

Addon zu den Übungsaufgaben I, SBV1

Lisa Panholzer, Lukas Fiel

November 6, 2018

1 Übungsaufgaben I, addon

1.1 Stressanalyse

a) Recherche, wie Stress aus EKG und HRV abgeleitet werden kann

Die Herzratenvariabilität beschreibt generell die Fähigkeit, dass sich die Herzfrequenz je nach Umwelteinflüssen (physische und psychische Belastung) anpassen kann. Je nach Situation kann die Herzratenvariabilität ansteigen oder sich reduzieren. Auch bei einem gesunden Herzen variiert die Frequenz des Herzschlages.

Allgemein kann gesagt werden, dass körperliche Beanspruchung bzw. kurze psychische Belastung eine Erhöhung der Herzfrequenz zur Folge habe. Entspannt sich diese Situation in der Umwelt, geht die Herzfrequenz wieder zurück.

Wird jedoch eine psychische Belastung wie zum Beispiel Stress zu einer Dauerbelastung, kann sich der Organismus hier gut anpassen. Bei einer chronischen Stresssituation (wenig Wechsel zwischen An- und Entspannung) ist die Anpassungsfähigkeit eingeschränkt. Daraus folgt, dass die Herzfrequenz hier niedrig ist.

Es gibt unterschiedliche Methoden, wie Stress aus einer EKG Messung bzw. der Herzratenvariabilität abgeleitet bzw. dargestellt werden kann.

- Histogramm: In einem festgelegten Zeitraum werden die Herzschläge gemessen und gezählt. In dem Histogramm wird dargestellt, wie viele der Herzschläge in eine Klasse fallen.
- Standardabweichung: Hiermit kann die Streuung der RR-Intervalle bestimmt werden.
- Spektralanalyse.

b) Analyse der EKG Sequenzen

c) Herzratenvariabilität

Analyse des Hautleitwerts

1.2 Geschwindigkeitsermittlung

a) Ermittlung von Geschwindigkeit und Distanz

Aufgabenstellung: Zu untersuchen war ein Datensatz, der mit Daten eines Beschleunigungssensors gefüllt war. Bekannt war, dass es sich um die Daten einer geradlinigen Bewegung entlang einer Tischkante handelte die etwa 2 m lang war. Neben den Daten des Sensors waren auch Zeitstempel enthalten. In Figure 1 ist gut zu erkennen, dass die Werte des Datensatzes bereits bearbeitet wurden. Diese Tatsache wurde in der Vorlesung erwähnt. Unseres Wissens wurde eine Baseline - Korrektur durchgeführt und alle Werte der Messung bis zur zu beobachtenden Bewegung wurden 0 gesetzt.

Idee:

Eine ersten Auswertung der Daten mittels Octave zeigt Figure 1. Ein reines Aufsummieren der Werte vermittelte einen ersten Eindruck. Die Summe der Messungen (im Folgenden "Geschwindigkeit" genannt) ist an diesem Punkt aber negativ. Da wir von einer Ruhelage des Objekts am Ende der Bewegung ausgehen gilt es diesen Wert zu korrigieren.

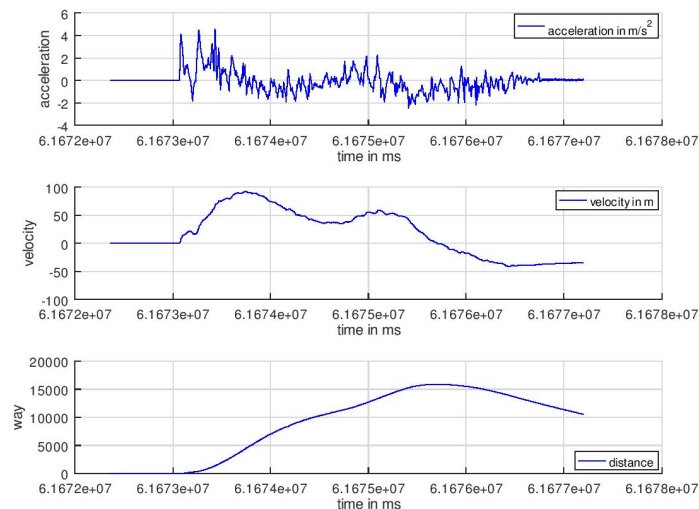


Figure 1: Die Daten des Beschleunigungssensors wurden aufsummiert um die Geschwindigkeit zu erhalten. Dieser Prozess wurde wiederholt um einen ersten Eindruck über ein mögliches Wegsignal zu gewinnen.

