

Technische Universität Ilmenau
Fakultät IA
Fachgebiet Biosignalverarbeitung

Praktikum EKG-Signalanalyse
WS 2021/22

Versuchsprotokoll: EKG-Signalanalyse

Praktikumsbetreuer: Marc Heppner

Teilnehmer: Robert Jeutter & Leonard Seifert

Datum Versuchsdurchführung: 14.01.22

Datum Protokollabgabe: 28.01.22

Erklärung:

„Hiermit versichern wir, dass wir dieses Praktikum selbständig vorbereitet, mit dem Praktikumsleiter durchgeführt und selbständig nachbereitet haben und nur die angegebenen Quellen und Hilfsmittel verwendet haben.“

1 Vorbereitungsaufgaben

1.1 EKG-Vorverarbeitung

Entwicklung einer Strategie zur EKG-Vorverarbeitung. Bedenke dabei, dass die EKG-Vorverarbeitung maßgeblichen Einfluss auf die Qualität der QRS-Detektion hat.

Bei der EKG Signaldetektion enthält das auszuwertende Signal neben dem gewünschten EKG verschiedene Störsignale. Diese Störsignale können in zwei Kategorien eingeteilt werden: stochastische und deterministische Signale. In der Vorverarbeitung ist es unser Ziel die deterministischen Störungen zu entfernen.

Eines der bekannten niedrigfrequenten Signale entsteht durch die Atmung und bildet sich als Drift in dem Signal ab. Dieser Fehler kann mit einem Hochpass behoben werden, der die niederfrequenten Signale entfernt.

Desweiteren nutzen wir den bekannten Frequenzbereich in dem der größte Energieanteil des QRS Komplexes liegt zwischen 5-26 Hz mithilfe eines Bandpasses, um die Störsignale außerhalb dieses Bereiches im Vergleich zu unserem gewünschten Signal abzuschwächen.

Das Ergebnis der Vorverarbeitung soll ein driftarmes und von größeren deterministischen Störsignalen befreites Signal liefern.

1.2 QRS-Detektion

Entwicklung eines Algorithmus zur adaptiven QRS-Detektion. 250 ms Zeitvorlauf sollen dabei nicht überschritten werden

1. Hochpass zur Driftkompensation
 - a) Fourier Transformation
 - b) entfernen niedriger Frequenzen
 - c) Rücktransformation mit Inverser Fouriertransformation
2. Bandpass zur Beschränkung auf niederfrequente Biosignale zwischen 5-26 Hz
3. Minimale Distanz zwischen Peaks der R-Spitzen ermitteln

- a) Kopie des Signal erstellen zur verlustfreien Bearbeitung
 - b) Skalierung des Signals
 - c) Kleine Peaks entfernen bzw. auf null setzen
 - d) alle übrigen Peaks finden
 - e) minimalen Abstand aller (übrigen) Peaks ermitteln
4. Threshold festlegen (Halbe Signalstärke)
 5. Signalspitzen zwischen minimaler und maximaler Distanz, die über dem Threshold liegen, ermitteln

Entwickle zu diesem Algorithmus die zugehörige(n) MATLAB-Funktion(en)

```
function [R_Positionen, Entscheidungssignal, Schwellwertverlauf, Lernphase]
= QRS_Detektion (EKG_Signal, fa);
    size = length(EKG_Signal); % Laenge des Signals
    % fourier transformation
    signal = fft(EKG_Signal);
    % entferne niedriger frequenzen anhand Laenge und Rate
    signal(1 : round(size*5/fa))=0;
    signal(end - round(size*5/fa) : end)=0;
    % inverse fourier transformation
    signal=real(iff(signal));
    % bandpass fuer 5 bis 26 hz
    signal=bandpass(signal, [5 26], fa);

    % skalierung des Signals auf skala 1-10
    filtersignal=signal/(max(signal)/10);
    % entfernen aller peaks unterhalb des thresholds 10/2=5
    for data = 1:length(filtersignal)
        if filtersignal(data) < 5
            filtersignal(data) = 0;
        else
            filtersignal(data)=1;
        end
    end
    % finde alle uebrigen peaks, die ueber dem threshold lagen
    positions=find(filtersignal);
    % abstand der ersten beiden peaks
    distance=positions(2)-positions(1);
    % setze den abstand auf das minimum zweier peaks in der reihe aller uebrigen peaks
    for data=1:length(positions)-1
        if positions(data+1)-positions(data)<distance
            distance=positions(data+1)-positions(data);
        end
    end

    % Mittel des Signals als Threshold
    avg = mean(signal)
```

```
% Absolute Peaks innerhalb des Zeitfensters finden
[RPeaks, positions] = findpeaks(signal, 'MinPeakHeight', 1.5*avg,
                                'MinPeakDistance', distance, "MaxPeakWidth", 250);
R_Positionen = positions;
Entscheidungssignal = signal;
Schwellwertverlauf = RPeaks;
Lernphase = 0; % Bedeutung nicht erklärt
end
```

Dieser Code ist derzeit nicht online fähig und passt sich keinen Signalstärkenänderungen an.

2 Praktikumsauswertung

2.1 EKG-Ableitung mit Hilfe des Biosignalverstärkers

Leiten Sie jeweils ein 5 minütiges EKG eines Studenten innerhalb der folgenden vier Phasen ab. Achten Sie dabei auf die Auswahl der Kanäle!

- Lagetyp-Phase: Proband liegt und atmet normal
- Ruhe-Phase: Proband liegt und atmet normal
- RESP-Phase: Proband liegt und atmet langsam tief ein und tief aus
- STEH-Phase: Proband steht und atmet normal

Diese Aufgabe entfiel, da das Praktikum über Webex stattfand. Die EKG Signale für weitere Aufgaben waren bereits in der Simulation vom Fachgebiet vorbereitet.

