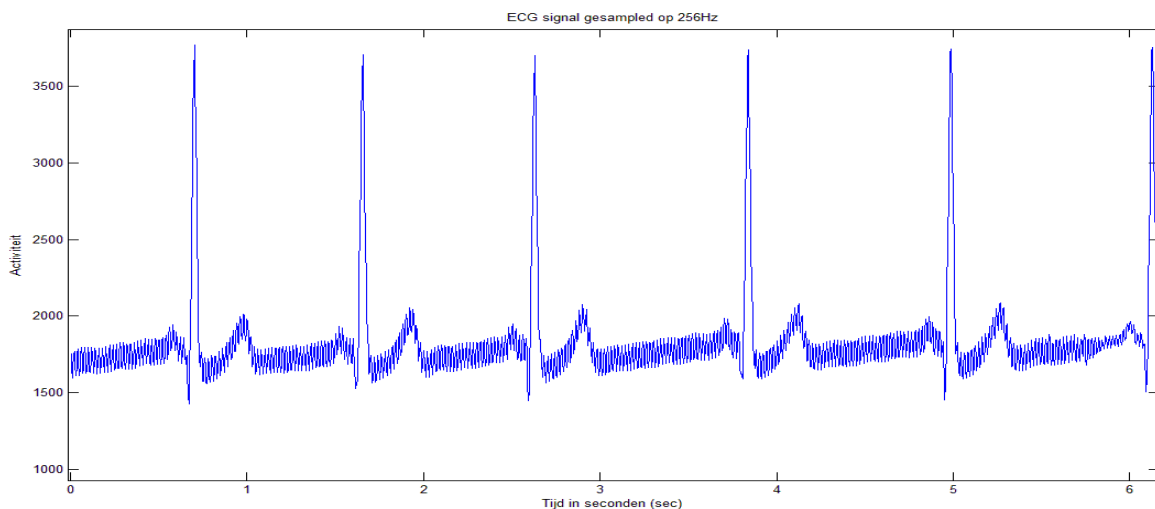
## 2.3. Mogelijke storingsbronnen op het ECG-signaal

Een belangrijk onderdeel bij het meten van ECG-signalen is het verwijderen van ongewenste signalen. Ongewenste signalen kunnen verwijderd worden door digitale filtertechnieken. Deze filtering gebeurt door wiskundige bewerkingen/algorithmes op het bemonsterd ECG-signaal toe te passen. Het uiteindelijke resultaat zou een beter signaal moeten opleveren.

Een ECG-signaal kan verschillende soorten storingen bevatten: netspanningsruis, bewegingsartefacten, nullijnverschuiving, enzovoort. In de volgende hoofdstukken gaan deze storingen aanbod komen.

### 2.3.1. Netspanningsruis

Een eerste belangrijke storing is de zogenaamde: “netspanningsruis”. Deze wordt veroorzaakt door elektromagnetische storingen (EMC) afkomstig van elektrische leidingen of lampen. Deze storingen zijn eenvoudig te identificeren aan het feit dat er een 50 Hz signaal is gesupponeerd op het gewone ECG-signaal. (Figuur 6)

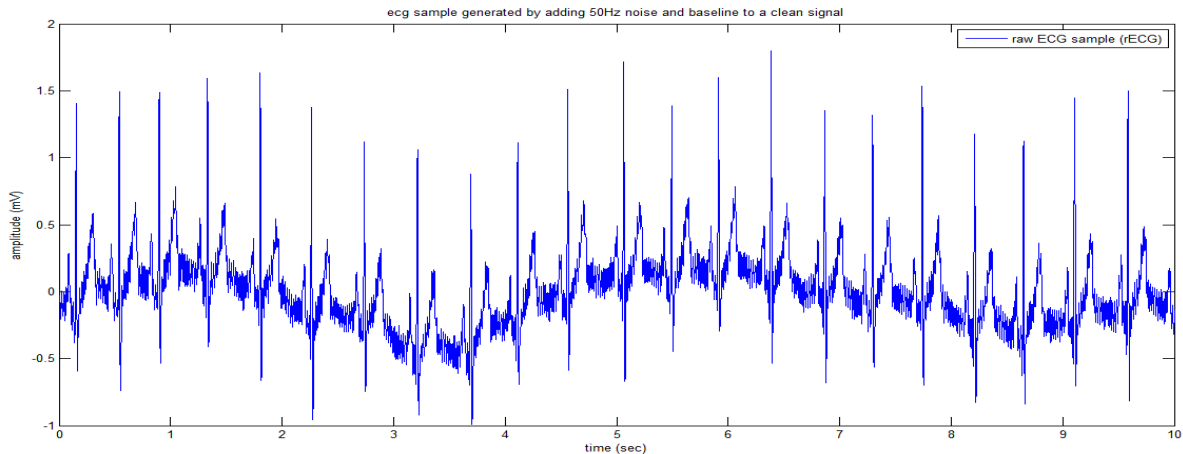


Figuur 6: ECG-signaal met 50Hz storing (eigen fig.)

Deze EMC zal het lichaam binnendringen en er een stroom veroorzaken. Omdat het menselijk lichaam in feite ook een weerstand heeft, veroorzaakt deze stroom een potentiaalverschil bij de elektroden. Dit verschil wordt versterkt door de IA en komt zo terecht bij het ECG-signaal.

### 2.3.2. Basislijnverschuiving

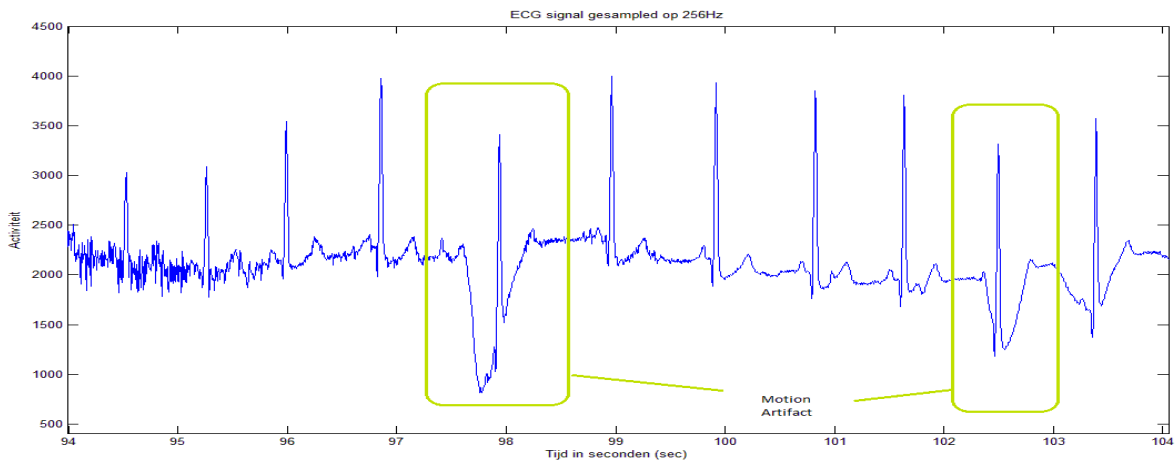
Basislijnverschuiving wordt herkend doordat er een zeer laag frequent signaal bij het ECG-signaal zit, de voornaamste oorzaken van dit fenomeen zijn ademhaling en beweging. Omdat de borstkast omhoog en omlaag beweegt zullen de elektroden meebewegen met gevolg dat de weerstand van de elektrode verandert. Dit heeft een rechtstreeks effect op de metingen. In Figuur 7 is deze storing weergegeven. Het laagfrequent gedeelte is duidelijk zichtbaar in de golving. De frequenties zijn meestal gelegen onder de 1 Hz. [4]



Figuur 7: ECG-signaal met basislijnverschuiving en 50Hz storing (eigen fig.)

### 2.3.3. Bewegingsartefacten

Bewegingsartefacten zijn het gevolg van fysieke beweging. Door beweging van de patiënt zal de huid rondom de elektroden ook bewegen. Hierdoor zal de impedantie aan de elektroden veranderen wat rechtstreeks te zien is in het ECG-signaal. De storing wordt herkend aan plotse amplitudeveranderingen. In Figuur 8 zijn een aantal bewegingsartefacten weergegeven in de groene kaders. Een plotse verandering in het signaal gaat gepaard met hoge frequenties.



Figuur 8: ECG-signaal met bewegingsartefact (eigen fig.)

## 2.4. Digitale signaalverwerking

### 2.4.1. Introductie

Doordat er zoveel ongewenste signalen aanwezig zijn in het gemeten ECG-signaal, zullen de signalen digitaal verwerkt moeten worden. Dit valt onder de noemer digitale signaalverwerking of vertaald naar het Engels digital signal processing (DSP). Het voordeel hiervan is dat het volledig softwarematig gebeurt en dat aanpassingen eenvoudig te implementeren zijn.

Signaalanalyse is een belangrijk onderdeel van DSP. Het is belangrijk te weten welke frequenties er aanwezig zijn en welke van deze frequenties onder de noemer storing vallen. De frequenties worden bepaald met de Discrete Tijd Fourier Transformatie(DTFT).

### 2.4.2. Discrete Tijd Fourier Transformatie (DTFT)

Het is mogelijk om een systeem of een signaal voor te stellen door een verzameling van basissignalen. Deze zijn te bepalen met de DTFT zie formule 3.1. Uit deze formule blijkt dat het signaal N keer discreet wordt bemonsterd. De monsters ondergaan de DTFT wat resulteert in een verzameling van basissignalen met ieder een specifieke frequentie. [5].

Wanneer de analyse op een computer gebeurt, wordt er over de discrete fourier transformatie (DFT) gesproken. Deze transformatie verschilt erin dat de frequentie ( $\omega$ ) ook nog eens discreet bemonsterd wordt. Dit resulteert in een verzameling van discrete frequentie componenten. DTFT wordt als het ware op bepaalde intervallen opnieuw bemonsterd. De formule van de DFT is ter verduidelijking weergegeven in 3.2.

#### Formule:

$$X(e^{j\omega}) = F[x(n)] = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\omega \cdot n} \quad \{3.1\}$$

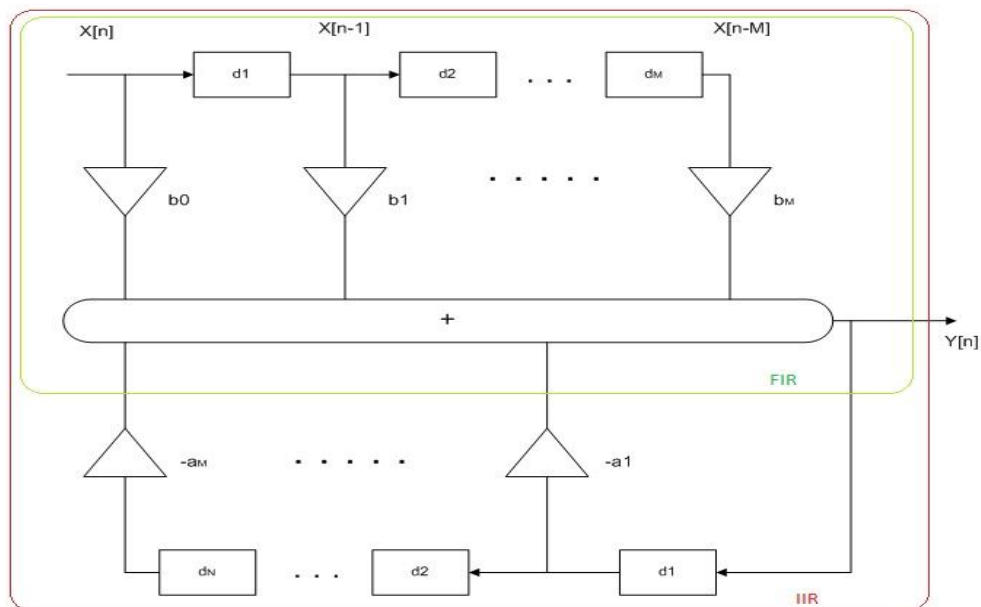
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}k \cdot n} \quad \{3.2\}$$



### 2.4.2.1. Filter voorstelling

Er zijn twee soorten filters namelijk een eindige impulsresponsiefilter en een oneindige impulsresponsiefilter. Deze filters zijn ook wel gekend als finite impulse response filters (FIR-filters) en infinite impulse response filters (IIR-filters). Formule 4.1 geeft de differentievergelijking van een FIR-filter weer. Uit deze formule blijkt dat het resultaat  $Y[n]$  afhankelijk is van de huidige ingangswaarde en van de voorgaande waarden. Formule 4.2 geeft de differentievergelijking van een IIR-filter weer. Nu is de uitgang  $Y[n]$  van de filter tevens afhankelijk van voorgaande uitgangswaarden. Dit betekent dat de impuls responsie in theorie eeuwig zou blijven voortduren. [6]

Voordelen van een FIR-filter zijn dat er een lineaire faseverschuiving in functie van de frequentie is (een constante groepsvertraging) en dat een FIR-filter altijd stabiel is. Bij een IIR-filter is de stabiliteit niet gewaarborgd. Men kan er wel voor zorgen dat een IIR-filter stabiel wordt door een gekende filter te gebruiken. IIR-filters gebruiken voor dezelfde filterkarakteristiek als een FIR-filter veel minder argumenten waardoor er minder eenheidsvertraging is en met gevolg ook minder arithmetische bewerkingen nodig zijn.



Figuur 9: FIR-filter en IIR-filter

De twee types van filters kunnen voorgesteld worden met behulp van een blokdiagram. De blokdiagrammen stellen de differentievergelijkingen voor. In Figuur 9 is een dergelijk blokdiagram weergegeven. Het deel dat zich in het kleinste kadertje bevindt stelt een FIR-filter voor. Het deel dat zich binnen het grootste kader bevindt is een IIR-filter.

De d-blokken stellen een eenheidsvertraging voor, wanneer  $X[n]$  vertraagd wordt met één eenheid dan wordt er  $X[n-1]$  bekomen. Een andere manier om naar deze vertragingen te kijken, als het gaat om de implementatie in een computer system is dat elke  $X[n-M]$  waarde eigenlijk een geheugenlocatie is, dit om waarden uit het verleden te onthouden. Elke van die opgeslagen getallen worden vermenigvuldigd met de coëfficiënten  $a$  of  $b$ . De afzonderlijke resultaten worden dan opgeteld, waardoor het resultaat  $Y[n]$  bekomen wordt.

