



ONTWIKKELING VAN EEN  
ALGORITME OM DE REACTIVE  
STRENGTH INDEX BIJ EEN  
COUNTERMOVEMENT JUMP  
TE BEPALEN

Afstudeerscriptie Bewegingstechnologie  
2017/2018

Auteur: Jorrit Cornelissen  
Studentnummer: 12055530  
Datum: 13-06-2018



## Personalia

### Auteur:

Naam: Jorrit Cornelissen  
 Studentnummer: 12055530  
 Geboortedatum: 14-10-1993  
 Email: [Jorritcornelissen@gmail.com](mailto:Jorritcornelissen@gmail.com)

### Opdrachtgever:

Naam: Nederlandse Volleybalbond  
 Onderdeel: Beach Team Nederland  
 Contactpersoon: Menzo Tanis (performance analist)  
 David Jones (*Strength & Conditioning Coach*)  
 Email: Menzo.tanis@nevobo.nl  
 David.jones@nocnsf.nl

### Begeleider:

1<sup>e</sup>

Naam: A. H. Lagerberg, MSc  
 Faculteit: Gezondheid, Voeding & sport  
 Functie: Docent bewegingstechnologie  
 Email: [A.H.Lagerberg@hhs.nl](mailto:A.H.Lagerberg@hhs.nl)

2<sup>e</sup>

Naam: C.A.M. Doorenbosch, PhD  
 Faculteit: Gezondheid, Voeding & sport  
 Functie: Docent bewegingstechnologie  
 Email: [C.A.M.Doorenbosch@hhs.nl](mailto:C.A.M.Doorenbosch@hhs.nl)

**Opleiding:** Bewegingstechnologie

**Instituut:** De Haagse Hogeschool

**Jaar:** 2017/2018

## Voorwoord

Dit afstudeeronderzoek is uitgevoerd in het kader van de opleiding Bewegingstechnologie aan de Haagse Hogeschool. Dit rapport is bedoeld voor de coaches van Beachvolleybal Team Nederland van de Nederlandse Volleybalbond en eenieder die zich in dit onderwerp interesseert.

Allereerst wil ik Aad Lagerberg bedanken voor de intensieve begeleiding die ik van hem heb gekregen tijdens deze afsluitende fase van mijn studie. Ik had nooit verwacht dat ik programmeren als leuk zou kunnen ervaren, en dat is met zijn hulp wel gebeurd. Ten tweede wil ik David Jones en Menzo Tanis bedanken voor de kans die ik heb gekregen om mijn afstudeeronderzoek te doen voor Beachvolleybal Team Nederland. Verder wil ik Michiel Krijger–Hombergen bedanken voor de assistentie bij het in werking stellen van de *forceplate* apparatuur. Tot slot wil ik graag dr Caroline Doorenbosch bedanken voor de feedback die ze mij heeft gegeven op eerdere versies van dit rapport.

## Samenvatting

Beachvolleybal Team Nederland (BTN), onderdeel van de Nederlandse Volleybalbond, is steeds op zoek naar vernieuwende elementen in trainingsmethoden die de prestaties van zijn (top)sporters kunnen verbeteren. Binnen dit kader bleek er behoefte te zijn aan een algoritme om de *Reactive Strength Index* (RSI), een veel gebruikte meetuitkomst die informatie geeft over de snelheid van de krachtleverantie, te bepalen bij een *countermovement jump* (CMJ). De berekening van de RSI is de spronghoogte gedeeld door de contacttijd. De data om de RSI te berekenen wordt verkregen met behulp van een *forceplate*.

Het doel van dit project is om een voor de gebruiker eenvoudig toepasbaar algoritme te ontwikkelen waarmee de RSI bij een CMJ berekend kan worden.

Het geschreven algoritme is ontwikkeld en zodanig uitgewerkt dat de (toekomstige) gebruiker (BTN) beperkte voorkennis nodig heeft om het algoritme te kunnen gebruiken. Daarnaast voldoet het aan alle, door BTN vooraf gestelde eisen, en is het in gebruik genomen.

Voor de meting in dit onderzoek zijn 18 atleten (7 mannen, 11 vrouwen) geselecteerd door BTN. Zij hebben 5 keer een *countermovement jump* uitgevoerd met een gemiddelde spronghoogte van  $0.35 \pm 0.06$  (m), afzettijd  $0.95 \pm 0.09$  (s) en  $RSI_{mod} 0.38 \pm 0.07$ .

Omdat BTN ook openstaat voor adviezen en aanbevelingen met betrekking tot sprongtraining, zijn de achtergronden en de beoogde toepassingen van de RSI onderzocht om meer inzicht te krijgen in deze parameter. De eenheid RSI is vooral bedoeld om inzicht te krijgen in de *Stretch Shortening Cycle* (SSC) die de springer tijdens de afzet doorloopt.