2.2 Bestimmung der elektrischen Herzachse

Mit Hilfe des Matlab-GUI „Datenanzeigen“ können die abgeleiteten Signale dargestellt und ausgewertet werden.

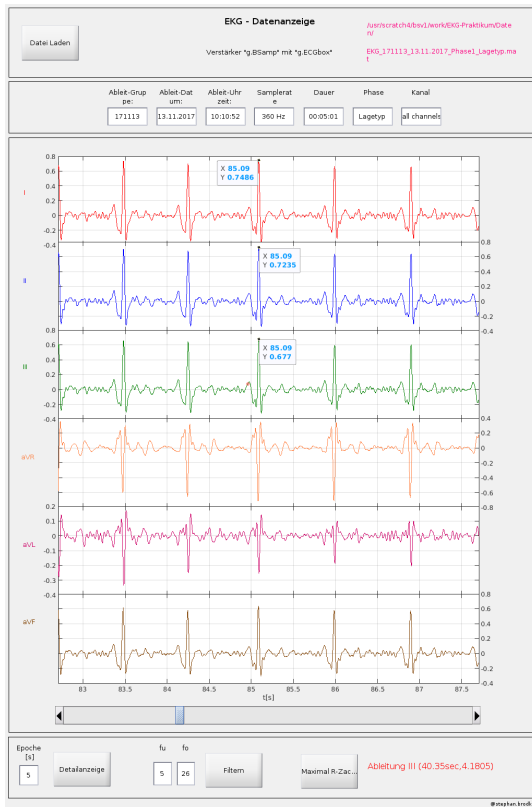


Abbildung 1: Phase1_Lagetypp.mat

Der vorgegebene Testdatensatz „Phase1_Lagetypp.mat“ wurde in das Programm geladen. Das Signal wurde in der GUI mit einer unteren Filtergrenze von 5 Hz und oberen Filtergrenze von 26 Hz gefiltert.

Für die Herzachsenbestimmung sollte kein Abschnitt zu Beginn des Messvorgangs verwendet werden, da dort oft Einschaltfehler der Messgeräte auftreten und der Patient unruhiger ist. Wir entschieden uns für die Epoche zwischen 83 und 87 Sekunden, da dieser Artefaktfrei ist (hier links in Abbildung 1 zu sehen).

Die höchste R-Zacke in dieser Epoche haben wir bei 85,09 Sekunden gemessen. Die Amplitude wurde mit den GUI „Data-Cursor“ ausgelesen. Diese sind für

- Kanal I: 0,7486 mV
- Kanal II: 0,7235 mV
- Kanal III: 0,677

Die Funktionswerte werden in den, aus der Praktikumsanleitung vorgefertigten, Cabrera-Kreis eingetragen um den Lagetypp zu bestimmen. Zur Umrechnung von mV in cm wenden wir den Maßstab [1mV:2cm] an. Aus der Vektoraddition der Schnittpunkte zweier Kanäle lässt sich die Herzachse in Grad messen. Die Lagetyppbestimmung ist in Abbildung 2 zu sehen.

Da wir mit einem anonymen Datensatz ohne bekanntes Alter des Probanden arbeiten, rechnen wir am ehesten mit dem durchschnittlichen Normaltyp. Diesen erhalten wir, eine weitere Diskussion entfällt dadurch.

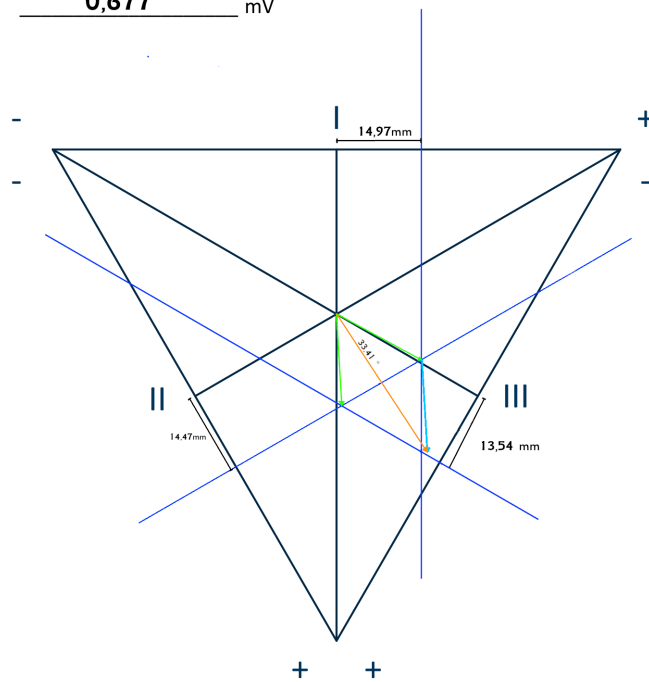
Abbildung 2: Lagetypbestimmung mit Cabrera-Kreis

Zeitpunkt: 85,09 Sekunden

Amplitude I 0,7486 mV

Amplitude II 0,7235 mV

Amplitude III 0,677 mV



Lagetyp: Normaltyp (30°-60°)

2.3 EKG-Vorverarbeitung

Überprüfe die Wirksamkeit der vorgeschlagenen Vorverarbeitungsmethoden und deren Parametereinstellungen anhand der EKG-Signale der MIT-Datenbank. Überlagere die Signale mit einem Drift und einer 50 Hz-Sinusschwingung!

In der Ansicht zur EKG Vorverarbeitung zeigt die GUI vier Kanäle. Kanal „Rohsignal“ zeigt das Eingangssignal ohne Fehler. Kanal 1 zeigt das Rohsignal nach Überlagerung mit einem Drift und einer 50Hz Störungs-Schwingung. Kanäle 2 und 3 werden für die Vorverarbeitungsansicht verwendet, durch Anwenden von Filtern können die Signalausgaben hier gezeigt werden.