Filters kunnen ook met behulp van de DTFT worden voorgesteld. Vertrekkende vanuit de differentievergelijking wordt er de DTFT op toegepast. Door de transformatie te doen wordt er van het tijdsdomein overgaan naar het frequentiedomein. In formule 4.3 staat de transformatie weergegeven voor een IIR-filter. Deze techniek wordt veel gebruikt om eigenschappen van de filter te bepalen in het frequentiedomein. Denk maar aan de frequentierespons, het is zelfs mogelijk om de differentievergelijking er mee op te lossen. [6]

**Formules:**

$$y[n] = \sum_{k=0}^m b_k \cdot x[n - k] \quad \{4.1\}$$

$$y[n] = \sum_{k=0}^M b_k \cdot x[n - k] - \sum_{k=1}^N a_k \cdot y[n - k] \quad \{4.2\}$$

$$Y(e^{j\omega}) = \sum_{m=-\infty}^{\infty} b_k \cdot x(e^{j\omega}) \cdot e^{-j\omega \cdot m} - \sum_{n=-\infty}^{\infty} a_k \cdot y(e^{j\omega}) \cdot e^{-j\omega \cdot n} \quad \{4.3\}$$

**2.4.2.2. De z-transformatie**

De z-transformatie is een andere veel gebruikte voorstellingswijze voor het beschrijven en het oplossen van de differentievergelijkingen. Het is eigenlijk een veralgemening van de DTFT (hierboven besproken). In de z-transformatie wordt  $e^{-j\omega}$  vervangen door  $z^{-n}$  (formule 5.1). [7]

De z-transformatie is een handig hulpmiddel dat wordt gebruikt bij het berekenen en analyseren van systemen, in ons geval zijn die systemen digitale filters. Het is mogelijk om de differentievergelijking te herschrijven als een rationale functie (zie 5.1 tot 5.3). Wanneer de waarden van  $z$  in 5.3 ervoor zorgen dat de teller nul wordt dan krijgt men de nulpunten van het systeem. Indien de waarden van  $z$  ervoor zorgen dat de noemer in 5.3 nul wordt dan krijgt men de polen van het systeem.[9]

Het weergeven van de polen en nulpunten in het complexe vlak geeft het polen- en nulpunten diagram. Met behulp van deze polen en nulpunten is het gedrag van het systeem in het frequentiedomein te bepalen. Vandaar dat dit een handig hulpmiddel is in het ontwerpen en het testen van digitale filters. [7]

### **Formules:**

$$y(z) + \sum_{k=1}^N a_k \cdot y(z) \cdot z^{-k} = \sum_{k=1}^M b_k \cdot x(z) \cdot z^{-k} \quad \{5.1\}$$

$$y(z) \cdot \left( 1 + \sum_{k=1}^N a_k \cdot z^{-k} \right) = \sum_{k=1}^M b_k \cdot x(z) \cdot z^{-k} \quad \{5.2\}$$

$$h(z) = \frac{y(z)}{x(z)} = \frac{\sum_{k=1}^M b_k \cdot z^{-k}}{\left( 1 + \sum_{k=1}^N a_k \cdot z^{-k} \right)} \quad \{5.3\}$$

## **2.4.3. Filters**

### **2.4.3.1. Hoogdoorlaatfilter**

Een hoogdoorlaatfilter wordt gebruikt om de laagfrequente storingen uit een signaal te filteren. Dit zijn storingen zoals de basislijnverschuiving. Voor medische toepassingen wordt er veelal een afsnijfrequentie van 0,05Hz genomen. Hierbij gaat men er vanuit dat de patiënten relatief stil blijven zitten en dat de omgeving storingsvrij is. Wanneer er meer beweging is kan er beter een iets hogere afsnijfrequentie genomen worden, dit tussen de 0,67 Hz – 1 Hz. [8]

### **2.4.3.2. Laagdoorlaatfilter**

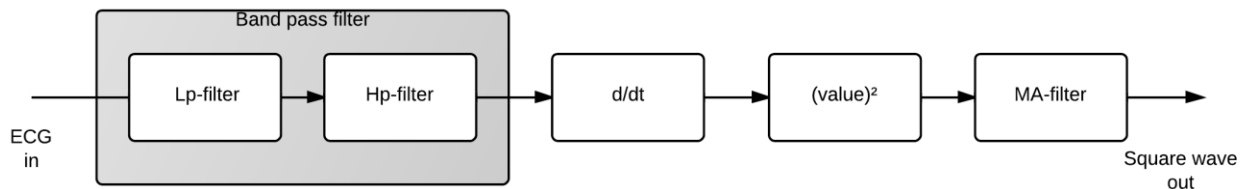
De laagdoorlaatfilter staat in voor het onderdrukken van de hoogfrequente storingen zoals powerline noise en bewegingsartifecten. De frequentie van het QRS-complex is gelegen rond de waarde van 17 Hz. [9] Deze moet zeker zichtbaar blijven, vandaar dat er best een filter boven deze waarde gekozen wordt, maar wel onder de 50Hz.

## 2.5. QRS piekdetectie

### 2.5.1. Pan-Tompkins algoritme

Voor hartritmedetectie is het belangrijk om het QRS-complex te kunnen bepalen. Indien het tijdsinterval is gekend tussen de R-toppen dan is het mogelijk om het harritme hieruit te berekenen. In formule 6.1 staat de berekening weergegeven. Om deze complexen op een robuuste wijze en in reële tijd te detecteren, hebben Pan en Tompkins in 1985 een algoritme ontwikkeld. [9]

In Figuur 10 staat het verloop van dit algoritme. In de eerste stap zal het oorspronkelijke ECG-signaal door een banddoorlaatfilter gaan. Deze filter laat enkel de frequenties rond de 17Hz door. Rond deze frequentie bevindt zich namelijk het QRS-complex, de frequenties daarbuiten zijn niet van belang om het harritme te bepalen. De weg gefilterde frequenties zijn veelal stoorsignalen waardoor ze niet gewenst zijn. [9]



Figuur 10: Pan-Tompkins flowchart

Als tweede stap wordt het gefilterde signaal gedifferentieerd. De differentiaal geeft de verandering van het signaal aan op een bepaald ogenblik. Doordat een QRS-complex een hoge hellingsgraad heeft zal deze waarde groot zijn in vergelijking met andere delen van het ECG-signaal.

De derde stap bestaat er uit om elke waarde die met de differentiaal berekend werd te kwadrateren. Dit zorgt er voor dat alle uitkomsten positief zullen zijn en dat de QRS-complexen extra opvallen.

Als laatste stap wordt dit signaal doorheen een lopend gemiddelde filter gestuurd, ook wel gekend als moving average (MA) filter. Deze filter zorgt ervoor dat er een gemiddelde van een aantal opeenvolgende punten wordt genomen. Belangrijke opmerking hierbij is dat het aantal punten niet te groot mag zijn. Maar de filter moet wel het gemiddelde kunnen nemen van een volledig QRS-complex.

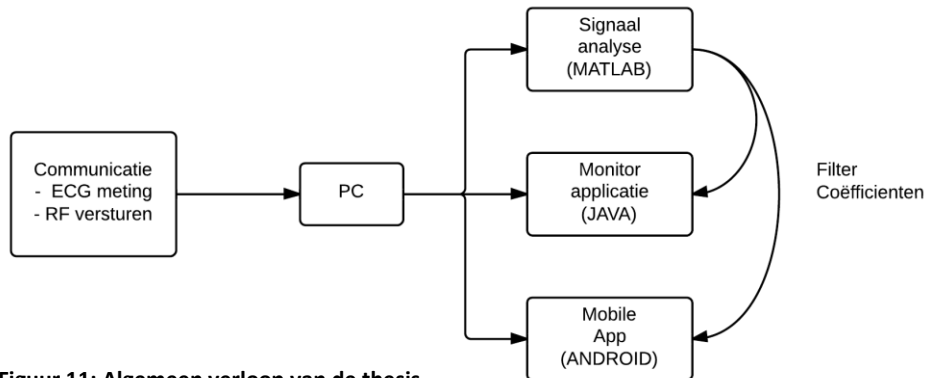
Het hierboven beschreven algoritme geeft een soort blokgolf als resultaat. De stijgende flank van de blokgolf geeft de breedte van het QRS-complex aan. Nu rest er enkel nog de detectie van de stijgende flanken. Dit kan met behulp van een grenswaarde, wanneer een aantal opeenvolgende waarden boven deze grens komen, kan men zeggen dat er een QRS-complex is gedetecteerd.

#### Formule:

$$HR = \frac{60}{RR - interval (sec.)} \left( \frac{bpm}{Min} \right) \quad \{6.1\}$$

### 3. Materialen en methoden

Het doel is om uiteindelijk een toegewijde Java-applicatie te hebben die in staat is om meerder ECG-signalen simultaan en in reële tijd weer te geven. De ECG-signalen bevatten veelal storingen. Om die te onderdrukken zijn er filters nodig. Voor filterontwerp en signaalanalyse wordt het MATLAB softwarepakket gebruikt. Dit programma geeft meer ruimte voor “trail en error”. De filtercoëfficiënten berekent door MATLAB zijn later geïmplementeerd in de eindtoepassing.



Figuur 11: Algemeen verloop van de thesis

In Figuur 11 staat het verloop van de thesis. In begin is er veel met MATLAB gewerkt. Op het ogenblik dat er voldoende kennis was over de te verwerken signalen en de filters is er overgegaan op het realiseren van de applicatie. Als uitbreiding is ook de start gemaakt voor een Android App.

#### 3.1. Communicatie opstelling

In de inleiding is er vermeld dat het doel is om hartpatiënten te monitoren tijdens een loopwedstrijd. Hiervoor is er een opstelling bedacht die hieronder staat weergegeven. Om mobiel te zijn wordt er een golfkarretje gebruikt met daarop een cardioloog. Deze cardioloog kan op het computerscherm een aantal ECG-signalen van de patiënten controleren. Wanneer hij iets abnormaal opmerkt kan hij ingrijpen. De computer is gekoppeld aan een ontvanger die de gegevens van zes verschillende ECG sensoren kan ontvangen. Deze sensoren zijn bevestigd op de patiënten en zullen de gegevens draadloos doorsturen naar de ontvanger.



Figuur 12: Opstelling van het communicatie experiment

### 3.2. ECG sensor node

Voor deze thesis wordt er gebruikt gemaakt van een sensor die ontwikkeld is door imec. Dit toestel staat weergegeven rechtsboven in Figuur 12. Het toestel is in staat om een ECG-sigitaal en accelerometer gegevens waar te nemen. Eens dit is waargenomen wordt het draadloos doorgestuurd. Het RF-sigitaal wordt ontvangen door een basisstation ontvanger. Deze ontvanger communiceert via het RxTx protocol met de PC, waar de gegevenspakketten verwerkt kunnen worden.

Het registreren van het ECG-sigitaal gebeurt m.b.v. een speciale instrumentatieversterker. Deze versterker gebruikt een chopper stabilisatietechniek. Door deze techniek wordt de DC-offset die ontstaat tussen de twee elektrodes geëlimineerd. Bijkomend zal deze techniek een positief effect hebben op de flikkerruis die veroorzaakt wordt door de transistors. [10] De CMRR van deze IA zal hoog zijn, de reden hiervoor werd toegelicht in het hoofdstuk over de werking van de schakeling.

De sensor node heeft de mogelijkheid om de versterking en de afsnijfrequentie van de analoge filters op de chip zelf aan te passen. Wanneer het gemeten sigitaal op een gewenst niveau is, wordt deze bemonsterd door een ADC. De ADC heeft een standaard sample frequentie van 256Hz, deze kan ook aangepast worden. Deze bemonsterfrequentie is ruim voldoende omdat het frequentiespectrum van een ECG-sigitaal onder 128 Hz ligt.

Na de bemonstering stuurt de ECG node het sigitaal samen met de accelerometer gegevens door via een dipoolantenne. Dit soort antennes hebben een typisch radiatiepatroon zoals een “donut”. Het basisstation gebruikt een directionele antenne van het Yagi-type. Dit soort antennes hebben de eigenschap dat het radiatiepatroon in één richting is gelegen, wat het mogelijk maakt signalen over een langere afstand te ontvangen.

De antenne van zowel de ECG-node als het basisstation maken gebruik van een module van het merk Nordic, deze verstuurt zijn gegevens op een frequentie van 2.4 GHz. Met deze module is het mogelijk om snelheden van 1Mbit/s te halen, in het geval van één node is dit veel te hoog. Dit is gedaan zodat de gegevens in blokken (burstmode) verstuurd kunnen worden. Hierdoor wordt er minder energie verbruikt en kan één basisstation gegevens van meerdere noden ontvangen. [10]

### 3.3. MATLAB: Signaalanalyse/verwerking

Vooraleer er aan signaalverwerking wordt gedaan is het altijd een goed idee om eerst het signaal te analyseren. Om de frequentie inhoud van het ECG-sigitaal te bepalen zal MATLAB gebruik maken van de DFT.

Na de analyse is het tijd om de signalen te verwerken met behulp van filters. MATLAB kan die zeer eenvoudig via commando's berekenen. Zo zijn er commando's waarmee men de a- en b-coëfficiënten van een gespecificeerde filter kan berekenen. Deze coëfficiënten kunnen als parameter dienen in een filterfunctie om zo het signaal te conditioneren.

Deze thesis maakt gebruik van IIR-filters. Zoals al vermeldt, gebruiken die voor gelijkaardige specificaties minder operaties dan een FIR-filter. Het gevolg is dat er minder berekeningen nodig zijn en dit komt de reële tijdeigenschappen ten goede.

Er is gekozen voor een laag- en hoogdoorlaatfilter. Door deze twee filters te combineren krijgt men een banddoorlaatfilter. De afsnijfrequentie van de hoogdoorlaatfilter mag niet te laag zijn omdat de applicatie moet werken tijdens het beoefenen van sporten. Voor de laagdoorlaatfilter moet het QRS-complex zeker doorgelaten worden, die bevindt zich rond de 17 Hz.

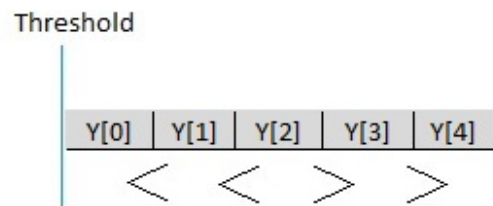
Ook is er gekeken of het mogelijk is om m.b.v. een adaptieve filter het ECG-sigitaal te verbeteren. Deze filter gebruikt als fout-sigitaal de gegevens van de accelerometer. Echter zal dit alleen werken indien er correlatie is tussen het ECG-sigitaal en de accelerometer gegevens. Doordat de accelerometer aan een lagere frequentie bemonsterd wordt dan het ECG-sigitaal, zullen die gegevens eerst een upsampling ondergaan zodat de bemonsteringsfrequentie gelijk is aan die van het ECG-sigitaal.

De techniek van upsampling is eenvoudig, zo voegt men eerst een aantal nullen toe. Deze moeten tussen elk monsterpaar. Daarna wordt het geheel uitgemiddeld met een MA-filter en zal het sigitaal nog versterking ondergaan afhankelijk van hoeveel keer er upgesampled is. Om frequentiespiegeling (imaging) tegen te gaan zal het resultaat nog eens door een laagdoorlaatfilter gaan.

### 3.4. Detectiealgoritmes

#### 3.4.1. R piekdetectie

Om de R-toppen te detecteren is er eerst een eenvoudig algoritme bedacht. Dit is gebaseerd op het principe van een lokaal maximum. Nadat de waarden door een filter zijn gegaan, zal het algoritme de opeenvolgende waarden gaan controleren.



Figuur 13: Eenvoudige piekdetectie

In Figuur 13 is te zien wanneer het algoritme een R-top detecteert. De eerste voorwaarde is dat alle waarden boven een bepaalde grens moeten liggen, deze grens wordt ook wel de drempelwaarde genoemd.

Indien dit het geval is moet de middelste waarde  $Y[2]$  het grootste zijn. Daarbij komt nog dat  $y[0]$  kleiner als  $Y[1]$  moet zijn en  $Y[4]$  kleiner als  $Y[3]$  moet zijn. Is dit het geval dan verondersteld het algoritme dat er een R-Top is. Er is gekeken naar de robuustheid van dit algoritme, vooral wanneer de signalen sterk verstoord zijn.

#### 3.4.2. Pan-Tompkins algoritme

Het Pan-Tompkins algoritme zou beter moeten werken als het vorige. Vooraleer het te implementeren in Java, is het eerst getest in MATLAB. Dit om te verifiëren of het algoritme daadwerkelijk alle QRS-complexen detecteert. In de onderstaande tabel is de sequentie die men dient te doorlopen om aan het Pan-Tompkins algoritme te voldoen kort opgeschreven. Het is ook mogelijk om terug te grijpen naar Figuur 10, hier staat een visuele voorstelling a.d.h.v. een flowchart.

