
Table of Contents

.....	1
VT1 Berechnung für Delphi uninteressant	1
Vt2 finden für Delphi wichtig	3
für Delphi uninteressant	6

```
function [EQ_Watt_VT1,EQ_Watt_VT2,EQ_HF_VT1,EQ_HF_VT2,Stufe] =  
    EQ_VT(breath,subfolders,subfolderNumber,Stufe)  
  
%Funktion findet die aktuelle Probandenid heraus und bestimmt danach  
    die  
%passende Schwelle  
  
[~,Alter,Geschlecht] = userdata;  
  
if subfolderNumber <= 33  
    Probanden_id = str2double(subfolders(subfolderNumber).name(1:2));  
else  
    Probanden_id = subfolderNumber-1;  
  
end  
%bestimmen des maximal gemessenen VO2max  
VO2max=max(breath.Gesamtergebnis(:,9));  
  
%huntstudie wird aufgerufen, welche den ungefähren Bereich für die  
    zweite  
%Schwelle bestimmt.  
[RCP_VO2, Toleranz_faktor,Heartrate,Heartrate_var] =  
    hunt_studie(Alter(Probanden_id),Geschlecht(Probanden_id));  
  
%Range bestimmen für die mögliche zweite Schwelle  
  
VT2_min = VO2max*(RCP_VO2-Toleranz_faktor);  
VT2_max = VO2max*(RCP_VO2+Toleranz_faktor);  
VT2_Durchschnitt = VO2max * RCP_VO2;  
  
stages=size(breath.Gesamtergebnis,1);
```

VT1 Berechnung für Delphi uninteressant

```
%Finden vom Minimum von EQO2
```

```
VT1_EQ=0;
```

```

%neue Variante

for EQloop=2:round(size(breath.Gesamtergebnis,1)*0.6)

    Steigung_O2(EQloop)= breath.Gesamtergebnis(EQloop+1,11)-
breath.Gesamtergebnis(EQloop,11);
    Steigung_CO2(EQloop)= breath.Gesamtergebnis(EQloop+1,12)-
breath.Gesamtergebnis(EQloop,12);

    if Steigung_O2(EQloop)>0.2 && Steigung_CO2(EQloop)<0 &&
VT1_EQ==0 || Steigung_O2(EQloop) > 0 && Steigung_O2(EQloop) >
1.6*Steigung_CO2(EQloop) && VT1_EQ==0 && Steigung_CO2(EQloop) > 0

        VT1_EQ=EQloop;

    end

end

% if VT1_EQ==0

    [~,VT1_EQ] = min(breath.Gesamtergebnis(:,11));

% end


% %bestimmen der Wattzahl bei VT1
% m1 = breath.Gesamtergebnis(VT1_EQ+1,11)-
breath.Gesamtergebnis(VT1_EQ,11);
% watt_m2 = breath.Gesamtergebnis(VT1_EQ+1,14)-
breath.Gesamtergebnis(VT1_EQ,14);
% watt_m = m1/watt_m2;
%
% watt_n = breath.Gesamtergebnis(VT1_EQ+1,11)-
(watt_m*breath.Gesamtergebnis(VT1_EQ+1,14));
%
%
% VT1_EQ_watt =
breath.Gesamtergebnis(VT1_EQ,11)+( (breath.Gesamtergebnis(VT1_EQ
+1,11)-breath.Gesamtergebnis(VT1_EQ,11))/2);
%
%
% EQ_Watt_VT1 = (VT1_EQ_watt-watt_n)/watt_m ;
%
%
%
%
% %Bestimmen der zugehörigen Herzfrequenz

```

```

% m1 = breath.Gesamtergebnis(VT1_EQ+1,13)-
breath.Gesamtergebnis(VT1_EQ,13);
% m2 = breath.Gesamtergebnis(VT1_EQ+1,14)-
breath.Gesamtergebnis(VT1_EQ,14);
% m = m1/m2;
% % Bestimmen des VO2 werts
% n = breath.Gesamtergebnis(VT1_EQ+1,13)-
(m*breath.Gesamtergebnis(VT1_EQ+1,14));
%
% EQ_HF_VT1 = (EQ_Watt_VT1*m)+n;
%