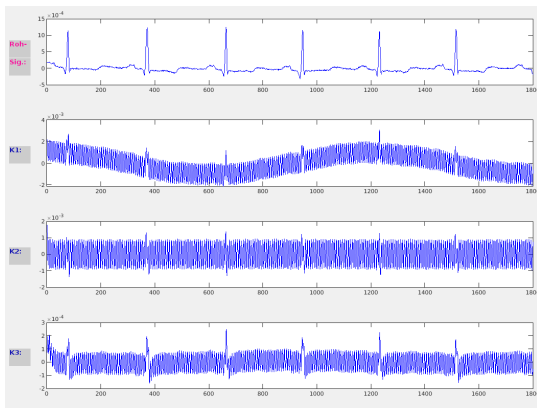


Abbildung 3: DB100 V1

Mit Datenbank 100 wird das EKG Signal geladen und in Kanal „Rohsignal“ dargestellt. Für die erste Vorverarbeitung wurde ein Hochpass 1. Ordnung mit $f_a = 360$ und $f_o = 25$ des Types IIR-Einseitig Butterworth angewendet und in Kanal „K2“ abgebildet. Dadurch wurde der Drift entfernt, Rauschen ist noch vorhanden, Signalspitzen wurden verzerrt. Kanal „K2“ wird weiter mit einem Tiefpass 1. Ordnung mit $f_u = 5$ des Types IIR-Einseitig Butterworth versehen und in Kanal „K3“ abgebildet. Zu sehen in Abbild 3.

Das Ergebnis ist ein nicht verwertbares Signal, da zu Beginn eine Verzerrung auftritt und die Signalspitzen verrauscht sind. Die R-Spitzen können aus diesem Endsignal nicht eindeutig ermittelt werden. Zur Korrektur soll der Filter von K2 auf K3 angepasst werden.

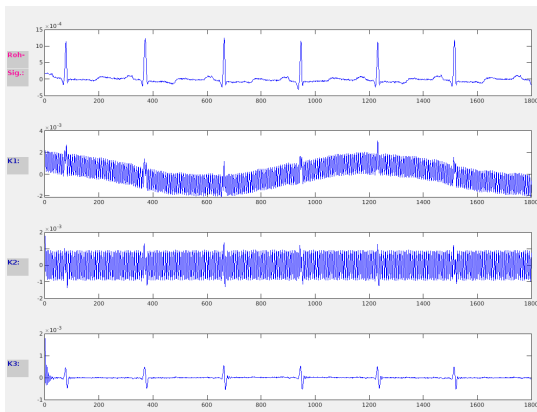


Abbildung 4: DB100 V2

Um das Rauschen ohne Signalverzerrung umzusetzen wird in Abbildung 4 von Kanal 2 auf Kanal 3 ein Notch Filter angewendet. Die Filterung von Kanal 1 auf Kanal 2 bleibt bestehen.

Der Notch Filter entfernt das 50Hz Rauschen zuverlässig. Die Verzerrung der Form des Signals blieb jedoch bestehen.

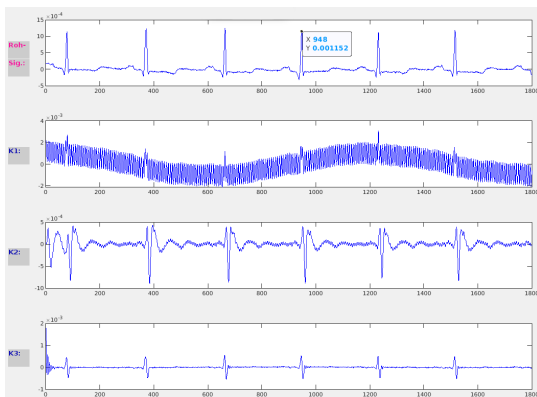


Abbildung 5: DB100 V3

In Abbild 5 ist Datenbank 100 in Kanal 2 mit einem IIR-Einseitig Bandpass 5. Ordnung des Types Butterworth mit $f_o = 5$ und $f_u = 25$ gefiltert. Kanal 3 wurde nicht verändert, das Ergebnis des letzten Notch Filters ist noch sichtbar.

Bereits aus Kanal 2 ist ersichtlich, dass das Signal nicht die gewünschten Eigenschaften zur QRS-Erkennung haben wird. Sowohl der Amplitudengang als auch der Phasengang sind gespiegelt.

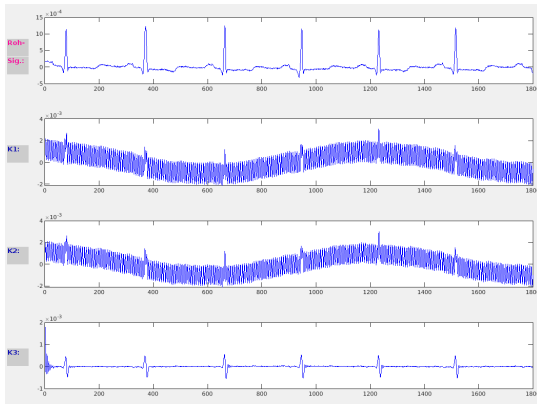


Abbildung 6: DB100 V4

In Abbild 6 ist ein Wechsel von IIR auf FIR Filtern vorgenommen worden. Aus Kanal 1 auf Kanal 2 wurde ein FIR Hochpass-Filter 1. Ordnung mit $f_u = 2$ angewendet. Kanal 3 ist weiterhin unverändert übernommen.