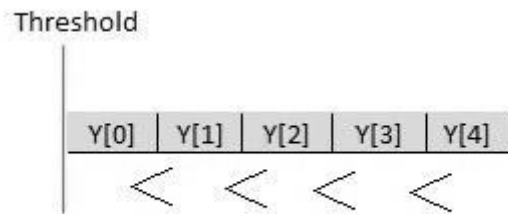
Tabel 1: Pan-Tompkins stappen

Sequentie	Functie	Info
1	hoogdoorlaatfilter	Butterworth 2de orde cutoff 11 Hz
2	laagdoorlaatfilter	Butterworth 2de orde cutoff 22 Hz
3	d/dt	Differentiaal
4	(value) <sup>2</sup>	Kwadraat
5	MA-filter	Moving average 30 punten



### 3.4.2.1. Stijgende flankdetectie

Om aan hartritmebepaling te kunnen doen, moet men de QRS-complexen kunnen detecteren. Het Pan-Tompkins algoritme geeft een blokvolgesignaal als resultaat, waarvan de stijgende flank overeenkomt met het complex.



Figuur 14: detectie van een stijgende flank

Om die flank te detecteren gaat men kijken of vijf opeenvolgende waarden monotoon stijgend zijn. Dit is geïllustreerd in Figuur 14. Ook moeten de waarden weer boven een bepaalde grenswaarde liggen. Wanneer hieraan voldaan is, concludeert men dat er een stijgende flank is. Het moment van detectie wordt gelijkgesteld aan het moment van een R-top.

### 3.5. Programmeren in Java

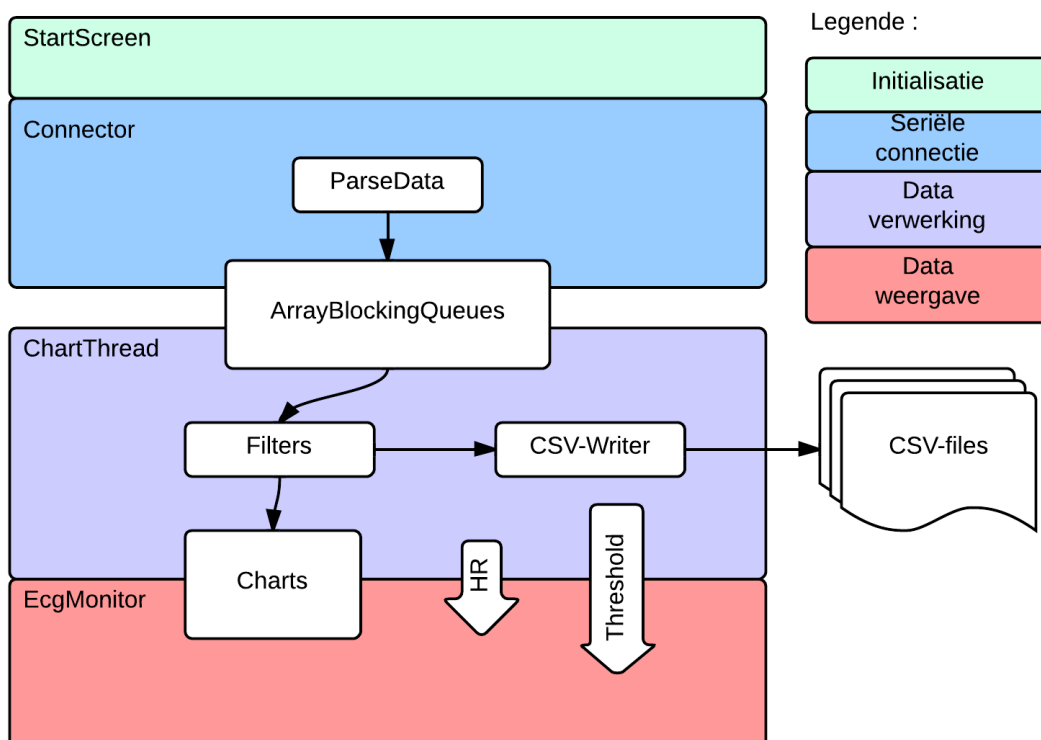
Als programmeertaal is er gekozen voor Java. Dit heeft een aantal redenen. Zo kunnen Java-programma's op eender welk platform functioneren zonder dat ze opnieuw gecompileerd moeten worden. De Java Virtuele Machine (JVM) maakt dit mogelijk. Deze machine zal instaan voor het uitvoeren van de virtuele machine byte-code afkomstig van het oorspronkelijk geschreven programma.

Daarbij komt nog eens dat de JVM verschillende rekenprocessen gelijktijdig kan uitvoeren. Een andere benaming voor zo een rekenproces is een thread. Dit is een stukje code dat op bepaalde tijdstippen wordt uitgevoerd. Dit is zeer handig voor het gelijktijdig verwerken van gegevens en dat komt het reële tijdsaspect ten goede.

Nog een aantal voordelen zijn: een grote community, veel voorbeeldcodes, open source bibliotheken en het feit dat er in de opleiding uitgebreid is gewerkt met deze programmeertaal. Deze feiten hebben ertoe geleid dat de keuze naar Java is gegaan.

#### 3.5.1. Java klassen

Een Java-programma is gebaseerd op klassen. Met deze klassen kunnen verschillende objecten worden aangemaakt. Het zijn de objecten die het uiteindelijke programma zullen vormen. Een aantal belangrijke klassen waarvan de ontwikkelde applicatie gebruik maakt, zullen in de volgende hoofdstukken aan bod komen.



Figuur 15: Objectenstructuur/grafisch overzicht van het programma

In Figuur 15 is de objectenstructuur weergegeven van het programma. Deze afbeelding schept duidelijkheid over hoe de objecten onderling gerelateerd zijn. Het is zeker een goed idee om bij het lezen van iedere klasse even naar deze afbeelding terug te kijken. Hierdoor is het makkelijker om te volgen.

### 3.5.1.1. De connector klasse

De connector klasse zorgt voor de connectie tussen het basisstation en de PC. Java zal gebruik maken van de JSSC en JavaCom bibliotheek. Deze gaan de seriële verbinding tussen de computer en het basisstation mogelijk maken.

JavaCom gaat trachten om de juiste USB/Seriële poort, waaraan het basisstation is gekoppeld, te vinden. Eenmaal die poort gevonden is zal die m.b.v. de JSSC bibliotheek geopend worden, waarna de initialisatie kan plaatsvinden. Voor de initialisatie moeten de baud-snelheid, de gegevens bits, de stop bits en de pariteits bits ingesteld worden. In Tabel 2 zijn de instellingen weergegeven voor het gebruikte basisstation.

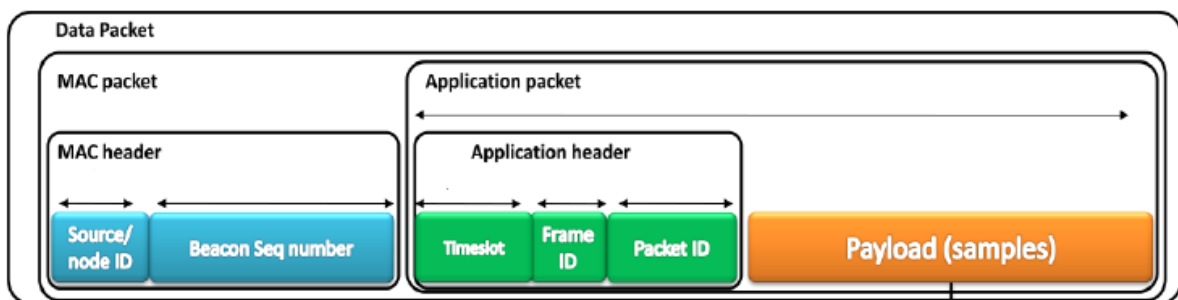
Tabel 2: Seriële instellingen voor het basisstation

	Instellingen
Baud rate	1000000
Data bits	8
Stop bits	1
Pariteits bits	Geen

Als de seriële poort is ingesteld dan zal er een gebeurtenismasker gemaakt worden. Die zorgt ervoor dat er naar bepaalde gebeurtenissen geluisterd wordt. Het masker dient om een *eventlistener* aan te maken. Deze *listener* wordt opgeroepen wanneer er een seriële gebeurtenis plaats vindt. Op het moment dat dit zich voordoet gaat het programma kijken naar de soort van gebeurtenis. In het geval van een *Rx-event*, weet het programma dat er gegevens via de seriële poort van één van de noden ontvangen zijn. De gegevens gaan dan door naar de werkingsklasse ook wel de *ParseData* klasse genoemd

### 3.5.1.2. ParseData klasse

Deze klasse zal instaan voor het ontrafelen van de gegevenspakketten. De pakketten zijn gebouwd op een bepaalde manier, die gespecificeerd is door imec. De opbouw een dergelijk pakket is weergegeven in de onderstaande afbeeldingen. Doordat er met meerdere ECG-sensoren wordt gewerkt moet als eerste gecontroleerd worden welke node de gegevens heeft gestuurd. Elke node heeft een specifiek nummer van 1 tot 6. Deze staan op de "Source/node ID" positie.



Figuur 16: Gegevenspakketten afkomstig v.d. ECG noden

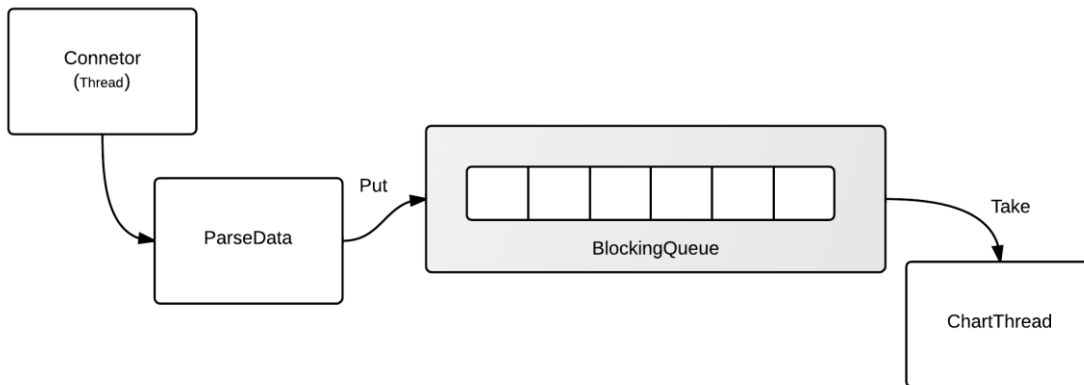
Eens geweten is van welke sensor de gegevens afkomstig zijn, gaat de software controleren wat voor soort gegevens er ontvangen zijn. Voor "frame ID's" gelijk aan nul, staan in de "payload" de ECG-gegevens. Wanneer het ID gelijk is aan één dan bevinden zich hier de XYZ accelerometer gegevens. Wanneer de "payload" een ECG-sigitaal bevat dan zullen er zich hier vijftien opeenvolgende monsters bevinden. Voor de accelerometer gegevens zullen dit vijf XYZ-monsters zijn. De ontvangen gegevens worden weggeschreven in een *arrayblockingqueue*. Dit is een soort interface. In het hierop volgende hoofdstuk wordt er hier dieper op ingegaan.

### 3.5.1.3. ArrayBlockingQueue klasse

De *ArrayBlockingQueue* kan men zien als een soort interface die het mogelijk maakt dat twee threads op een veilige manier met elkaar communiceren. De ene thread kan elementen toevoegen aan de *BlockingQueue*, waarop de andere thread deze elementen er dan op een veilige manier kan uithalen. Deze interface werkt op het “first in first out” (FIFO) principe, wat wil zeggen dat de elementen die er als eerste ingaan er ook als eerste uit zullen komen.

De grootte en het type van de *BlockingQueue* wordt vooraf gespecificeerd, de *queues* van de applicatie gaan 1024 bytes groot zijn. Op het moment dat de *queue* vol is en de thread toch nog elementen probeert toe te voegen, zal de thread pauzeren. Indien de *queue* leeg is en een andere thread er toch elementen probeert uit te halen dan zal deze thread ook wachten tot er weer iets toegevoegd wordt aan de *queue*.

In de onderstaande figuur is weergegeven hoe dat deze *ArrayBlockingQueue* geïmplementeerd is in de applicatie. De connector zal zich gedragen als een soort thread, die op de ogenblikken dat hij gegevens ontvangt het *ParseData* object zal aanspreken. Deze zorgt ervoor dat gegevens zoals de ECG- en ACC-waarden verwerkt worden, waarna ze in een dergelijk *queue* terecht komen. De *ChartThread* zal deze waarden er op specifieke tijdstippen uithalen en er de nodige bewerkingen mee doen.



Figuur 17: ArrayBlockingQueue

#### 3.5.1.4. ChartThread klasse

De objecten van de *ChartThread* klasse zijn van het type *threads*. De tijdstippen om deze uit te voeren zijn zo berekend dat de applicatie de reële tijdeigenschappen van de applicatie nastreeft.

Zoals eerder vermeldt, is het ECG-sigitaal bemonsterd aan frequentie van 256 Hz, dit komt overeen met één monsternamen om de 3,9 milliseconden. In het *ParseData* hoofdstuk is er verteld dat de ECG-gegevenspakketten 15 monsterwaarden bevatten. Dit betekent dat er op tijdsintervallen van 58,6 milliseconden een ECG-gegevenspakket ontvangen wordt van een bepaalde sensor node.

Om in reële tijd te werken, moet de applicatie op deze tijdstippen het ontvangen ECG-sigitaal zo snel mogelijk zichtbaar maken op het controlepaneel. Dit wil zeggen dat de thread om de 58,6 seconden de gegevens uit de *BlockingQueue* moet halen.

Deze klasse zal eveneens de hartritmes berekenen. Het Pan-Tompkins algoritme zal instaan om de QRS-complexen te detecteren. Voor de berekening van het ritme, gaat men het aantal monsters bijhouden tussen vijf opeenvolgende RR-intervallen. Wanneer het aantal monsters gekend is, is het eenvoudig de gemiddelde tijd te berekenen van deze intervallen. (Formule 7.1) Indien men dan 60 deelt door het verkregen tijdsinterval, wordt het aantal hartslagen per minuut verkregen. (Formule 7.2)

De berekende hartslag wordt continue vergeleken met een bepaalde grenswaarde. Wanneer deze waarde boven de grens komt, bevindt de patiënt zich in een gevarenzone en dit wordt tevens weergegeven op het controlepaneel. Er zullen twee grenswaarden zijn één voor “waarschuwing” en één voor “gevaar”. Beide grenzen zijn handmatig in te stellen.

Ook zal de *ChartThread* klasse gebruik maken van de filterklasse. Deze filterklasse zal zorgen voor de conditionering van het ECG-sigitaal. Bijkomende taken van de van de *ChartThread* klasse zijn: het toepassen van het Pan-Tompkins algoritme en weergave van het geconditioneerd sigitaal in de monitorapplicatie.

#### **Formules:**

$$\text{Tijdsinterval} = \frac{\# \text{ samples over } 5 \text{ RR} - \text{ intervallen}}{256 \text{ Hz} \cdot 5} \text{ (sec.)} \quad \{7.1\}$$

$$\text{HR} = \frac{60}{\text{Tijdsinterval (sec.)}} \text{ (HR/Min)} \quad \{7.2\}$$

### 3.5.1.5. Filter klasse

De objecten die verkregen zijn met behulp van deze klasse, staan in voor het filteren van de ECG-signalen. De klasse maakt het mogelijk om de filters die ontwikkeld zijn in MATLAB ook te verkrijgen in de Java-code. Zoals vermeldt in de hoofdstukken over MATLAB zal er enkel gewerkt worden met IIR-filters.

Deze klasse zal werken m.b.v. 2 registers, namelijk met een X-register en een Y-register. Dit om een aantal x en y-waarden bij te houden. De filter vermenigvuldigt deze waarden met de in MATLAB berekende coëfficiënten, waarna alle uitkomsten worden opgeteld om zo te resulteren in de uitgangswaarde van de filter.