```

Vt2 finden für Delphi wichtig

```

%Es wird das Minimum vom EQCO2 gesucht von VT1EQ aus bis zur letzten
Stufe

[EQCO2_value, ~]= min(breath.Gesamtergebnis(VT1_EQ:stages,12));

%danach wird der letzte Wert gesucht, der den gleichen Wert vom
Minimum
%hat +0,2l
VT2_EQ = find(breath.Gesamtergebnis(VT1_EQ:stages,12) < EQCO2_value
+0.21,1, 'last');

%Wegen des Befehls find ist der Index falsch, darum wird dieser
angepasst.

VT2_EQ = VT2_EQ+VT1_EQ-1;

%

% vt2_begrenzung = round(size(diff,2)*3/4);
%
% %VT2EQ wird gefunden. Danach wird geprüft, ob es auch stimmig ist
%
% %"find" gibt den Index innerhalb der Suchmenge wieder, da diese
kleiner ist,
% %als der gesamte Vektor EQVCO2, wird vt2_begrenzung -1 addiert
% VT2_EQ =
find(diff(vt2_begrenzung:size(diff,2))>diff_gesamt ,1,'first') +
vt2_begrenzung-1;

%Überprüfung, hierzu wird der minimale Abstand zwischen VO2 und dem
Durchschnittsvo2 Wert von Huntstudie berechnet
%danach wird der Sprung von positiv auf negativ gesucht
%der Wert, welcher dichter an den beiden dran ist gewinnt

VO2_abst = VT2_Durchschnitt-breath.Gesamtergebnis(:,9);
EQVT2_hunt_ind = find(VO2_abst < 0 ,1,'first');

```

```

Stufe(1,1)=VT2_EQ;
Stufe(1,2)=EQVT2_hunt_ind;

% if abs(VO2_abst(VO2_abst_spr)) < VO2_abst(VO2_abst_spr-1)
%
%     EQVT2_hunt_ind = VO2_abst_spr;
%
% else
%
%     EQVT2_hunt_ind = VO2_abst_spr-1;
%
% end

%entspricht die Schwelle bereits dem Durchschnittswert der Hunt
%Studie, so
%kann dieser direkt verwendet werden.

if EQVT2_hunt_ind == VT2_EQ

elseif EQVT2_hunt_ind > VT2_EQ

    for loop=1:EQVT2_hunt_ind - VT2_EQ

        %wenn es stark aus dem Rahmen fällt an der bestimmten Schwelle,
        dann
        %soll die Schwelle erhöht werden.
        if (breath.Gesamtergebnis(VT2_EQ
+1,13)+breath.Gesamtergebnis(VT2_EQ,13))/2 <= (Heartrate-
Heartrate_var) && VT2_EQ < size(breath.Gesamtergebnis,1)
            [~, VT2_index]= min(breath.Gesamtergebnis(VT2_EQ
+1:stages,12));
            VT2_EQ=VT2_EQ+VT2_index;
        end
    end

else
    if breath.Gesamtergebnis(VT2_EQ,13) > Heartrate
+(0.5*Heartrate_var)

        VT2_EQ = VT2_EQ-1;

    end

end

%Wenn Person aus der Norm fliegt. Weil die HF niedriger ist, als in
%der
%Studie mit Abstand.

if VT2_EQ == size(breath.Gesamtergebnis,1)

```

```

%bestimmen der Wattzahl bei VT2
m1 = breath.Gesamtergebnis(VT2_EQ,12) -
breath.Gesamtergebnis(VT2_EQ-1,12);
watt_m2 = breath.Gesamtergebnis(VT2_EQ,14) -
breath.Gesamtergebnis(VT2_EQ-1,14);
watt_m = m1/watt_m2;

watt_n = breath.Gesamtergebnis(VT2_EQ,12) -
(watt_m*breath.Gesamtergebnis(VT2_EQ,14));

VT2_EQ_watt =
breath.Gesamtergebnis(VT2_EQ-1,12) + ((breath.Gesamtergebnis(VT2_EQ,12) -
breath.Gesamtergebnis(VT2_EQ-1,12))/2);

EQ_Watt_VT2 = (VT2_EQ_watt-watt_n)/watt_m ;

%Bestimmen der zugehörigen Herzfrequenz
m1 = breath.Gesamtergebnis(VT2_EQ,13) -
breath.Gesamtergebnis(VT2_EQ-1,13);
m2 = breath.Gesamtergebnis(VT2_EQ,14) -
breath.Gesamtergebnis(VT2_EQ-1,14);
m = m1/m2;
% Bestimmen des VO2 werts
n = breath.Gesamtergebnis(VT2_EQ,13) -
(m*breath.Gesamtergebnis(VT2_EQ,14));

EQ_HF_VT2 = round((EQ_Watt_VT2*m)+n);

else

%bestimmen der Wattzahl bei VT2
m1 = breath.Gesamtergebnis(VT2_EQ+1,12) -
breath.Gesamtergebnis(VT2_EQ,12);
watt_m2 = breath.Gesamtergebnis(VT2_EQ+1,14) -
breath.Gesamtergebnis(VT2_EQ,14);
watt_m = m1/watt_m2;

watt_n = breath.Gesamtergebnis(VT2_EQ+1,12) -
(watt_m*breath.Gesamtergebnis(VT2_EQ+1,14));