Der Drift wurde nicht entfernt und stört weiterhin die Erkennung. Im Experiment wurde gezeigt, dass auch höhere Ordnungen dieses Filters das Signal nicht verbessern. Im nächsten Schritt wird zu einem Bandpass übergegangen.

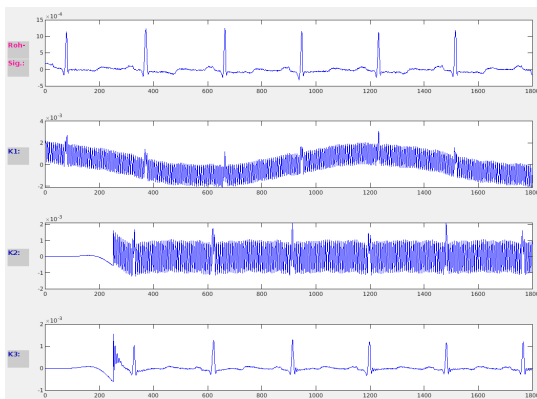


Abbildung 7: DB100 V5

Von Kanal 1 auf Kanal 2 wird ein einseitiger FIR-Bandpass mit $f_u = 2$ und $f_o = 30$ Hz 5. Ordnung eingesetzt. Von Kanal 2 auf Kanal 3 wird der Notch Filter erneut eingesetzt und in Abbildung 7 sichtbar.

Zu Beginn des Ausgangssignals kommt es zu niedrigen Werten. Das liegt daran, dass in jedem Arbeitsschnitt die letzten 250ms in dem FIR Filter zur Berechnung der Filterantwort herangezogen werden. Zusätzlich wurde das Signal um ungefähr 250ms verzögert.

Bei 250ms des Zeitverlaufs bezieht sich der Bandpass auf den ersten eingegangenen Wert des K1 Signals und führt eine Nullpunktanpassung aus. Dadurch entsteht eine Signalspitze die falsch als R-Spitze erkannt werden könnte.

Der Notch Filter entfernt das 50Hz Rauschen zuverlässig.

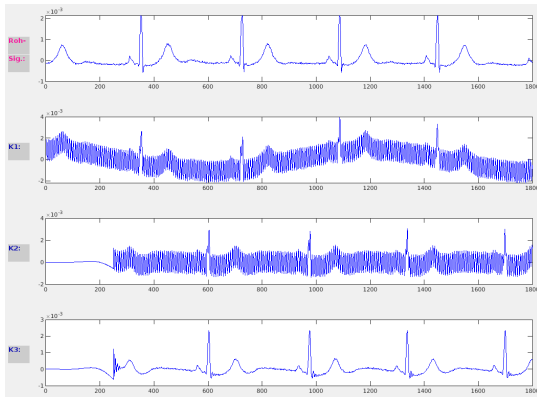


Abbildung 8: DB106

Derselbe FIR-Bandpass und Notch-Filter aus 7 wurde hier in Abbildung 8 verwendet und führt zu einem guten Ergebnis, aus dem die R-Spitzen klar ablesbar sind.

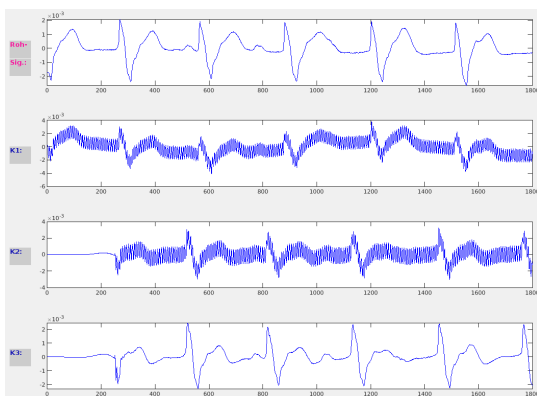


Abbildung 9: DB108

In Abbildung 9 wurde wieder der FIR-Bandpass und Notch Filter aus 7 verwendet. Das Ergebnis ist wieder gut und für eine Detektion nutzbar.

Die Versuche zeigen, dass Notch-Filter eine gute Eliminierung des 50 Hz-Netzausrauschens erreichen. IIR- und FIR-Filter können insbesondere zur Eliminierung von niederfrequenten Störungen durch den Drift verwendet werden. Die Realisierung mit FIR-Hochpässen ist aufwändig, da eine hohe Filterordnung für das enge Filtergrenze vonnöten ist.

Besonders in den Abbildungen 7, 8 und 9 ist auch ersichtlich, dass der FIR-Filter den Nachteil der längeren Laufzeit aufweist und damit das Signal verzögert. Dennoch liefert der FIR-Filter gute und verwertbare Ergebnisse und ist selbst durch seine Stabilität und Vorhersagbarkeit im Vergleich zum rückgekoppelten IIR-Filter von Vorteil. Aus den ersten Versuchen der Sektion 2.3 ist dabei auch zu berücksichtigen, dass IIR-Filter das Ausgangssignal im Gegensatz zu FIR-Filtern verzerren, was Signale so verändern kann, dass pathologische Fehler nicht mehr erkennbar sind.

2.4 QRS-Detektion mit Hilfe von MATLAB

Implementiere den entwickelten Algorithmus zur QRS-Detektion in die Funktion `QRS_Detektion.m`

Der Code der QRS Detektion aus den Vorbereitungsaufgaben wurde in das Programm eingefügt. Der vorgeschlagene Algorithmus verwendet einen Hochpass- und Bandpassfilter um danach mit einem aus dem Verlauf gemittelten Threshold die R-Spitzen zu erkennen.