### 3.5.1.6. CSV writer klasse

Het doel van de objecten van deze klasse is om coma gescheiden bestanden aan te maken. Die zijn ook wel gekend als CSV-bestanden. De klasse implementeert hiervoor een andere klasse, genaamd *java.io.FileWriter*. Die *filewriter* zorgt ervoor dat de CSV-bestanden geschreven worden m.b.v. een stream. Dit wil zeggen dat na de verwerking van de gegevens het mogelijk is om ze meteen weg te schrijven. Dit biedt als bijkomend voordeel dat er zeker geen gegevens verloren gaan.

Voor elke sensor node worden 2 CSV-bestanden aangemaakt. Eén bestand zal de ECG-gegevens bijhouden, hierin komen zowel de ruwe als gefilterde gegevens te staan. Het tweede bestand zal dienen voor de accelerometer gegevens. Hier krijgt elke richting een andere kolom. Achteraf kunnen deze bestanden voor offline analyse zeer nuttig zijn.

De *chartThreads* zullen de CSV-objecten aanspreken op de momenten dat er gegevens moeten worden weggeschreven.

### 3.5.1.7. Chart klasse

Het is gewenst om de ECG-gegevens te tonen in een grafiek. De klasse die hiervoor zal instaan is de *Chart* klasse. Van deze klasse zullen er zes objecten zijn. Elk object stemt overeen met één grafiek waarin het ECG-signaal van één sensor node te zien is. De klasse maakt gebruik van de *JFreechart* bibliotheek.

Met deze bibliotheek is het zeer eenvoudig grafieken aan te maken, en op een dynamische manier waarden aan de grafiek toe te voegen. Dit zorgt voor het reële tijdseffect. Ook kunnen grafieken opgeslagen worden in verschillende bestandstypes (PNG,JPEG,PDF).

### 3.5.1.8. EcgMonitor klasse

Deze klasse maakt het eigenlijke controlepaneel aan. Die zorgt ervoor dat de zes *chart* (grafiek) objecten naar de voorgrond komen. Daarnaast zullen er ook tekst-vakken zijn waarin de hartslagen staan. Deze hartslagen corresponderen met de daarnaast weergegeven ECG-signalen. Ook zullen de namen van de patiënten naast de grafiek waarop ze betrekking hebben te zien zijn.

## 3.6. Android applicatie

Bij wijze van uitbreiding is de start gemaakt voor een Android applicatie. Android is een besturingssysteem dat gebruikt wordt op mobiele telefoons en handcomputers. Doordat de ontwikkeling van een dergelijke applicatie gebeurt met Java, was het eenvoudig deze overstap te maken.

Als ontwikkelingstoestel is er een *Samsung Galaxy S4* gebruikt. Dit omdat hij beschikt over Bluetooth Low Energy (BLE) en een Jelly Bean 4.3 besturingssysteem. Deze zijn nodig om met een ECG-sensor, ook ontwikkeld door imec, te communiceren.

### 3.6.1. ECG-sensor van imec



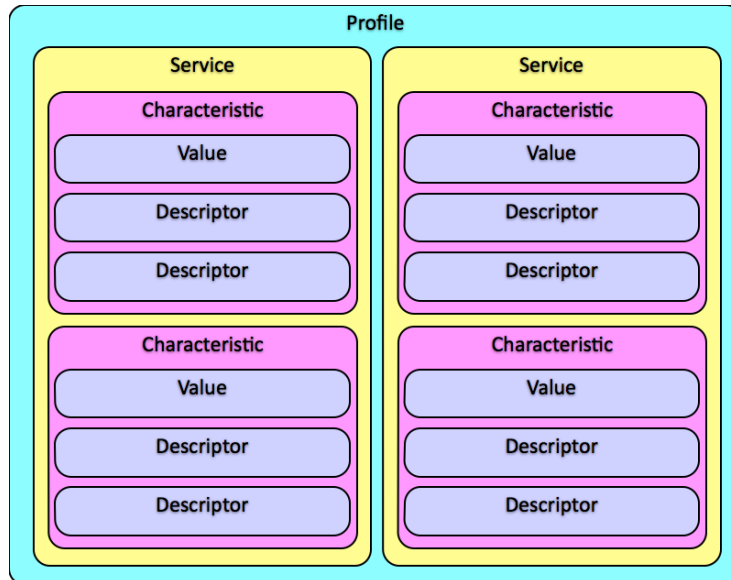
Figuur 18: ECG-sensor van imec op BLE

Op Figuur 18 is de sensor van imec te zien, deze werkt op BLE. De ECG-sensor wordt op de borststreek bevestigd met twee elektroden. Het toestel beschikt over de mogelijkheid om verschillende parameters door te sturen. Deze parameters zijn: accelerometer gegevens, ECG-gegevens, batterijniveau en de hartslag. Voor deze Android App is er geopteerd om enkel de hartslag en het ECG-signaal weer te geven.



### 3.6.2. BLE met Android

BLE wordt pas ondersteund vanaf Android versie 4.3 ofwel api level 18. Een andere benaming voor BLE is ook wel Bluetooth Smart.



Figuur 19: BLE GATT profiel [11]

BLE is rond een *GATT profile* gebouwd. De opbouw van een dergelijk profiel staat weergegeven in bovenstaande figuur. Het doel hiervan is dat alle BLE-toestellen eenzelfde structuur implementeren zodat er een standaard is. Deze standaard zorgt ervoor dat de gegevens van alle BLE-toestellen op ongeveer dezelfde manier te verkrijgen zijn. [11]

Het profiel bestaat op zijn beurt uit *services*, deze bakenen een bepaald type van gegevens af, zoals ECG-gegevens of accelerometer gegevens bijvoorbeeld. De services waarover de imec-sensor beschikt staan in Tabel 3. Iedere service is op zijn beurt weer opgebouwd uit *characteristics*. [11]

De *characteristic* bestaat minimaal uit een *value* en een *discriptor*. De *value* van de *characteristic* van de *ECG-service* zal bijvoorbeeld de ECG-gegevens bevatten. De *discriptor* kan gebruikt worden om een notificatie in te stellen of om te kijken of een *characteristic* readable of writeable is. [11]

Om services aan te spreken wordt er gebruik gemaakt van een *universally unique identifier* (UUID). Deze staan ook in de onderstaande tabel.

Tabel 3: Services van de imec-sensor

	Service	UUID
1	Battery Service	180f
2	Heart Rate service	180d
3	ECG service	ffb0
4	Accelerometer service	ffa0
5	Enable service	ffd0



## 4. Resultaten en Discussie

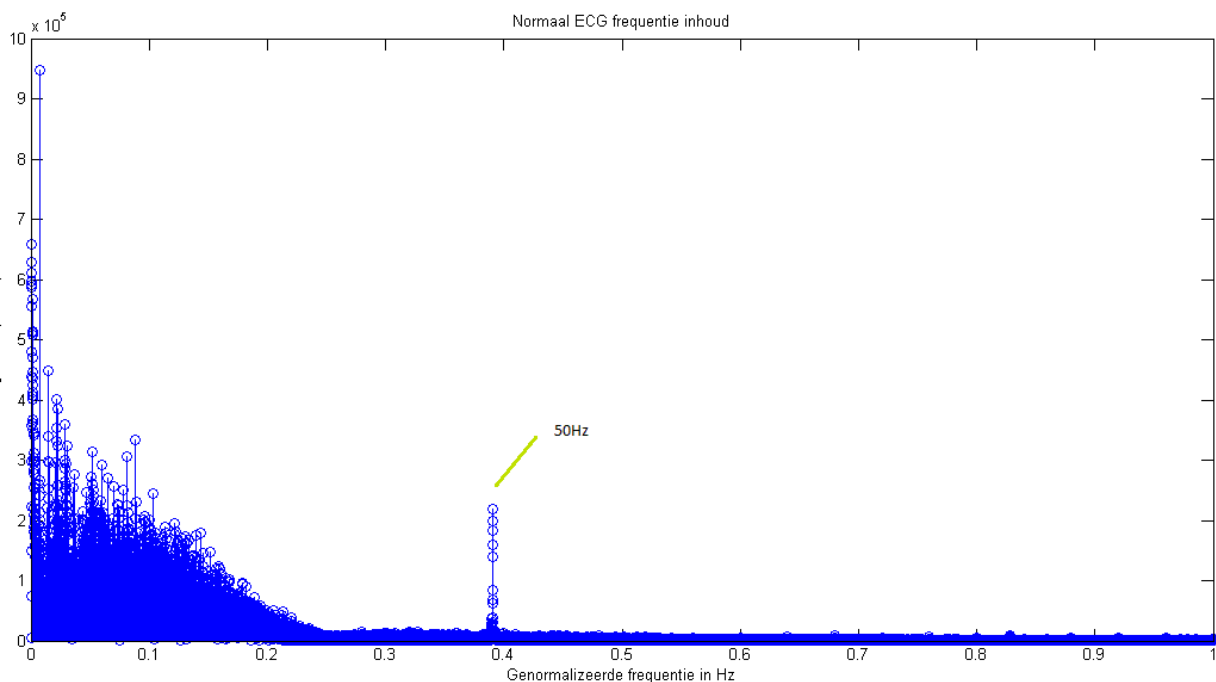
In het hoofdstuk over materialen en methoden is verteld hoe dat de gemeten signalen ontvangen worden via de computer. Eenmaal deze zijn ingelezen, moet er met behulp van filters signaalconditionering gebeuren. Ook zorgt een speciaal algoritme voor een robuuste en accurate hartslagdetectie. Omdat het doel was om zes hartpatiënten te monitoren tijdens de loopwedstijd “Genk loopt” op 4 mei, moest de werking van de applicatie intensief getest worden.

### 4.1. MATLAB: Signaal analyse/verwerking

#### 4.1.1. Signaal analyse

In Figuur 20 is de frequentie inhoud van een ECG-signaal te zien. Dit signaal is opgenomen, met de sensor van imec, wanneer een gezonde vrijwilliger in rust was. Merk op dat de frequentie-as genormaliseerd is, d.w.z. dat de bemonsteringsfrequentie van 256 Hz wordt verdeeld over  $2\pi$ .

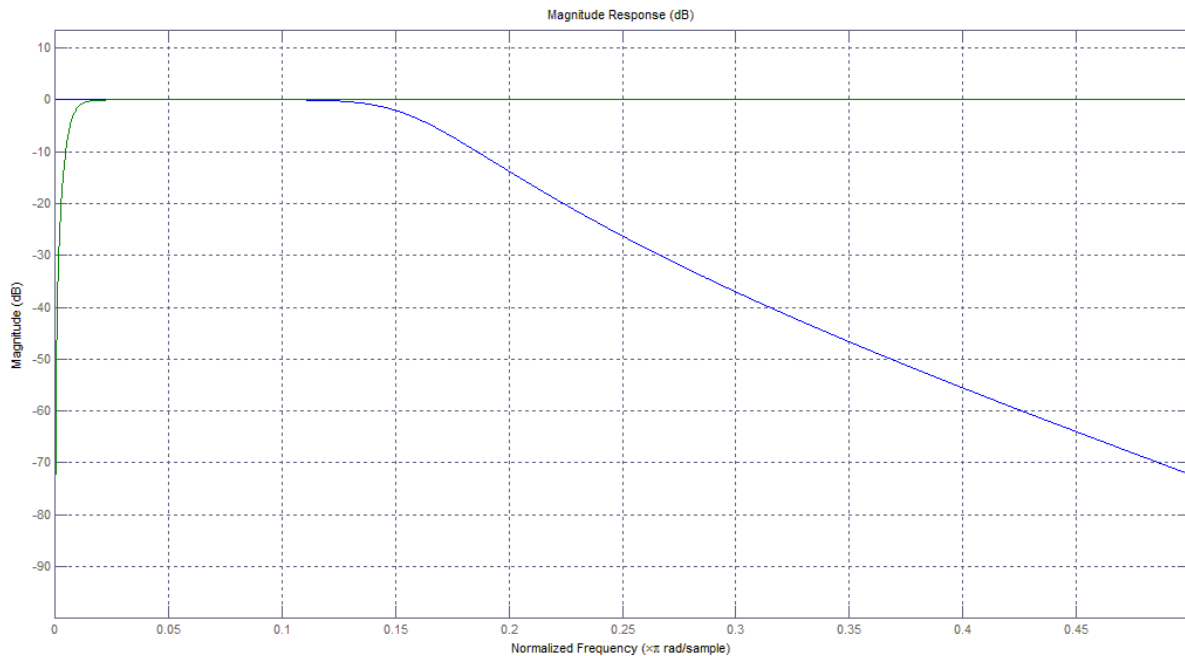
In de afbeelding is te zien dat de voornaamste frequenties onder de  $0,25\pi$  liggen, dit komt overeen met een frequentie van  $32\text{ Hz}^1$ . Bovendien valt de grote aanwezigheid van het 50 Hz component op, de oorzaak hiervan is de netspanningsruis die verklaard werd in de literatuurstudie. Als laatste bemerking is er de sterke aanwezigheid van de laagfrequente signalen. Deze zijn te wijten aan de basislijnverschuiving en de DC-offset die ontstaat tussen de elektroden.



**Figuur 20: Frequentie inhoud van een gemeten ECG-signaal**

<sup>1</sup>  $0,25 \cdot \pi = 0,25 \cdot 128\text{ Hz} = 32\text{ Hz}$

Uit deze analyse kan men concluderen dat er het beste geopteerd wordt voor een banddoorlaatfilter. Deze filter is te verwezenlijken door een combinatie van een hoogdoorlaatfilter en een laagdoorlaatfilter. De frequentie van de hoogdoorlaatfilter moet zo gekozen worden dat de laagfrequente storingen weg worden gefilterd. Voor de laagdoorlaatfilter kan het beste een frequentie onder de 50Hz gekozen worden.



**Figuur 21: Banddoorlaatfilter (groen = hoogdoorlaatfilter, blauw = laagdoorlaatfilter)**

In Figuur 21 is de frequentieresponsie van de resulterende banddoorlaatfilter weergegeven. Dit is de filter die gebruikt is in de uiteindelijke Java applicatie. De groene lijn representeert de response van de hoogdoorlaatfilter waarop de blauwe die van de laagdoorlaatfilter vormt. Deze filter zal de frequentie inhoud van Figuur 20 beïnvloeden.

## 4.1.2. Signaal verwerking

### 4.1.2.1. Hoogdoorlaatfilter

Deze filter onderdrukt de effecten van de basislijnverschuiving. Deze worden voornamelijk veroorzaakt door ademhaling en bewegingen. De storing manifesteert zich daarom in de lagere frequenties. Omdat de applicatie aan monitoring moet doen tijdens het lopen, is er gekozen voor een afsnijfrequentie van 1 Hz, wat een robuuster systeem zou moeten geven. Als filter is er geopteerd voor een Butterworth 2<sup>de</sup> orde IIR-filter, dit omdat het voldoende bleek te zijn om de ongewenste ruis te onderdrukken. In Figuur 22 is het effect van deze filter te zien, bovenaan een sterk verstoort ECG-sigitaal. Onderaan is echter te zien dat de basislijnverschuiving volledig is weggewerkt en dat het signaal mooi rond nul ligt. Daarbij komt dat er maar twee tijdsvertragingen zijn. Wat goed is om in reële tijd te werken.