Uit nadere analyse bleek dat de correlatie tussen spronghoogte en de  $RSI_{mod}$  ( $r = 0.84$ ) hoog is. Hetgeen wil zeggen dat de  $RSI_{mod}$  weinig aanvullende informatie geeft ten opzichte van de spronghoogte. De RSI is een verkapte weergave van het vermogen. Maar met een *forceplate* kan een meer gedetailleerde analyse gemaakt worden van het vermogen. De correlatie tussen het voor lichaamsgewicht genormaliseerde vermogen en de spronghoogte bleek sterk ( $r = 0.80$ ). Net als de  $RSI_{mod}$  geeft dit weinig aanvullende informatie.

Uit de resultaten verkregen met het gemaakte algoritme blijkt dat de RSI wel iets zegt over de snelheid van de *countermovement* maar niet iets over het effect van de *countermovement* op de sprongprestatie. Tijdens de *countermovement* wordt een SSC doorlopen. Om inzicht te krijgen in de SSC is er gekeken naar twee elementen, de *Residual Force Enhancement* en de voorspanning. Deze voorspanning wordt gedefinieerd als de toename van de *Ground Reaction Force* (GRF) als gevolg van de *countermovement*. De voorspanning is voor het lichaamsgewicht genormaliseerd om vergelijking van atleten mogelijk te maken. Er is gekeken of de procentuele toename van de GRF een verband heeft met de spronghoogte. Dit bleek slecht te correleren ( $r = 0.17$ ).

De verdieping in de achtergronden van de SSC heeft geleid tot het advies om onderzoek te doen naar de procentuele spronghoogte-toename tussen de *squat jump* en de CMJ. De volgende stap in een vervolgstudie zou kunnen zijn om te bepalen in hoeverre de voorspanning correleert met de toename in spronghoogte (SJ versus CMJ). Als deze correlatie sterk blijkt, kan met behulp van een *forceplate* mogelijk een feedbacktool worden ontwikkeld, waarmee specifiek getraind kan worden op het verhogen van de effectiviteit van de *countermovement*.

## Inhoudsopgave

1 Inleiding en opdrachtomschrijving .....	5
2. Het spronganalyse script .....	7
2.1 Eisen en wensen .....	7
2.2 De ontwikkeling van het algoritme. ....	7
3.Theoretische verdieping.....	15
3.1 Stretch Shortening Cycle .....	15
3.2 Mechanische en spierfysiologische werkingsprincipes van de SSC .....	16
3.3 RSI <sub>mod</sub> of vermogen? .....	20
3.4 De meerwaarde van RSI <sub>mod</sub> of vermogensbepalingen.....	20
3.5 Het advies .....	22
Referenties .....	25
Bijlagen .....	26
Bijlage 1: Het geschreven script. ....	26
Bijlage 2: Sprongresultaten .....	34

## 1 Inleiding en opdrachtsomschrijving

Sinds de zomer van 2017 is in het Zuiderpark in Den Haag de Haagse Sportcampus gevestigd, een bruisende ontmoetingsplaats voor sporters. Dit nieuwe sportcomplex is ambitieus en wil toonaangevend zijn in de ontwikkeling van (top)sport. De campus beschikt over een keur aan faciliteiten voor veel verschillende sporten, zoals voor judo, (beach)volleybal, dans en basketbal. Daarnaast is het een kenniscentrum met sport gerelateerde opleidingen van de Haagse Hogeschool en het ROC Mondriaan. Onderdeel van de campus zijn de indoor en outdoor beachvolleybalvelden waar de Nederlandse olympische beachvolleybaltop traint en wedstrijden speelt. Beachvolleybal Team Nederland (BTN), onderdeel van de Nederlandse Volleybalbond (NeVoBo), heeft hier zijn thuisbasis.

Om de topsporters te monitoren op prestaties (*performance*) en om wedstrijdanalyses uit te voeren, maakt BTN gebruik van een speler-volgsysteem. Dit systeem bevat uitgebreide sets van wedstrijdgegevens en trainingsresultaten. Twee keer per jaar zijn er meetmomenten om de progressie van de atleten te monitoren. Hierbij wordt onder meer de spronghoogte bepaald van een *spike jump* (korte aanloop en vervolgens een maximale verticale sprong) en een *block jump* (een verticale sprong zonder aanloop). Deze metingen worden uitgevoerd met een *Vertec jump test*, waarbij gebruik gemaakt wordt van een spronghoogtemeter (de Vertec – Figuur 1) met horizontale lamellen (links Figuur 1) die weggeslagen dienen te worden om de spronghoogte te bepalen.

BTN heeft behoefte aan een meer gedetailleerde analyse van de afzetfase van een *countermovement jump*. Hierbij zou gebruikt gemaakt kunnen worden van een *forceplate*, een instrument waarmee de zogenaamde *Ground Reaction Force* (de kracht die de grond uitoefent op een lichaam dat er op staat) gemeten kan worden. Op de sportcampus in het Zuiderpark zijn drie *forceplates* van KISTLER aanwezig. De eerste ligt in het Bewegingslaboratorium, een ruimte met verschillende IT-faciliteiten die pas onlangs is. De tweede en derde *forceplate* liggen in de gymzaal aangrenzend aan het laboratorium.