Korrektur:

Die Beschleunigungsdaten werden mittels eines gleitenden Mittelwerts gefiltert. Jeder Wert unter einem gewissen Threshold wird zu 0 gesetzt. Weil am Ende der Messung nur leichtes Rauschen erkennbar ist, können diese Werte zu 0 gesetzt werden was ein Erkennen des Bereichs der interessanten Bewegung möglich macht. Nach dem Aufsummieren der Beschleunigungsdaten muss sich nun darum gekümmert werden, dass die Geschwindigkeit am Anfang und Ende der Messung 0 ist. Da die anfänglichen Daten sowieso 0 sind muss nur das Ende betrachtet werden. Hier erkennt man eine wesentliche Abweichung. Es wird ein linearer Fehler angenommen, den es zu korrigieren gilt, wesshalb eine linear ansteigende Korrektur des Offset implementiert wurde. Aus den so gewonnenen Geschwindigkeitsdaten konnte wiederum der zurückgelegte Weg durch Summation berechnet werden. Eine Abbildung der bearbeiteten Daten zeigt Figure 2. Die Auswertung ergibt, dass der mittels bearbeiteten Daten errechnete Weg (15866mm) viel weiter als der durch die gefilterten Daten errechnete Weg (21412.68931mm) von der Annahme einer Bewegung über 2 m Distanz abweicht.