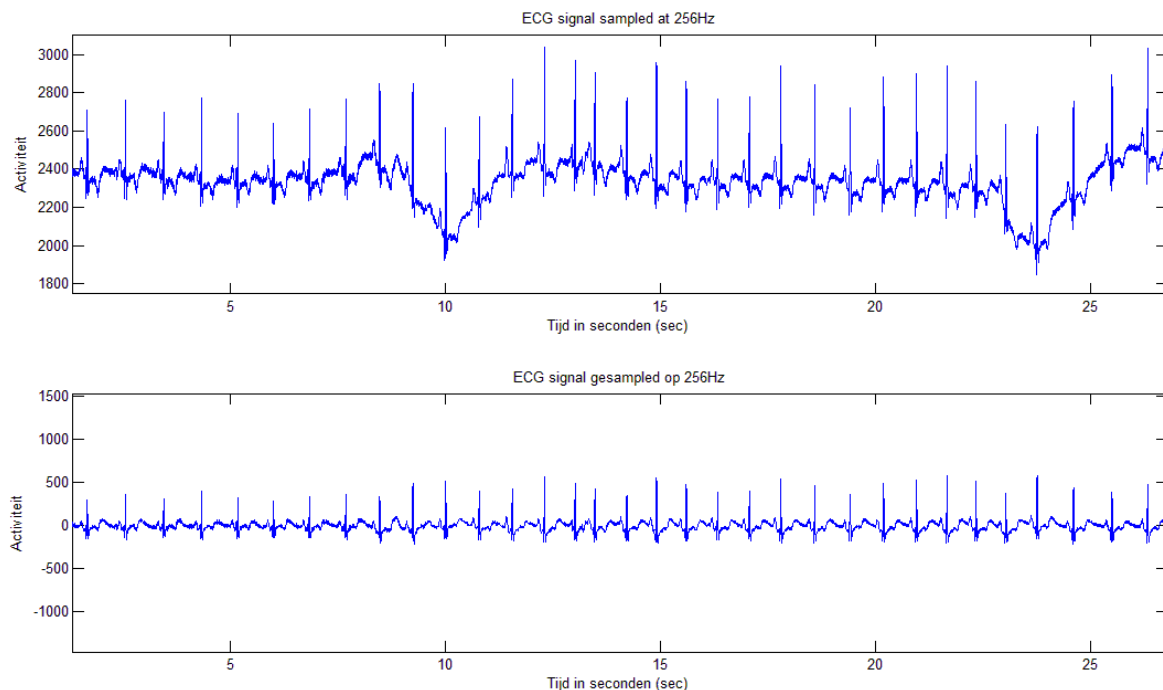
## MATLAB code:

```
%-----Filters maken-----  
%fs = monsterfrequentie  
%fn = naquistfrequentie  
%fb = afsnijfrequentie  
fs=256;  
fn=fs/2;  
fb=1;  
w1=fb/fn;  
%----- Hoogdoorlaat op 1Hz Butterworth 2de orde-----  
[g,k] = butter(2,w1,'high');%2de orde  
fvtool(g,k)  
ECG_HighPassButter = filter(g,k,vEcg);
```

De fvtool in MATLAB is zeer handig. Hierin staan namelijk de a- en b-coëfficiënten van de differentievergelijking. Door deze in te vullen in de vergelijking wordt het gewenste filtereffect verkregen. Deze kan men later gebruiken in de Java-applicatie. De coëfficiënten zijn weergegeven in Tabel 4.

Tabel 4: Coëfficiënten van de hoogdoorlaatfilter

	A	b
1	1	0,982794708297877
2	-1,96529337262269	-1,96558941659575
3	0,965885460568817	0,982794708297877



Figuur 22: Hoogdoorlaatfilter 2de orde Butterworth, afsnijfrequentie op 1Hz

#### 4.1.2.2. Laagdoorlaatfilter

De frequentie van het QRS-complex ligt rond de 17 Hz. Dit wil zeggen dat de laagdoorlaatfilter deze frequentie zeker moet doorlaten. De storing van het net die op 50 Hz ligt is echter niet gewenst. Vandaar dat er is geopteerd om de afsnijfrequentie op 20 Hz te leggen.

Hier is tevens voor een Butterworth filter gekozen maar dit maal voor een 6<sup>de</sup> orde filter. Dit levert een grotere verzwakking op in vergelijking met een 2<sup>de</sup> orde filter. Een hogere orde zorgt wel voor meer tijdsvertraging. Deze tijdsvertraging is echter niet significant door de grote rekenkracht van de computer. Het effect van deze filter is weergegeven in Figuur 23. Hier is te zien dat de 50Hz ruis volledig weg is waardoor de P-toppen en T-toppen duidelijk zichtbaar zijn.

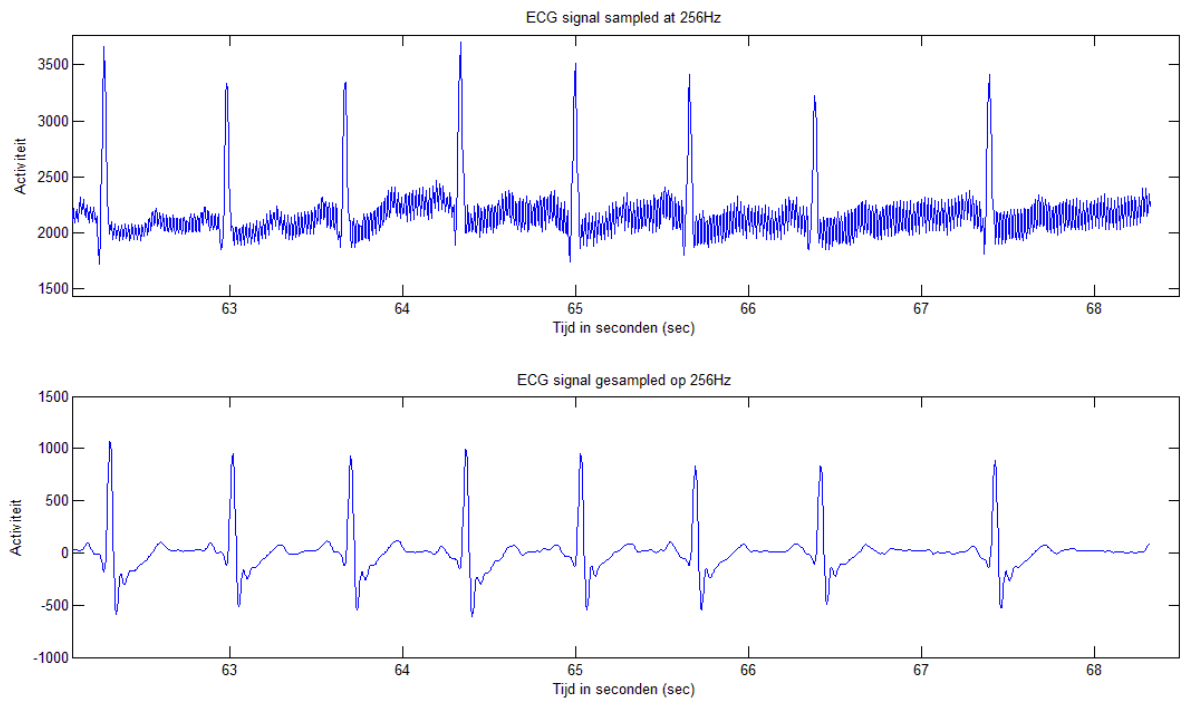
In Tabel 5 staan de coëfficiënten die de Java-applicatie later zal gebruiken om de filter te bekomen.

#### MATLAB code:

```
%----- Laagdoorlaat op 20Hz Butterworth 6de order -----  
f1=20;%Afsnijfrequentie  
w2=f1/128;  
[d,c] = butter(6,w2,'low');  
fvtool(d,c)  
ECG_LowPassButter = filter(d,c,ECG_HighPassButter);
```

Tabel 5: Coëfficiënten laagdoorlaatfilter

	a	b
1	1	0,000094521
2	-4,106914936	0,000567125
3	7,245028565	0,001417821
4	-6,979549163	0,001890428
5	3,856859052	0,001417821
6	-1,155807039	0,000567128
7	0,14643289	0,000094521



**Figuur 23: Effect van de laagdoorlaatfilter**

#### 4.1.2.3. Adaptieve filter

In deze sectie is er gekeken of er een correlatie is tussen de accelerometer gegevens afkomstig van de ECG-node (en dus de bewegingen volgt van een gezonde vrijwilliger) en het gemeten ECG-sigitaal.

Dit omdat een adaptieve filter enkel nut heeft wanneer er een verband is tussen de storing op het ECG-sigitaal en de accelerometer waarden. Dit wordt gecontroleerd m.b.v. correlatie. Eerst zal upsampling ervoor zorgen dat het ACC-sigitaal van 64 Hz naar 256 Hz bemonsteringsfrequentie gaat. Dit sigitaal moet dan doorheen een laagdoorlaatfilter met een afsnijfrequentie van 60 Hz. Deze filter voorkomt een fenomeen dat frequentiespiegeling (imaging) wordt genoemd.

In Figuur 24 is het ECG-sigitaal samen met een ACC-sigitaal weergegeven. Op een bepaald moment gaat de persoon van wandelen over in lopen. Dit is duidelijk te zien in het ACC-sigitaal, op dat moment wordt het ECG-sigitaal ook sterk verstoord. Wanneer men kijkt of er een correlatie is tussen deze twee signalen dan blijkt dat er een correlatie coëfficiënt is van 0.08, wat zeer weinig is.

Op dezelfde manier is er gekeken naar de correlatie bij vijf andere signalen zowel bij het hardlopen als bij het lopen dit om zeker te weten dat het niet ging over een toevallige slechte meting. In de onderstaande tabel zijn de correlaties weergegeven voor deze metingen.

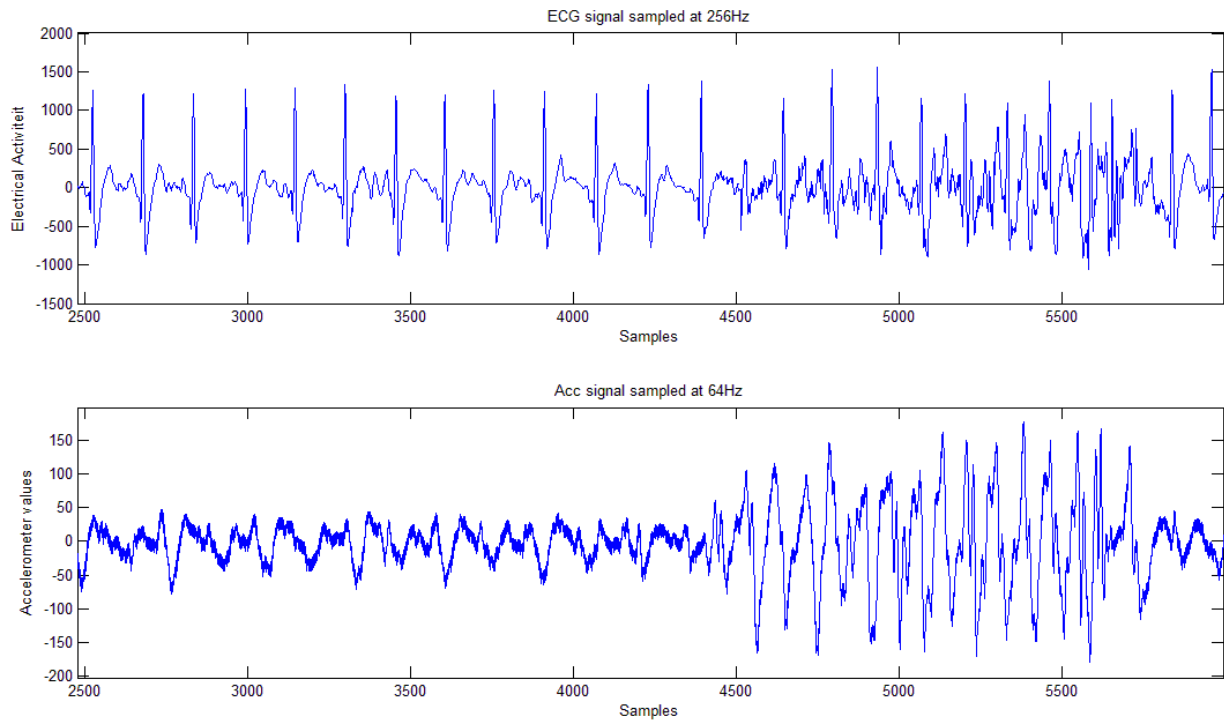
Tabel 6: Correlatie ECG-sigitaal met accelerometer gegevens

Meting	Correlatie	Info
1	0,1	Hardlopen
2	0,11	Hardlopen
3	0,06	Lopen
4	0,1	Lopen
5	0,025	Lopen

Doordat er zo weinig correlatie is tussen de ECG- en ACC-gegevens zal een adaptieve filter geen positief effect hebben op het ECG-sigitaal. De storing in het ECG-sigitaal wordt voornamelijk veroorzaakt doordat de huid rondom de elektroden wordt vervormd. Dit zorgt voor een impedantie verandering en dat reflecteert zich in het ECG-sigitaal.

Een mogelijke oplossing is het plaatsen van accelerometers op de elektroden. Hierdoor bevinden de accelerometers zich dicht bij de oorzaak van de storing waardoor er misschien een nauwkeuriger verband is tussen de impedantie verandering en de accelerometer gegevens. Door de beperking van de ECG-node die gebruikt is, is het niet mogelijk geweest om dit verder te onderzoeken.





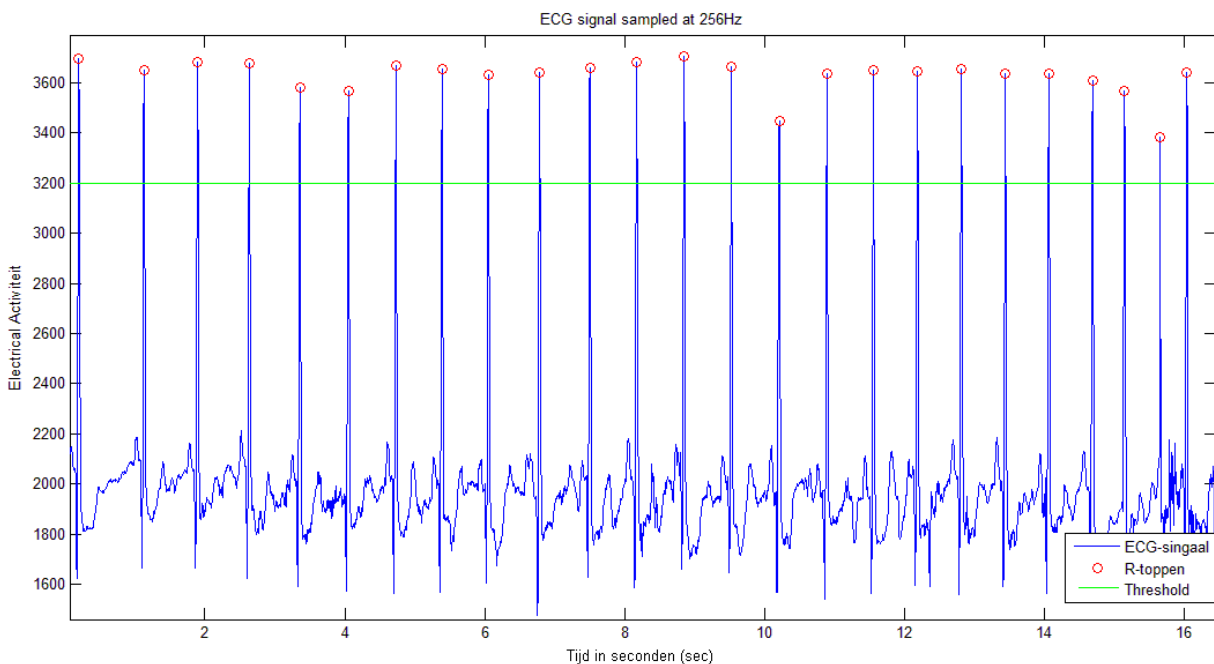
**Figuur 24: ECG-sigitaal (= bovenste grafiek) en ACC-sigitaal (= onderste grafiek)**

## 4.2. Detectiealgoritmes

Om te controleren of de hartslag-detectiealgoritmes effectief werken zijn deze gecontroleerd op een aantal signaltypes. Zo zijn er metingen gedaan op personen in rusttoestand alsook op personen die aan het lopen waren. Met behulp van MATLAB is dan gecontroleerd of alle hartslagen gedetecteerd werden.