Evaluieren Sie den QRS-Detektor anhand der EKG-Signale der MIT-Datenbank. Notieren Sie die Detektionsquote. Wo liegen die Stärken bzw. die Schwächen des Detektors? Versuchen Sie anhand der Ergebnisse dieses ersten Detektionstests den Detektionsalgorithmus bzw. (falls angebracht) auch die EKG-Vorverarbeitung zu optimieren! Wiederholen Sie den Detektionstest! Inwieweit konnten die Detektionseigenschaften verbessert werden?

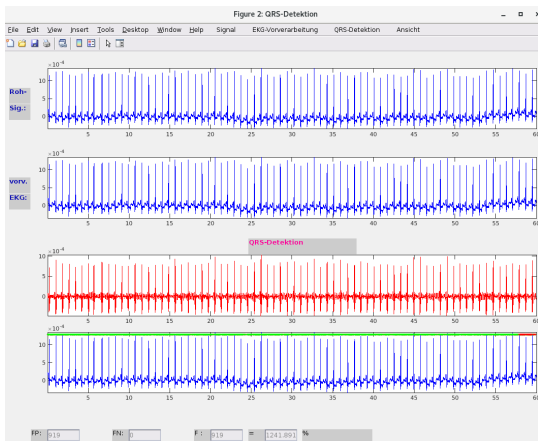


Abbildung 10: QRS DB100

Abbildung 10 zeigt den gesamten Verlauf des EKG Signals mit der QRS Detektion. Im unteren Bildteil wird die Fehlerzahl der Falsch-Positiven und Falsch-Negativen Detektionen und ein Prozentwert der Fehlerquote angezeigt.

Der Fehler in dieser Detektion ist sehr hoch. Viele Teile des Signals wurden fälschlich als R-Zacke erkannt, was auf einen zu niedrigen Threshold hinweist. Durch den niedrigen zeitlichen Anteil der R-Spitze im Vergleich zur Dauer des Restsignals liegt der Gesamtmittelwert sehr niedrig. Der Matlab Code musste deswegen angepasst werden.

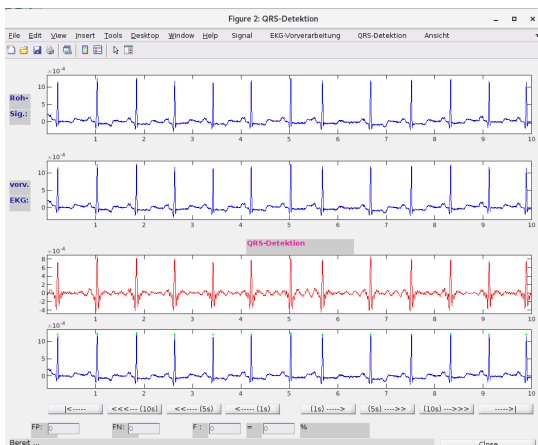


Abbildung 11: QRS DB100 V2

Der Treshold wird nicht mehr über den Mittelwert aller Punkte ermittelt. Stattdessen wird die maximale Amplitude bestimmt und halbiert. Dies dient dazu, nur Signale oberhalb der P-, T- und U-Wellen zu erkennen. Das sollte die Erkennung auf R-Spitzen beschränken.

Die Erkennung für Datensatz 100 in Abbildung 11 ist wesentlich besser geworden im Vergleich zu Abbildung 10.

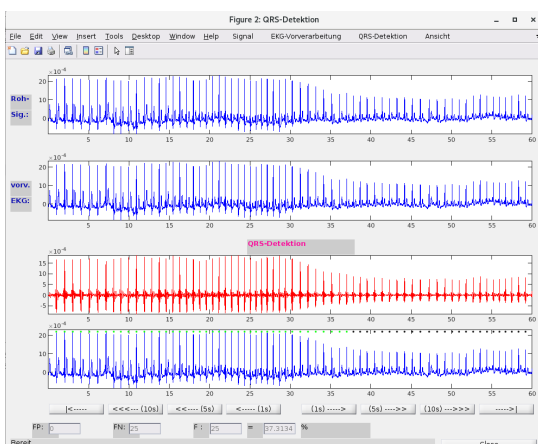


Abbildung 12: QRS DB106

In Abbild 12 sieht man die Detektion des Datensatz 106. Während die Erkennung zu Beginn gut ausfällt, fällt diese im letzten Drittel rapide herab. Das liegt an der Amplitudenänderung des Signals, an die sich der Matlab Algorithmus nicht anpasst. Stattdessen wird weiter mit dem globalen Threshold gearbeitet und R-Spitzen und diesem nicht mehr erkannt. Eine Verbesserung wäre durch eine aktive Schwellwertanpassung während des Verlaufs möglich (hier nicht implementiert).

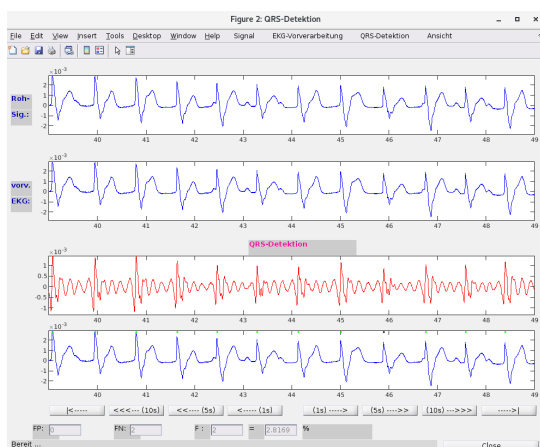


Abbildung 13: QRS DB108

Der Datensatz 108 in Abbildung 13 zeigt einen Verlauf mit erhöhter T-Welle als gewöhnlich, dies weist auf einen Herzinfarkt im Stadium 1 hin. Da diese T-Wellen relativ hoch ausschlagen werden diese falsch Positiv erkannt. Der Matlab Algorithmus wurde daraufhin angepasst, indem der Schwellwert von der Hälfte der Amplitude auf das obere fünfteil der Amplitude erhöht wurde. Daraufhin war die Detektion beinahe Fehlerfrei mit 2,8% Fehlerquote, wie in der Abbildung zu erkennen. Insgesamt wurden 2 R-Spitzen nicht erkannt. Es gab keine Falsch-Positiven Detektionen.