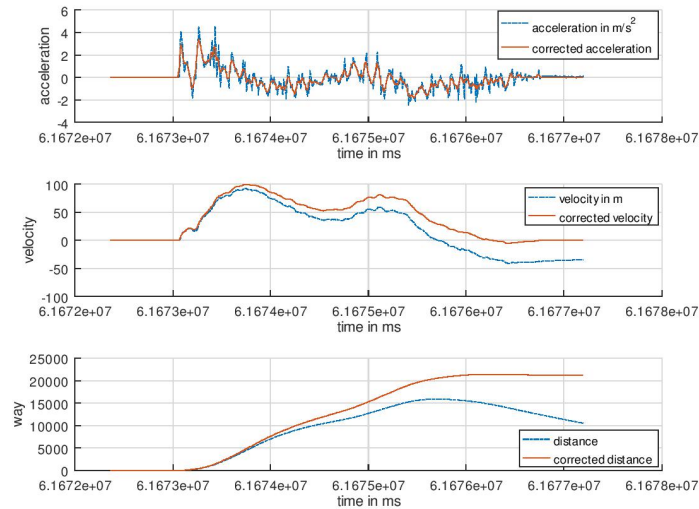


Figure 2: Gefilterte und korrigierte Darstellung der Daten.

```

; columns
1
2 clc
3 clear all
4 close all
5
6 pkg load io
7
8 'load_files_from_'
9 folderPath = 'C:/Users/Lukas/Documents/Signal-und-Bildverarbeitung/workspace/project/
    ↳ plugins/SBVplugin/SBVaddon/testData/';
10 fileName = 'acceleration.csv';
11 filePath = [folderPath, fileName]
12
13 acceleration = xlsread(filePath, 'tisch2m', 'A2:B577');
14
15 'process_plain_data'
16 velocity(1) = 0; # assume the velocity to be zero at the beginning
17 N = length(acceleration(:,1));
18 for j=2:N
19     velocity(j) = acceleration(j,2) + velocity(j-1);
20 endfor
21
22 way(1) = 0; # assume the way to be zero at the beginning
23 for j=2:N
24     way(j) = velocity(j) + way(j-1);
25 endfor
26
27 'plot_plain_data'
28 fig1 = figure();
29 subplot(3,1,1);hold on;
30 xlabel("time in ms");
31 ylabel("acceleration");
32 grid on
33 plot(acceleration(:,1),acceleration(:,2),'-b;acceleration in m/s^2;");
34 subplot(3,1,2);hold on;
35 xlabel("time in ms");
36 ylabel("velocity");
37 grid on
38 plot(acceleration(:,1),velocity,'-b;velocity in m;")
39 subplot(3,1,3);hold on;
40 xlabel("time in ms");
41 ylabel("way");
42 grid on
43 plot(acceleration(:,1),way,'-b;distance;")
44 legend("location","southeast");
45
46 'begin_CORRECTION'
47 # ACCELERATION CORRECTION
    ↳ #####
48 time = acceleration(:,1);
49 correctedAcceleration = acceleration(:,2);
50 # ACCELERATION - floating mean filter
51 radius = 2;
52 mask = 2 * radius +1;
53 for j = (radius+1) : N- (radius+1)
54     correctedAcceleration(j) = mean(correctedAcceleration(j-radius:j+radius));
55 endfor
56
57 # ACCELERATION - threshold
58 THRESHOLD = 0.12;
59 for j = (radius+1) : N- (radius+1)
60     meanValue = abs(mean(correctedAcceleration(j-radius:j+radius)));
61     if (meanValue < THRESHOLD)
62         correctedAcceleration(j) = 0;
63     endif
64 endfor
65 correctedAcceleration(1:radius) = correctedAcceleration(radius+1);
66 correctedAcceleration(N-radius:N) = correctedAcceleration(N-radius-1);
67
68 # VELOCITY CALCULATION
69 correctedVelocity(1) = 0; # assume the velocity to be zero at the beginning
70 N = length(time);
71 for j=2:N
72     correctedVelocity(j) = correctedAcceleration(j) + correctedVelocity(j-1);
73 endfor
74 correctedTmpVelocity = correctedVelocity;

```

```

75
76 # VELOCITY CORRECTION
77 # find indizes of actual movement in accelerator data
78 nonzeroIndizes = find(correctedAcceleration(:)); # find nonzero data
79 lastPrecedentZeroIndex = min(nonzeroIndizes); # get first nonzero index
80 lastNonZeroIndex = max(nonzeroIndizes); # get last nonzero index
81
82 # increasing offset correction
83 # calculate line from first meaningful data to the last
84 k = correctedVelocity(length(correctedVelocity))/(lastNonZeroIndex -
    ↳ lastPrecedentZeroIndex);
85 d = - k * lastPrecedentZeroIndex;
86 correctedVelocity(1:lastPrecedentZeroIndex) = 0; # if there is no acceleration -> set
    ↳ velocity to 0
87 correctedVelocity(lastNonZeroIndex:N) = 0; # if ther is no acceleration -> set velocity
    ↳ to 0
88 for j=lastPrecedentZeroIndex:lastNonZeroIndex-1
89     correctedVelocity(j) = correctedVelocity(j) + abs(k * j + d);
90 endfor
91
92 # DISTANCE CALCULATION
93 correctedWay(1) = 0; # assume the way to be zero at the beginning
94 for j=2:N
95     correctedWay(j) = correctedVelocity(j) + correctedWay(j-1);
96 endfor
97
98
99 'plot_result'
100 distance = max(way)
101 correctedDistance = max(correctedWay)
102
103 figure()
104 subplot(3,1,1)
105 grid on
106 hold on
107 xlabel("time in ms");
108 ylabel("acceleration");
109 plot(acceleration(:,1),acceleration(:,2),"-.;acceleration in m/s^2;");
110 plot(time,correctedAcceleration,"-;corrected acceleration;");
111 subplot(3,1,2)
112 hold on
113 grid on
114 xlabel("time in ms");
115 ylabel("velocity");
116 plot(acceleration(:,1),velocity,"-.;velocity in m;");
117 plot(time,correctedTmpVelocity,"-;corrected velocity;");
118 plot(time,correctedVelocity,"-;corrected velocity with linear shift;");
119 subplot(3,1,3)
120 hold on
121 grid on
122 plot(time,way,"-.;distance;");
123 plot(time,correctedWay,"-;corrected distance;");
124 legend("location","southeast");
125 xlabel("time in ms");
126 ylabel("way");

```