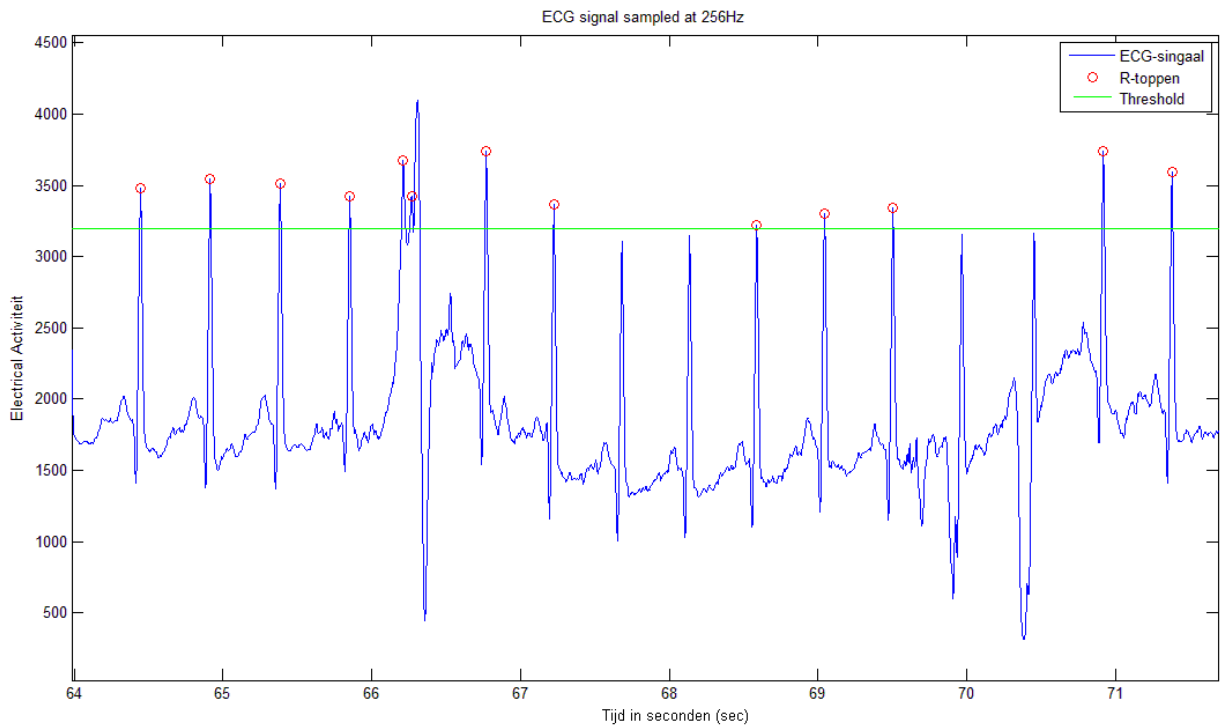
### 4.2.1. R-top detectie

Figuur 25 hieronder geeft de werking van het algoritme weer. Het signaal is afkomstig van een gezonde vrijwilliger die rustig aan het wandelen is. De groene lijn geeft de grenswaarde weer die statisch is ingesteld op 3200. De rode bolletjes geven de waargenomen R-toppen weer. Zoals blijkt uit onderstaande figuur kan het algoritme van een zuiver ECG-signaal de R-toppen zeer goed detecteren.



**Figuur 25: Detectie op het ECG-signaal in rust**

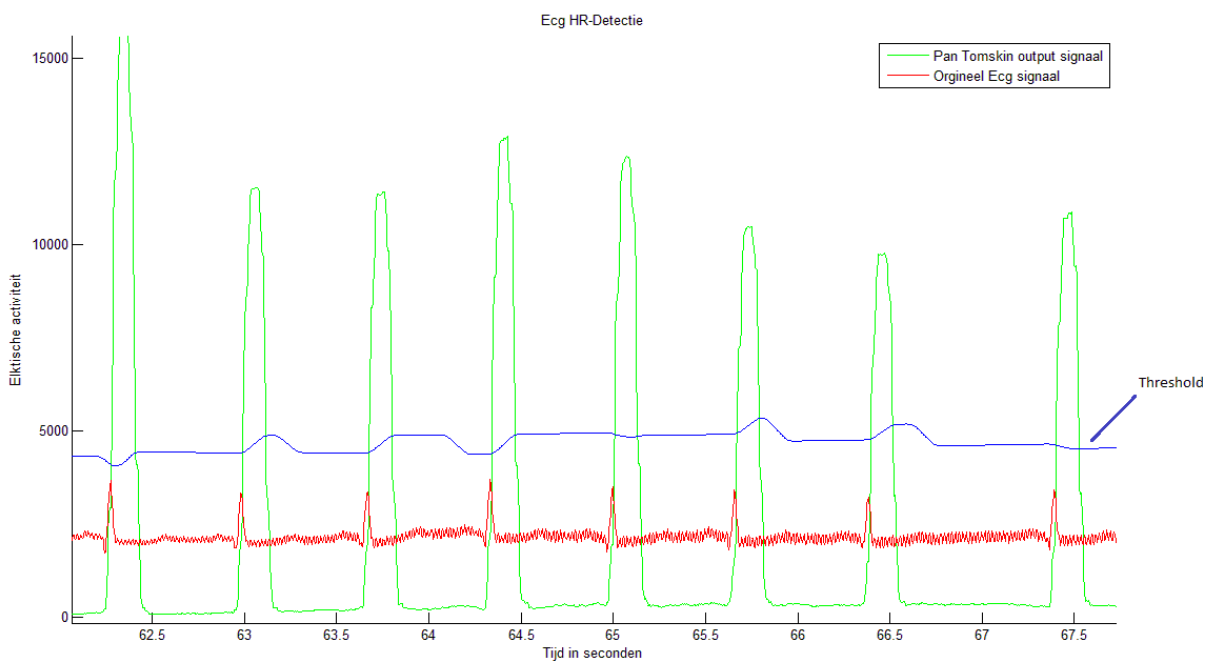
De situatie verandert echter wanneer het signaal afkomstig is van een persoon die aan het lopen is. Het signaal in Figuur 26 is opgenomen bij een vrijwilliger die aan het lopen was tegen een snelheid van 8,5 km/h. Zoals blijkt zullen sommige pieken niet meer te detecteren zijn. Dit omdat ze onder de grenswaardelijn liggen. De grenswaarde lager leggen zou een oplossing kunnen zijn of deze grenswaarde dynamisch laten veranderen, maar dan bestaat de kans dat sommige pieken onterecht gezien worden als een R-top. Zoals dit zich al voordoet in Figuur 26. (na de 66 seconden)



**Figuur 26: Detectiealgoritme op een persoon die aan het lopen is**

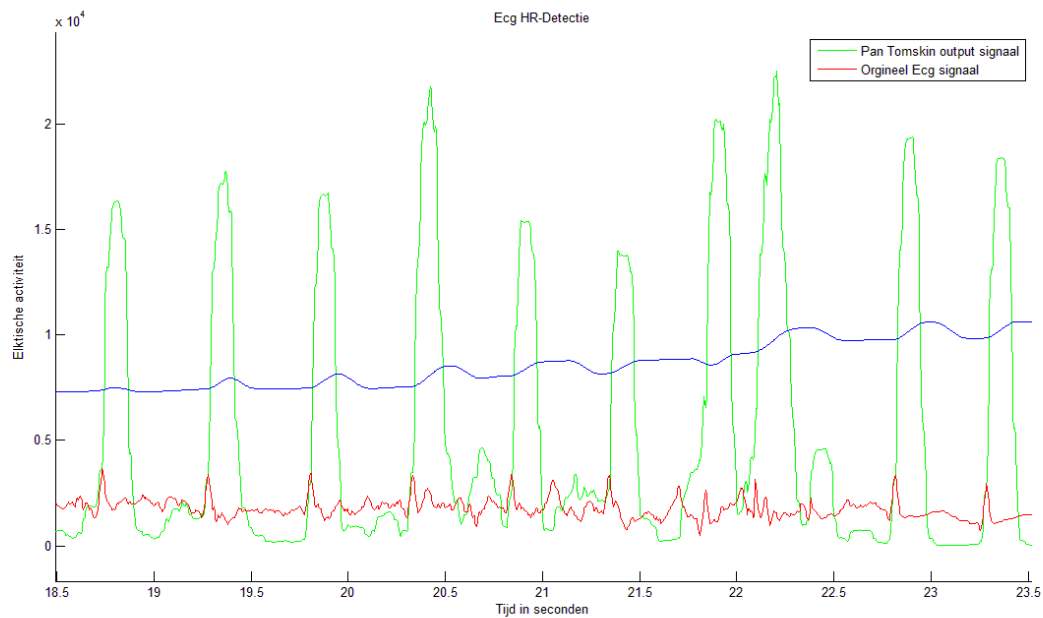
#### 4.2.2. Pan-Tompkins algoritme

Het signaal in Figuur 27 is opgenomen bij een gezonde vrijwilliger die rustig aan het wandelen is. Wat opvalt, is dat het QRS-complex zeer goed wordt waargenomen door het algoritme. Zoals vermeld werd in de literatuurstudie komt de stijgende flank van het blokgolfvormig signaal exact overeen met dit QRS-complex. Er kan dus gesteld worden dat het algoritme de complexen goed detecteert wanneer een persoon in rust is.



**Figuur 27: Pan-Tompkins detectiealgoritme tijdens een minimale inspanning**

Het doel van deze thesis is echter om een applicatie te realiseren die het mogelijk maakt om aan monitoring te doen tijdens het beoefenen van sport. Dit wil zeggen dat er geen ideaal ECG-signaal binnenkomt. Vandaar dat het algoritme nogmaals gecontroleerd is met een ECG-signaal dat is opgenomen wanneer een persoon aan het lopen was.



**Figuur 28: Detectie van QRS-complex tijdens actieve inspanning**

Het ECG-signaal in de bovenstaand figuur is sterk verstoord, desondanks slaagt het algoritme er in de complexen nog goed te detecteren. Hieruit valt te besluiten dat dit algoritme goed werkt en bijgevolg zal dit de beste oplossing zijn voor de uiteindelijke applicatie.

In Figuur 28 geeft de blauwe lijn de grenswaarde aan. Deze is berekend met een MA-filter. Deze filter berekent een gemiddelde over 1500 opeenvolgende punten. Wat een dynamische grenswaarde oplevert. Dit is tevens te zien in Figuur 28, want de lijn stijgt door de toenemende storing. Een dynamische grenswaarde zorgt voor een detectiealgoritme dat robuuster en accurater is.

### 4.3. Java programma

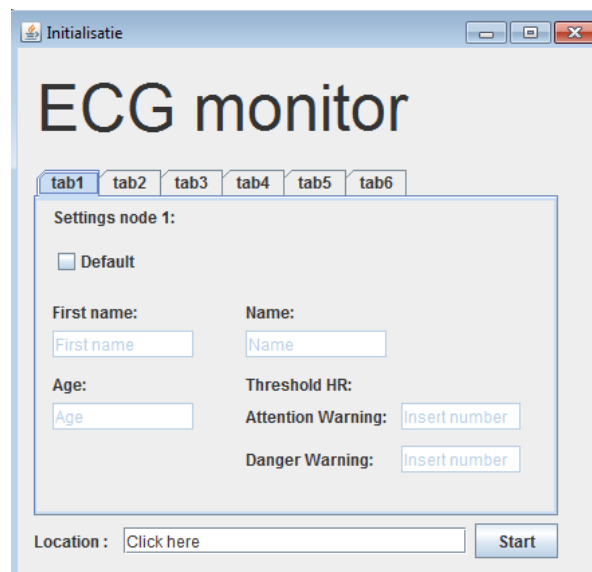
De gerealiseerde applicatie valt niet onder de noemer testapplicatie maar zal werkelijk bruikbaar zijn in een revalidatieomgeving. Zijn eenvoudige opzet en platformonafhankelijkheid maken deze applicatie tot een handige monitoringtool.

De werking van het programma wordt uitgelegd aan de hand van een diagram. Dit diagram geeft de relatie tussen de belangrijke onderdelen weer. Deze belangrijke onderdelen zijn de objecten die gemaakt zijn van de eerder besproken klassen. In Figuur 15 staat het diagram weergegeven met een overzicht van de objectenstructuur.

#### 4.3.1. Werking monitorings applicatie

Wanneer het programma wordt opgestart, komt eerst het startscherm naar de voorgrond. Deze heeft een opmaak zoals te zien is in Figuur 29. Het startscherm wordt verkregen door een *startScreen* object aan te maken. Dit object zal het grootste deel van de initialisatie op zich nemen. Uit de figuur blijkt dat er een aantal zaken moeten gebeuren, zo vraagt de applicatie om de locatie in te stellen waar de CSV-bestanden terecht moeten komen. Daarnaast kan er gekozen worden om de namen en leeftijden van de patiënten in te vullen. Deze komen dan bij in de CSV-bestanden te staan.

Wanneer men op “start” klikt, controleert het programma of deze zaken allemaal zijn ingevuld. Is dit niet het geval, dan wordt er een boodschappenvenster getoond met daarin de handelingen die nog dienen te gebeuren.



Figuur 29: Startscherm

Op het ogenblik dat alles correct is ingevuld kan de applicatie verder gaan met de initialisatie. Waarna de *chartThreads* starten. Het *connector* object zal instaan voor het ontvangen van de seriële gegevens. De ontvangen gegevens gaan dan naar een *ParseData* object dat ervoor zorgt dat alleen de essentiële gegevens overblijven. Als het gaat over essentiële gegevens dan bedoelt men de ECG- en ACC-waarden.

Het *connector* object krijgt van zes verschillende ECG-noden gegevens binnen. Om achteraf nog te weten van welke noden deze ECG- en ACC-gegevens kwamen, wordt die in verschillende *ArrayBlockingQueue* objecten weggeschreven.

Er zijn zes *chartThread* objecten die de reële tijd gegevensverwerking van de zes ECG-noden op zich nemen. Door deze threads op de juiste momenten te starten is het mogelijk om de reële tijds eigenschappen na te streven.

Gegevensverwerking houdt in: filtering van de ECG-signalen, hartritmedetectie en grenswaardecontrole. De harstslagdetectie gebeurt door het Pan-Tompkins algoritme. De grenswaardecontrole moet aangeven wanneer een bepaalde patiënt een te hoge hartslag heeft.