Der vorgeschlagene Matlab Algorithmus konnte durch Modifikation des Threshold verbessert werden aber passt sich nicht dynamisch verändernden Amplituden an.

Mögliche Adaptionen der QRS Erkennung zur verbesserten Detektion:

- adaptive Schwellwertanpassung, um R-Spitzen in einer dynamisch ändernden Amplitude weiterhin korrekt zu detektieren
- ein minimaler und maximaler Abstand zwischen R-Spitzen um andere erhöhte Wellen von der Detektion ausschließen zu können
- online Fähigkeit, um die Detektion auch ohne abgeschlossene Aufnahme betreiben zu können (Bsp. EKG Monitor in der Intensivstation)
 - der Threshold kann hierfür nicht aus dem Gesamtsignal ermittelt werden, stattdessen sollte ein Zeitfenster der vorhergehenden Sekunden betrachtet werden
- Schwellwert stark erhöhen direkt nach Erkennen einer R-Spitze und langsam senken. Dadurch können pathologische Veränderungen der S-, T- und U-Wellen weniger Einfluss auf die Detektion nehmen. Ein exponentiell abklingender Verlauf würde sich beispielsweise anbieten, da dieser sowohl bei sehr hohem als auch bei niedrigem Puls gute Ergebnisse zu erwarten hat.

2.5 Analyse der Herzfrequenzvariabilität

Analysiere die aufgezeichneten EKG-Signale während der drei Phasen RUHE, RESP und STEH. Interpretiere die Unterschiede in den Ergebnissen der einzelnen Phasen der HRV-Analyse

Die GUI zeigt in der Herzratenvariabilität (HRV) unterschiedliche Diagramme zur Analyse auf. Das Diagramm QRS-Detektion zeigt den Verlauf des verwendeten Datensatzes in blau mit erkannten R-Spitzen in grün. Das Diagramm „Herzfrequenz nach Interpolation“ darunter zeigt den zeitlichen Verlauf der Abstände zwischen zwei aufeinanderfolgenden R-Spitzen.

Das „HF Leistungsspektrum“ zeigt die Wahrscheinlichkeitsdichte unterschiedlicher auftretender Frequenzen über den gesamten Zeitverlauf. Durch Ausprägung markanter Formen können pathologische Veränderungen erkannt werden.

Das Scatter- und Phasendiagramm sind weitere Darstellungsformen für die Variabilität. Je weiter verstreut die Punkte, welche die aufeinanderfolgenden Abstände der R-Spitzen darstellen, liegen desto größer ist die HRV. Im gesunden Fall erwarten wir eine elliptische Form des Streudiagramms; andere Ausprägungen können beispielsweise auf Herzrhythmusstörungen hinweisen.

Das Histogramm ist das letzte Diagramm der GUI. Die gemessenen R-Spitzen-Abstände werden in feste Zeitbereiche gegliedert und als Balken über die prozentuale Häufigkeit des bestimmten Wertes dargestellt. Die Anzahl der Balken ist abhängig von der Variabilität des Probanden. Im gesunden Fall erwarten wir grob die Form einer Gaußkurve. Andere Formen können auf pathologische Veränderungen oder weitere äußere Einflüsse während der Messung (Bsp. starke Atmung) hinweisen.

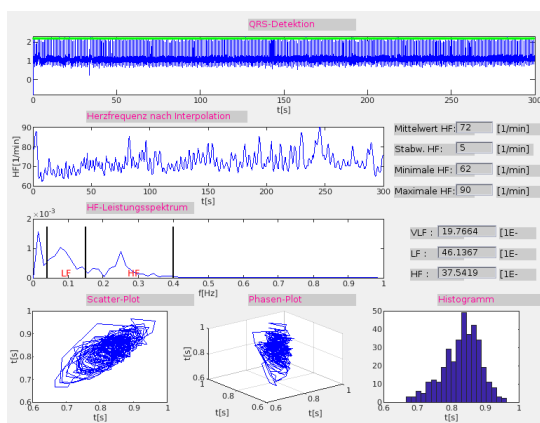


Abbildung 14: Phase RUHE

In der Ruhephase sitzt und atmet der Proband normal.

Das Diagramm nach Interpolation zeigt kleine Amplitudenänderungen bei großer Signalanzahl.

Im HF-Leistungsspektrum sind drei markante Spitzen ersichtlich. Diese entsprechen den in der Praktikumsanleitung (S.24) beschriebenen thermoregulatorischen Vorgängen im sehr niederfrequenten (VLF) Bereich, Blutdruckwellen III. Ordnung im niederfrequenten (LF) Bereich und Einflüsse der Atmung im hochfrequenten (HF) Bereich.

Der Scatter-Plot und Phasen-Plot zeigen eine gering gestreute Anordnung und eine etwa elliptische Form.

Im Histogramm erkennt man die erwartete gaußsche Glockenkurvenform.

Die Diagramme zeigen alle die erwarteten und in der Literatur beschriebenen Eigenschaften und weisen auf einen gesunden Probanden hin.