```

```

VT2_EQ_watt =
    breath.Gesamtergebnis (VT2_EQ,12)+((breath.Gesamtergebnis (VT2_EQ
+1,12)-breath.Gesamtergebnis (VT2_EQ,12))/2);

EQ_Watt_VT2 = (VT2_EQ_watt-watt_n)/watt_m ;

%Bestimmen der zugehörigen Herzfrequenz
m1 = breath.Gesamtergebnis (VT2_EQ+1,13)-
    breath.Gesamtergebnis (VT2_EQ,13);
m2 = breath.Gesamtergebnis (VT2_EQ+1,14)-
    breath.Gesamtergebnis (VT2_EQ,14);
m = m1/m2;
% Bestimmen des VO2 werts
n = breath.Gesamtergebnis (VT2_EQ+1,13)-(m*breath.Gesamtergebnis (VT2_EQ
+1,14));

EQ_HF_VT2 = round((EQ_Watt_VT2*m)+n);

end

```

für Delphi uninteressant

```

%Wenn die VT1 zu dicht bestimmt wurde, soll noch einmal die Steigung
%gesucht werden. Der Unterschied zu vorher ist, dass die Steigung von
O2
%hier nur größer 0 sein muss und nicht mehr > 0,2
%Dadurch ergeben sich neue Möglichkeiten.

if VT2_EQ - (VT1_EQ) <=1

    VT1_EQ=0;
    for EQloop=2:round(size(breath.Gesamtergebnis,1)*0.6)

        if Steigung_O2(EQloop)>0 && Steigung_CO2(EQloop)<0 &&
VT1_EQ==0 || Steigung_O2(EQloop) > 0 && Steigung_O2(EQloop) >
1.6*Steigung_CO2(EQloop) && VT1_EQ==0 && Steigung_CO2(EQloop) > 0

            VT1_EQ=EQloop;

        end

    end

    if VT1_EQ==0

        [~,VT1_EQ] = min(breath.Gesamtergebnis(:,11));

    end

```

end

```
m1 = breath.Gesamtergebnis(VT1_EQ+1,11)-  
breath.Gesamtergebnis(VT1_EQ,11);  
watt_m2 = breath.Gesamtergebnis(VT1_EQ+1,14)-  
breath.Gesamtergebnis(VT1_EQ,14);  
watt_m = m1/watt_m2;
```

```
watt_n = breath.Gesamtergebnis(VT1_EQ+1,11)-  
(watt_m*breath.Gesamtergebnis(VT1_EQ+1,14));
```

```
VT1_EQ_watt =  
breath.Gesamtergebnis(VT1_EQ,11)+((breath.Gesamtergebnis(VT1_EQ  
+1,11)-breath.Gesamtergebnis(VT1_EQ,11))/2);
```

```
EQ_Watt_VT1 = (VT1_EQ_watt-watt_n)/watt_m ;
```

```
%Bestimmen der zugehörigen Herzfrequenz  
m1 = breath.Gesamtergebnis(VT1_EQ+1,13)-  
breath.Gesamtergebnis(VT1_EQ,13);  
m2 = breath.Gesamtergebnis(VT1_EQ+1,14)-  
breath.Gesamtergebnis(VT1_EQ,14);  
m = m1/m2;  
% Bestimmen des VO2 werts  
n = breath.Gesamtergebnis(VT1_EQ+1,13)-(m*breath.Gesamtergebnis(VT1_EQ  
+1,14));
```

```
EQ_HF_VT1 = round((EQ_Watt_VT1*m)+n);
```

end

Published with MATLAB® R2018a