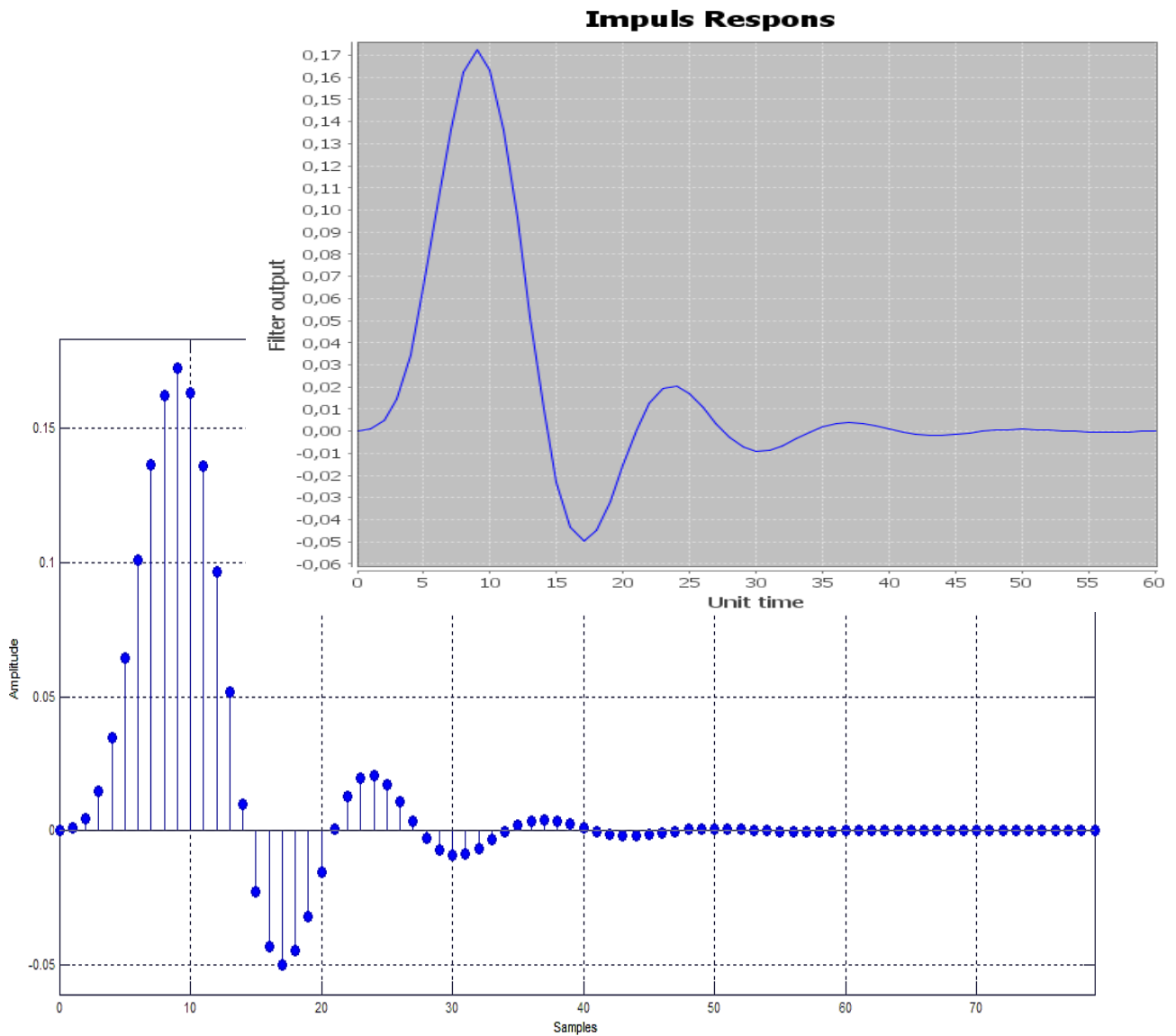
De gefilterde signalen zullen terechtkomen in aparte grafieken die aangemaakt zijn door de *chart* objecten, deze grafieken zijn zichtbaar in het uiteindelijke controlepaneel. Dit controlepaneel zal gemaakt worden door een *EcgMonitor* object. Hierin staan de ECG-signalen, de harstslagen, patiëntnamen en de grenswaarde indicatie. De applicatie is weergegeven in Figuur 30. De kleuren zijn zo gekozen zodat de ECG-signalen bij zonlicht nog goed te herkennen zijn.



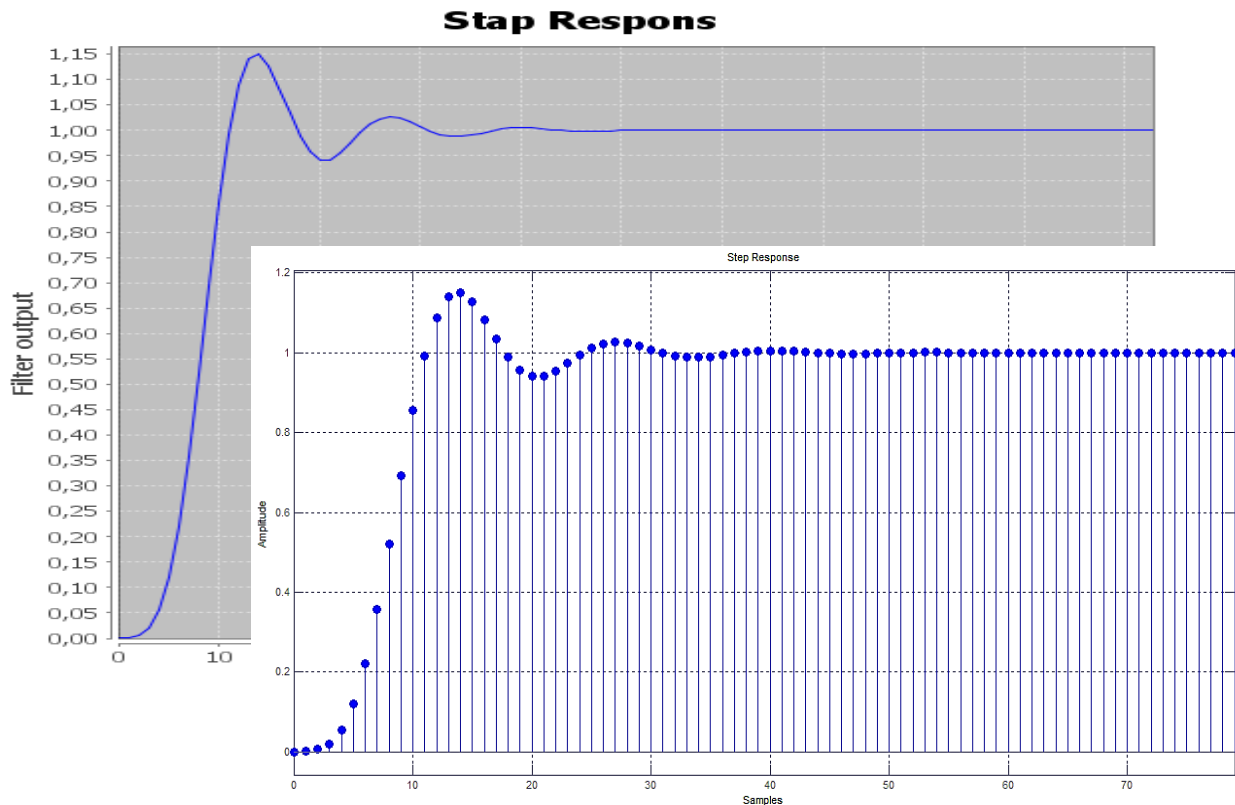
**Figuur 30: Controlepaneel**

### 4.3.2. Controle van de filterwerking

Om te controleren of de zelfgemaakte klasse werkelijk een juiste filterwerking heeft, is de stapresponsie en de impulsresponsie van de laagdoorlaatfilter vergeleken met die van MATLAB. Deze moeten exact overeenkomen. Indien dit niet het geval is dan heeft de filter klasse geen juiste werking. In Figuur 31 en Figuur 32 staan deze weergegeven. Zoals te zien is komen beide responsies exact overeen. Waaruit men kan besluiten dat de klasse een correcte werking heeft.



Figuur 31: Impuls responsie



Figuur 32: Stap responsie

#### 4.4. Testen bij cardiale revalidatie

Bij de eerste testen op één van de hartpatiënten bleek al snel dat het eerder gebruikte hartslag-detectiealgoritme geen robuuste manier was om aan detectie te doen. Zo moest de grenswaarde voor iedere patiënt apart ingesteld worden. Dit omdat de gemeten elektrische activiteiten verschillen van persoon tot persoon. Daarenboven werden sommige pieken onterecht gezien als R-toppen met als gevolg dat de berekende hartslag niet correct was. Deze onterechte interpretatie van de R-toppen was te wijten aan de grote aanwezigheid van storingen veroorzaakt door de vele bewegingen.

Hierdoor is er gezocht naar andere oplossingen en kwam het Pan-Tompkins algoritme naar de voorgrond als ideaal alternatief. Tevens bleek uit de testen dat het een goed idee is om de detectie van de QRS-complexen in reële tijd weer te geven. Dit gebeurt door een verticale rode lijn te tekenen bij het ECG-signaal.

Deze waarnemingen hebben bijgedragen tot het uiteindelijke controlepaneel zoals te zien is in Figuur 30.

Nog een belangrijke opmerking is dat de plaatsing van de elektroden een groot effect heeft op het gemeten signaal. Daarbij komt dat de beste plaats om te meten op het lichaam verschilt van persoon tot persoon. Daarom is elke patiënt van ons team apart gemeten om zo een maximale amplitude te vinden. Dit wordt gedaan door de maximale vector uitwijking te zoeken.



## 4.5. Validatie van applicatie

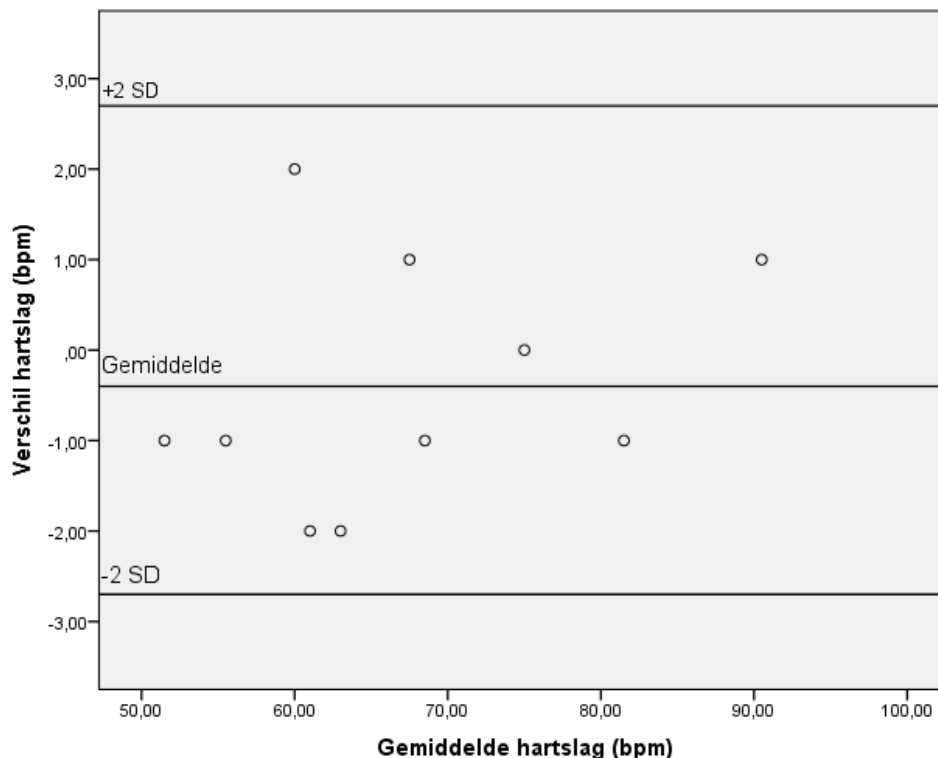
Om te controleren of de ontwikkelde applicatie werkelijk zijn hartritmes correct berekent. Zijn er een aantal meetwaarden vergeleken met die van een medisch gevalideerd toestel.

Hiervoor zijn er tien gezonde vrijwilligers gemeten in een rusttoestand waarbij ze plaatsnamen op een stoel. De metingen zijn simultaan uitgevoerd m.b.v. de twee toestellen zodat de hartritmes in principe hetzelfde moeten zijn.

Door deze metingen weer te geven in een Bland-Altman plot kan men dit verifiëren. Op de x-as komt het gemiddelde van twee overeenkomstige metingen te staan. Op de y-as komt het verschil van twee overeenkomstige metingen te staan.

Voor het betrouwbaarheidsinterval wordt twee maal de standaarddeviatie genomen. Dit omdat de metingen niet normaal verdeeld bleken te zijn. In Figuur 34 is te zien dat alle meetwaarden tussen dit interval vallen. Hieruit besluit men dat de applicatie correcte meetresultaten geeft. De metingen hebben wel een negatieve bias van -0,4 t.o.v. van het gevalideerd meettoestel. Wat er op wijst dat de berekende hartritmes van de applicatie iets lager liggen. Maar deze factor is praktisch verwaarloosbaar.

De twee metingen hebben een correlatie van 0,985 met een betrouwbaarheidsinterval van 99 procent.



Figuur 33: Bland-Altman tussen gevalideerd toestel en applicatie

#### 4.6. Test-loop en “Genk Loopt”

Na intensief testen van de applicatie is er besloten om een test-run te organiseren voor de pers. Deze test heeft gediend als generale repetitie voor de grote dag op 4 mei. De pers was massaal aanwezig. Zo werd het project vermeld in twee journaals en bij twee kranten stond het op de voorpagina.

Na deze test-loop zijn er nog een aantal kleine aanpassingen aan de applicatie gebeurd. Zoals het toevoegen van een verticale rode lijn bij het detecteren van een QRS-complex bijvoorbeeld. Na deze aanpassingen is de applicatie klaar bevonden om een zestal hartpatiënten in reële tijd te monitoren tijdens “Genk loopt”. Dit was een groot succes en alles is zonder problemen verlopen. Hieronder een aantal foto's.



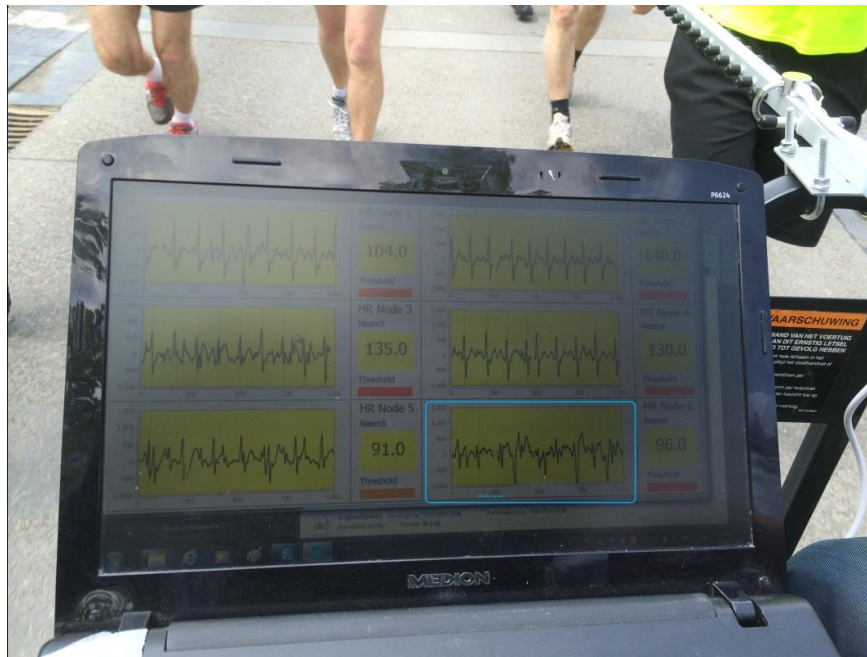
Figuur 34: Monitoring van het team van patiënten/begeleiders tijdens “Genk Loopt”



Figuur 35: Golfkar met patiënten erachter

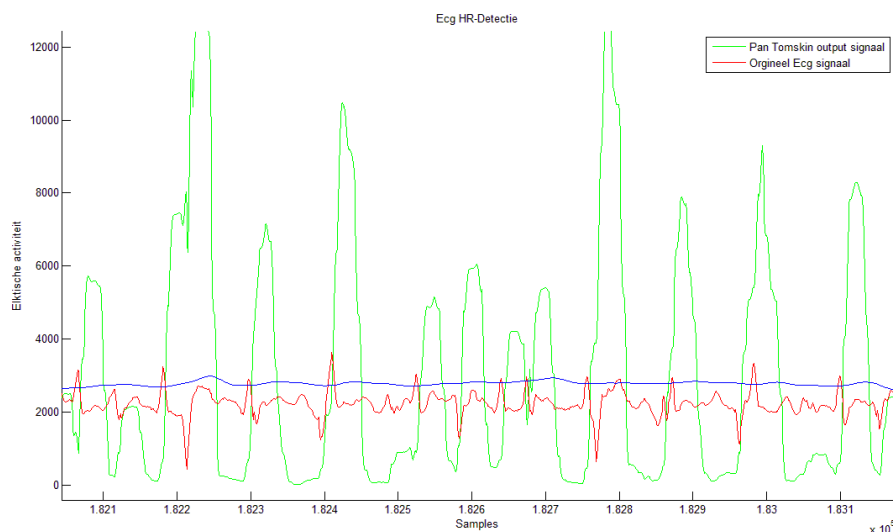
De ECG-signalen tijdens de reële tijdsmeting van “Genk loopt” waren over het algemeen goed. Bij één patiënt was het echter moeilijk om zijn ECG-signaal tijdens het lopen goed te interpreteren. Dit kan te wijten zijn aan het feit dat de elektroden niet op de juiste plaats hingen waardoor er geen maximale amplitude werd gemeten, of doordat de loopbeweging te veel impedantieverandering rondom de elektroden veroorzaakte. Tot deze conclusie komt men doordat de signalen van de patiënten met een lage BMI over het algemeen beter waren i.v.m. de patiënten met een hoge BMI.