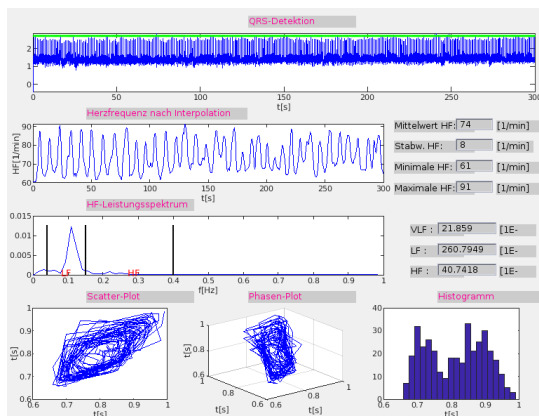


Abbildung 15: Phase RESP

In der Respirationsphase sitzt der Proband und atmet langsam tief ein und tief aus.

Das Diagramm nach Interpolation zeigt große Amplitudenänderungen. Im Vergleich zum vorhergehenden Datensatz bleibt die Amplitudendifferenz im zeitlichen Verlauf sehr konstant.

Im HF-Leistungsspektrum ist eine markante Spitzen im niederfrequenten (LF) Bereich zu sehen. Diese entspricht der erwarteten Ausprägung durch starke Atmung. Jedoch aufgrund der langsamen Atmung in den niederfrequenten (LF) Bereich verschoben.

Der Scatter-Plot und Phasen-Plot zeigen eine elliptische Ausprägung bei breitere Streuung der Datenpunkte.

Im Histogramm erkennt man zwei Ausprägungen der erwarteten Gaußkurven, die einander teilweise überlagern; entsprechend der Ein- und Ausatemphasen.

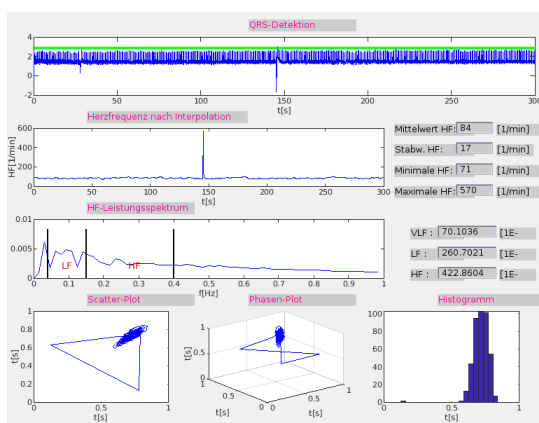


Abbildung 16: Phase STEH

In der Stehphase steht der Proband und atmet normal.

Aus der QRS-Detektion und den Scatter- und Phasenplot ist ein, wahrscheinlich aus einem Bewegungsartefakt resultierender, Ausreißer ersichtlich. Dieser muss hier einmalig manuell im Code behoben werden, indem der Datenpunkt genullt wird. Die korrigierte Detektion ist in Abbildung 17 sichtbar und zeigt deutlich, wie ein einzelner Ausreißer die Auswertung verfälschen kann.

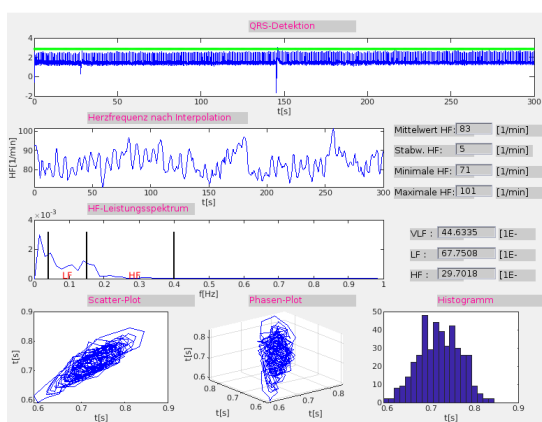


Abbildung 17: Phase STEH Korrigiert

Das Diagramm nach Interpolation zeigt eine hohe Variabilität im Zeitverlauf.

Im HF-Leistungsspektrum sind eine Spitze im sehr niederfrequenten (VLF) Bereich erkennbar, sowie eine breite Signaldichte im niederfrequenten (LF) Bereich. Der hochfrequente (HF) Bereich ist weitestgehend frei von Signalen. Im Vergleich zur Ruhephase sehen wir einen höheren Einfluss der sehr niederfrequenten Anteile (z.B. Blutdruck, thermoregulatorische Vorgänge) auf das Gesamtspektrum.

Der Scatter-Plot und Phasen-Plot zeigen eine in die Länge gezogene Ellipse mit hoher Dichte.

Im Histogramm erkennt man eine Gaußkurve mit wenigen Spitzen in der Mitte. Im Vergleich zur Ruhephase ist die Gaußkurve nach links verschoben, was von der höheren Herzfrequenz herrührt.

Vergleiche die HRV-Ergebnisse des Praktikumsprobanden mit den Ergebnissen eines Polyneuropathie-Patienten.

Polyneuropathie ist eine Gruppe von Erkrankungen, bei der mehrere periphere Nerven geschädigt sind. Bekannte Symptome sind unter anderem Muskelschwäche, Muskelkrämpfe und Lähmungen.

Zu beachten ist die verkürzte Laufzeit des Datensatzes in 18 von 180 Sekunden im Vergleich zu den Standard 300 Sekunden Aufnahmen der Datensätze 14, 15 und 17. Außerdem ist zu beachten, dass die Amplituden der QRS-Detektion sich sehr stark von den Erwartungen am gesunden Probanden unterscheiden, ob es sich hierbei um eine andere Größenordnung als mV handelt ist aus dem Diagramm nicht ersichtlich.

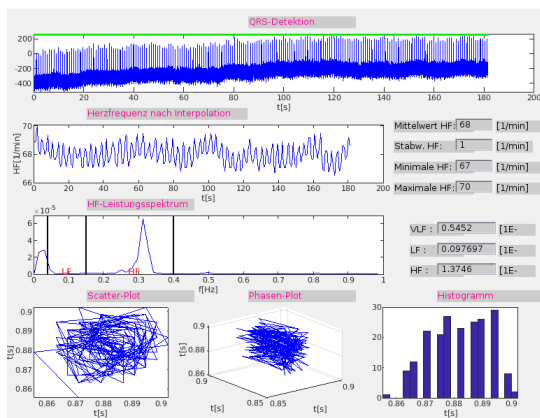


Abbildung 18: Polyneuropathie

Das Diagramm nach Interpolation zeigt eine gleichmäßige Herzrate mit geringer Variabilität.