In Figuur 36 is de monitorapplicatie te zien tijdens het loopevenement. In het blauwe kader staat het signaal dat moeilijker te interpreteren was.



**Figuur 36: Monitorapplicatie tijdens Genk loopt in actie**

Om de werking van het Pan-Tompkins algoritme op het slecht ECG-signaal te verifiëren, zijn na “Genk loopt” de gegevens met MATLAB gecontroleerd. Zoals te zien in Figuur 37 slaagt het algoritme er nog steeds in om de QRS-complexen te detecteren. Het algoritme krijgt het wel steeds moeilijker.



**Figuur 37: Pan-Tompkins van het slecht ECG-signaal tijdens "Genk Loopt"**

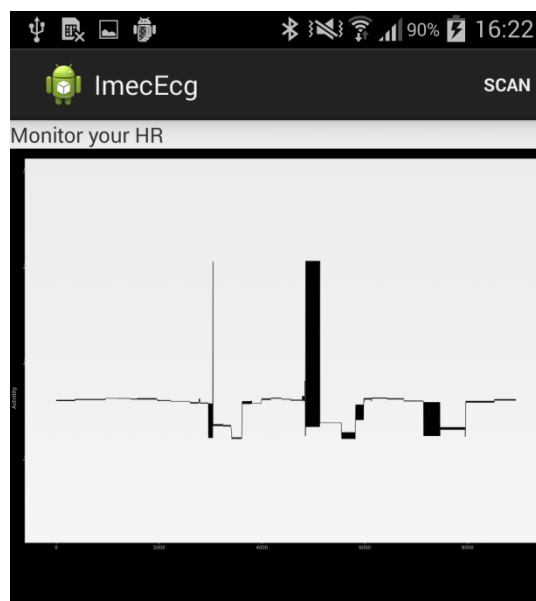
## 4.7. Uitbreiding a.d.h.v. Android App

Als uitbreiding op de hierboven beschreven applicatie, is er gekozen om een Android App te ontwikkelen. Het uiteindelijke doel van deze App is om patiënten hun ECG-signaal op eender welk moment te kunnen monitoren.

De smartphone neemt de taak van de gegevensverwerking op zich. Hierdoor kan de patiënt op zijn mobiele telefoon in reële tijd zijn eigen ECG-signaal volgen. Daarnaast gaat er ook een hartslagindicatie zijn en worden alle gegevens lokaal opgeslagen. In een latere fase is het de bedoeling dat de gegevens in reële tijd naar een server worden gestuurd. Aan deze server zou dan een monitorapplicatie kunnen hangen. Waardoor de cardioloog van op eender welke locatie zijn patiënten kan volgen.

### 4.7.1. Ontvangstprobleem

Bij het begin van de ontwikkeling waren er problemen met de weergave van de ECG-gegevens. Het weergegeven signaal zag er uit zoals te zien is in Figuur 38. Men kan zien dat er een ECG-vorm in het signaal zit maar dat het signaal niet helemaal correct is.



Figuur 38: Problemen weergave ECG-signaal

Na het nodige zoekwerk, bleek het probleem te liggen bij het feit dat de imec sensor een aantal keer dezelfde gegevens achter elkaar doorstuurt. Dit gebeurt niet systematisch maar eerder op een willekeurige manier. Om dit probleem op te lossen is er aan de ontvangtzijde een controlemechanisme geprogrammeerd. Die controleert of dat de ontvangen gegevens niet overeenkomen met al eerder ontvangen gegevens. Enkel wanneer de gegevens verschillen worden ze weergegeven. Hiermee is het probleem opgelost.

## 4.7.2. Algemene werking van de EcgMonitor App

Om connectie te maken met het ECG-toestel is er een “scan” knop voorzien. Indien men op deze knop drukt zal de applicatie op zoek gaan naar toestellen die werken op BLE. Wanneer er een apparaat is gevonden, wordt die toegevoegd aan een menu. In dit menu kan het gevonden toestel geselecteerd worden, waarna de initialisatie en connectie gestart wordt. In Figuur 39 staan deze functionaliteiten weergegeven.

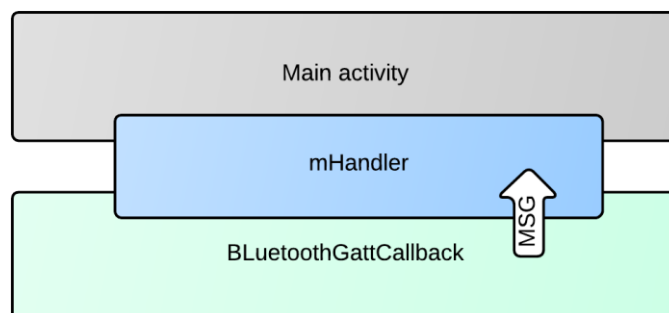


Figuur 39: Functies EcgMonitor APP

In Figuur 40 is de algemene werking van de applicatie te zien. Na de initialisatie en de connectie met het ECG-toestel wordt de *BluetoothGattCallback* opgeroepen telkens wanneer er gegevens via Bluetooth ontvangen zijn. Deze *callback* gaat kijken naar het gegevenstype en afhankelijk daarvan een bericht sturen naar de *mHandler*.

Het bericht bevat het gegevenstype en de ontvangen *characteristic*. De *mHandler* verzorgt de communicatie tussen *bluetoothGattCallback* en *mainActivity*. Dit is nodig omdat de *callback* een achtergrondthread is en deze kan niet rechtstreeks communiceren met de hoofdthread.(*mainActivity*).

De *mHandler* zal bepaalde methodes aanspreken zodat de *mainActivity* de nodige taken kan uitvoeren. De *mainActivity* heeft volgende taken: het verwerken van de ontvangen gegevens, weergave van ontvangen gegevens en de ontvangen gegevens opslaan in een bestand.

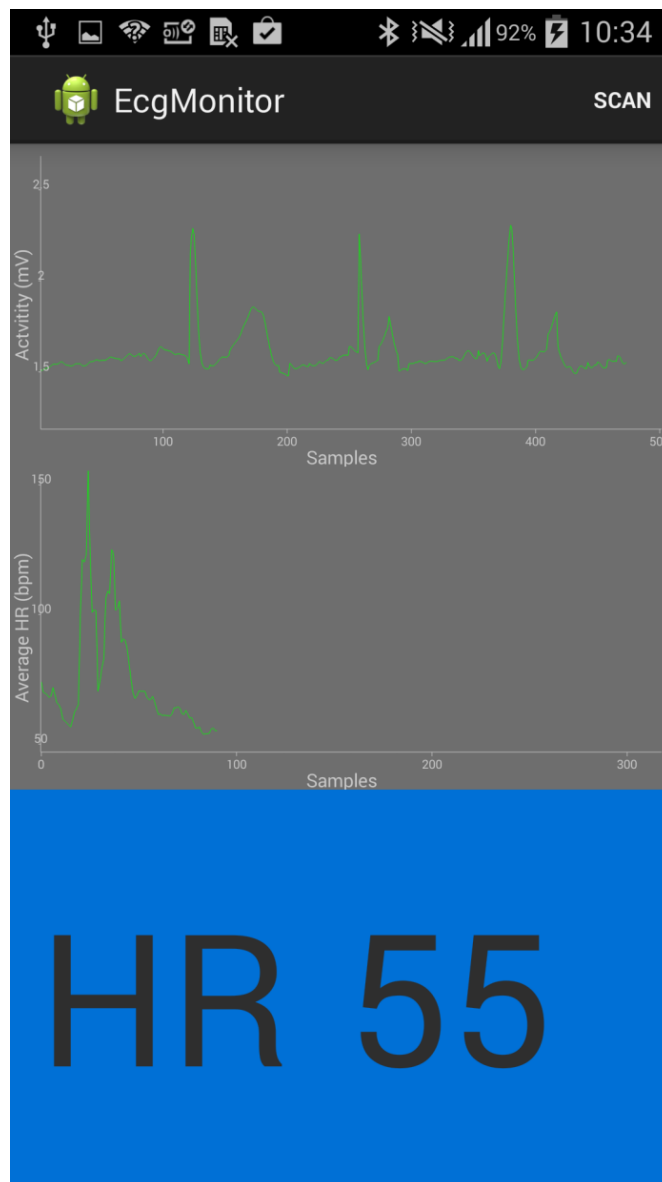


Figuur 40: Algemene werking van BLE

### 4.7.3. EcgMonitor Applicatie

Als resultaat is er een App die in staat is om de ECG-gegevens in reële tijd in te lezen en weer te geven in een grafiek. De sensor stuurt ook het gedetecteerde hartritme door, dit ritme wordt onder de grafieken gezet. Daarbij komend is er een grafiek waarin het hartritmehistoriek komt te staan.

De gegevens van het ECG-sigitaal en het hartritme komen in een CSV-bestand terecht. Waardoor offline analyse mogelijk is. In Figuur 41 staat een momentopname van de applicatie. Bovenaan staat het ECG-sigitaal. Daaronder staat een grafiek met de hartslaghistoriek en helemaal onderaan staat het momentele hartritme.



Figuur 41: EcgMonitor App in actie

## 5. Conclusie

### 5.1. Applicatie

Deze thesis heeft ertoe geleid dat er een werkende ECG-monitorapplicatie is, die in staat is om zes ECG-signalen in reële tijd te visualiseren. Het monitorscherm geeft de hartritmes, grenswaarden indicatie en namen weer.

De applicatie maakt hiervoor gebruik van zes ECG-noden, één basestation, één PC en één Java-programma. De gegevens worden in reële tijd verwerkt zodat de signalen zo snel mogelijk zichtbaar zijn in de monitorapplicatie. Tevens worden alle verwerkte gegevens weggeschreven in CSV-bestanden. Die kunnen later voor offline analyse van pas kunnen komen.

Naar analogie van het eerste monitorprogramma is er een overstap gemaakt naar een Android App. Dit met zicht op de toekomst. De App is nog niet volledig af, maar het begin is wel gemaakt. Het is mogelijk om met een smartphone een ECG-signaal in reële tijd te ontvangen en weer te geven. Bovendien is ook hier een hartslagindicatie. Alle ontvangen gegevens worden direct op geslagen in CSV-bestanden.

### 5.2. Werking

Op basis van de testen bij de cardiale revalidatie, de test-loop, "Genk loopt", en de validatie. Is gebleken dat de applicatie goed werkt en dat het zeker een handig hulpmiddel kan en zal zijn in het revalidatieproces van hartpatiënten. De kwaliteit van het gemeten signaal is echter wel sterk persoonsafhankelijk. Dit maakt het ooit moeilijk om toch een goed signaal op de monitorapplicatie te krijgen.

Doordat de metingen gebeuren in een sportomgeving, werkte het hartslag-detectiealgoritme op het principe van lokaal maximum niet goed. Het hartslag-detectiealgoritme van Pan-Tompkins gaf een veel beter resultaat en is daarom gebruikt in de uiteindelijke applicatie.

De geïmplementeerde filters werken goed in een sportomgeving. Dit is aangetoond bij het bespreken van de filters en het is gebleken bij het uitvoeren van de testen. Verder onderzoek naar adaptieve filters kan mogelijk het signaal nog verbeteren.

Uit de testen is tevens gebleken dat het ontvangstbereik van de antenne 25 meter is en dit over een hoek van 40 graden. Men zou kunnen kijken naar andere communicatiemogelijkheden. Denk maar aan de mogelijkheid om met een smartphone de gegevens van de ECG-node lokaal te verwerken en dan via 3G of 4G door te sturen naar een server. Aan die server kan dan een monitorapplicatie gekoppeld worden. De start voor deze mobiele applicatie is al gegeven, deze dient enkel nog verder uitgebreid en gevalideerd te worden.

In de toekomst is uitbreiding van deze thesis mogelijk door bijvoorbeeld een monitorsysteem voor meer dan duizend patiënten te ontwikkelen. Dit systeem moet dan beschikken over de nodige intelligentie om acute gevaren of problemen te detecteren en die door te geven aan een cardioloog.





## 6. Literatuurlijst

- [1] Hank Zumbahlen, *Linear Circuit Design Handbook*.: Newnes, 2008, p. 960.
- [2] Nathan M Kesto, "Electrocardiography Circuit Design," 2013.
- [3] APM Gorgels, "Elektrocardiografie," in *Cardiologie*.: Springer, 2007, pp. 105-118.
- [4] Pau Johnston Shen Luo, "A review of electrocardiogram filtering," *Journal of Electrocardiology*, pp. 486-496, 2010.
- [5] Hayes Monson H., "Fourier Analysis," in *Digital Signal Processing*.: McGraw-Hill, 1999, pp. 55-99.
- [6] Hayes Monson H., "Transform Analysis of Systems," in *Digital Signal Processing*.: McGraw-Hill, 1999, pp. 183-188.
- [7] Hayes Monson H., "The Z Transform," in *Digital Signal Processing*.: McGraw-Hill, 1999, pp. 142-146.
- [8] Peter Selvey. medteq. [Online]. <http://www.medteq.info/med/Articles>
- [9] Valtino X Afonso, "ECG QRS Detection," *Biomedical digital signal processing*, pp. 237-264, 1993.
- [10] Torfs T, Merken P, Penders J, Leonov V, Puers R, et al. Yazicioglu RF, "Ultra-low-power biopotential interfaces and their applications in wearable and implantable systems.," *Microelectronics Journal*, 2009 Aug. 27.
- [11] Dave Smith. (2013, Dec.) doubleencore. [Online]. <http://www.doubleencore.com/2013/12/bluetooth-smart-for-android/>
- [12] Robert Plonsey Jaakko Malmivuo, "Bioelectromagnetism," in *Bioelectromagnetism*. New York: Oxford University Press, 1995. [Online]. <http://www.bem.fi/book/15/15.htm>
- [13] Anthony Atkielski. (2007, January) <http://commons.wikimedia.org/>. [Online]. <http://commons.wikimedia.org/wiki/File%3ASinusRhythmLabels.svg>
- [14] Boston Scientific Corporation. (2006) bostonscientific. [Online]. <http://www.bostonscientific.com/lifebeat-online/heart-smart/bradycardia.html>

## Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

**Ontwikkeling van een klinisch dashboard voor continue en real-time monitoring gedurende sportactiviteiten: "Big Mother is watching you"**

Richting: **master in de industriële wetenschappen: elektronica-ICT**

Jaar: **2014**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

**Simons, Jordie**

Datum: **5/06/2014**