Im HF-Leistungsspektrum sind eine Spitze im sehr niederfrequenten (VLF) Bereich und eine im hochfrequenten (HF) Bereich abzulesen. Dies unterscheidet sich deutlich von gesunden Probanden.

Im Phasen-Plot erkennt man eine geringe Verteilung, die auf eine geringe Variabilität zurückzuführen ist. Die Form des Scatter- und Phasen-Plot weist zudem keine Ellipse auf was auf eine mögliche Herzrhythmusstörungen hindeutet.

Im Histogramm ist keine Gauß Verteilung sichtbar. Deutlich erkennbar sind hier vor allem Lücken. Auch dies weist auf eine Herzrhythmusstörung hin.

Ein Vergleich mit den HRV-Analysen in Abbildungen 14, 15 und 17 zeigt deutlich erkennbare Unterschiede auf, die für eine Diagnose des Probanden genutzt werden können. Alle Diagramme in Abbildung 18 zeigen eine sehr kleine Verteilung und Herzfrequenzvariabilität. Die Herzrate ist sehr gleichmäßig. Über die Verteilung im Phasen-Diagramm sind sehr enge R-Spitzen-Abstände zu sehen. Diese deuten ohne Vorwissen der Erkrankung auf ein Problem mit dem Nervensystem hin. Die Ausformung des Histogramms und Scatter-Diagramms weisen zusätzlich auf eine mögliche Herzrhythmusstörungen hin. Nach den Spitzen des HF-Leistungsspektrums wurde das EKG vermutlich während der Respirationsphase aufgenommen.

3 Quellen

- Prof. Dr.-Ing. habil. Peter Husar, Vorlesung „Grundlagen der Biosignalverarbeitung“, TU Ilmenau, 2021
- M.Sc. Marc-Patrick Heppner, Praktikumsanleitung „EKG Signalanalyse“, TU Ilmenau, 2021
- Univ.-Prof. Dr.-Ing. Martin Haardt, Vorlesung „Signale und Systeme 1“, TU Ilmenau, 2019
- eleceng.dit.ie, Matlab Beispiele, Abgerufen 11.01.2021
- biomedicalsignalandimage.blogspot.com, Matlab EKG Beispiel, Abgerufen 11.01.2021
- GitHub.com/Aburas98, Matlab EKG Beispiele, Abgerufen 04.01.2021
- section.io, QRS Detektion Beispiel, Abgerufen 11.01.2021
- Mathworks.com, EKG Analyse, Abgerufen 04.01.2021
- Netdoktor.de, Polyneuropathie, Abgerufen 25.01.2021
- Karl Dirk Kammeyer: Digitale Signalverarbeitung. 6. Auflage. Teubner, 2006, ISBN 3-8351-0072-6

Abbildungsverzeichnis

1	Phase1_Lagetyp.mat	6
2	Lagetypbestimmung mit Cabrera-Kreis	7
3	DB100 V1	8
4	DB100 V2	9
5	DB100 V3	9
6	DB100 V4	10
7	DB100 V5	10
8	DB106	11
9	DB108	11
10	QRS DB100	13
11	QRS DB100 V2	13
12	QRS DB106	13

13	QRS DB108	14
14	Phase RUHE	16
15	Phase RESP	17
16	Phase STEH	17
17	Phase STEH Korrigiert	18
18	Polyneuropathie	19

4 Anhang: Überarbeiteter Code

```

function [R_Positionen, Entscheidungssignal, Schwellwertverlauf, Lernphase]
= QRS_Detektion (EKG_Signal, fa);
    size = length(EKG_Signal); % Laenge des Signals
    % fourier transformation
    signal = fft(EKG_Signal);
    % entfernen niedriger freqenzen anhand Laenge und Rate
    signal(1 : round(size*5/fa))=0;
    signal(end - round(size*5/fa) : end)=0;
    % inverse fourier transformation
    signal=real(ifft(signal));

    % bandpass fuer 5 bis 30 hz
    signal=bandpass(signal, [5 30], fa);

    % skalierung des Signals auf skala 1-10
    filtersignal=signal/max(signal)/10;
    % entfernen aller peaks unterhalb des thresholds 10/2=5
    for data = 1:1:length(filtersignal)
        if filtersignal(data) < 5
            filtersignal(data) = 0;
        else
            filtersignal(data)=1;
        end
    end

    % finde alle uebrigen peaks, die ueber dem threshold lagen
    positions=find(filtersignal);
    % abstand der ersten beiden peaks
    distance=positions(2)-positions(1);
    % setze den abstand auf das minimum zweier peaks in der reihe aller uebrigen peaks
    for data=1:1:length(positions)-1
        if positions(data+1)-positions(data)<distance
            distance=positions(data+1)-positions(data);
        end
    end

    % Mittel der Signalamplitude als Threshold
    avg = (max(signal)/5)*4;

    % Absolute Peaks innerhalb des Zeitfensters finden
    [RPeaks, positions] = findpeaks(signal, 'MinPeakHeight', avg,
        'MinPeakDistance', distance, "MaxPeakWidth", 250);
    % positions[242] = 0; % optional zum entfernen von Ausschlaegen
    R_Positionen = positions;
    Entscheidungssignal = signal;
    Schwellwertverlauf = signal; % nicht verwendet
    Lernphase = 0; % nicht verwendet
end

```