Figuur 1 De Vertec, een spronghoogtemeter met horizontale lamellen

BTN heeft aangegeven informatie te willen over de snelheid waarmee de sporters kracht kunnen ontwikkelen tijdens de afzetfase van een *countermovement jump*. Met die informatie kunnen

specifiekere en geïndividualiseerde krachttrainingsprogramma's ontworpen worden. Een veel gebruikte meetuitkomst die informatie geeft over de snelheid van de krachtleverantie is de *Reactive Strength Index* (RSI) (Flanagan & Comyns, 2008) (de bedenkers van de RSI als meetuitkomst). De RSI wordt berekend door de spronghoogte te delen door de contacttijd (Flanagan & Comyns, 2008; Martinez, 2016). De RSI is ontwikkeld als onderdeel van de *Strength Qualities Assessment Test* aan de *Australian Institute of Sport* (Flanagan & Comyns, 2008; Martinez, 2016). De RSI van een *drop jump* (DJ), ook wel kaatssprong genoemd (waarbij de focus niet ligt op zo hoog mogelijk springen, maar op zo kort mogelijk contact maken met de ondergrond), is te bepalen met behulp van eenvoudige middelen, bijvoorbeeld een contactmat. Bij de berekening van de RSI van een *countermovement jump* wordt niet de berekening van de RSI gebruikt, maar van een gemodificeerde RSI, de zogenoemde  $RSI_{mod}$ . De modificatie van de RSI bestaat uit het vervangen van de contacttijd door de afzettijd. Voor de bepaling van de  $RSI_{mod}$  van een *countermovement jump* is een *forceplate* nodig omdat het begin van de afzetfase niet met een contactmat gedetecteerd kan worden; met een *forceplate* kan dit wel. Daarom, maar ook vanwege de beschikbaarheid ervan, is de keuze gemaakt met de *forceplate* aan de slag te gaan.

Door gebrek aan ervaring met het gebruik van de *forceplate*, is BTN zelf niet in staat om de gewenste analyses met betrekking tot de RSI uit te voeren. Daarnaast geven de standaard voorgeprogrammeerde protocollen geleverd in de standaard software van de *forceplate*, geen informatie over de RSI, afzettijd en spronghoogte, gegevens waaraan BTN behoefte heeft. Vandaar het verzoek van BTN om een op hun vraag toegespitst algoritme te ontwikkelen dat deze uitkomsten kan berekenen.

Daarnaast wil BTN graag een advies ontvangen over aanvullende mogelijkheden voor spronganalyses met behulp van de *forceplate*.

Dit heeft geleid tot de formulering van de volgende opdrachten:

*Ontwerp een voor BTN toepasbaar algoritme om uit forceplate data de gewenste uitkomstmaten te bepalen van de BTN-atleten.*

*Breng advies uit over aanvullende mogelijkheden voor spronganalyses met behulp van de forceplate.*

Dit verslag valt uiteen in twee onderdelen. Allereerst wordt in Hoofdstuk 2 de totstandkoming van het gemaakte algoritme toegelicht en wordt het gebruik geëvalueerd. Vervolgens wordt in het derde hoofdstuk theoretische verdieping gegeven over de RSI en een aanbeveling gedaan voor de mogelijkheden om met behulp van de *forceplate* de spronganalyse verder te verdiepen.

## 2. Het spronganalyse script

### 2.1 Eisen en wensen

Tijdens het schrijven van het algoritme zijn keuzes gemaakt en oplossingen bedacht om voorkomende problemen te verhelpen. De keuzes en oplossingen worden in dit hoofdstuk uiteengezet. Op basis van het gesprek met BTN zijn de volgende eisen opgesteld waaraan het algoritme moet voldoen:

- Berekening van de spronghoogte van een serie van 5 sprongen per atleet.
- Bepaling van de afzettijd van een serie van 5 sprongen per atleet.
- Berekening van de RSI van een serie van 5 sprongen per atleet.
- Berekening van de gemiddelde verticale snelheid in de concentrische fase.
- Berekening van de gemiddelde verticale kracht in de concentrische fase.
- Berekening van het gemiddelde verticale vermogen in de concentrische fase.
- Berekening van de gemiddelde verticale snelheid in de excentrische fase.
- Berekening van het gemiddelde en de standaarddeviatie van de hierboven genoemde uitkomstmaten
- De kracht en vermogens uitkomstmaten dienen genormaliseerd te worden weergegeven (gecorrigeerd voor lichaamsgewicht)
- Het algoritme dient gebruikt te kunnen worden door een gebruiker met basiskennis van Matlab.

Om de resultaten te kunnen gebruiken en toe te voegen aan het speler-volgsysteem heeft BTN voorkeur voor een weergave van de resultaten in een Excel-file. Hieruit komt de volgende vereiste voort:

- Het algoritme dient de resultaten te weg te kunnen schrijven naar een Excel-file.

### 2.2 De ontwikkeling van het algoritme.

Bij de totstandkoming van het algoritme zijn de volgende fasen doorlopen:

1. Hardware en software installatie
2. Pilotmeting van een sprong
3. Berekening van de gewenste uitkomstmaten op basis van de data van de pilotmeting
4. 1<sup>e</sup> uitvoering van de metingen bij de BTN atleten
5. Doorontwikkelen van het basisalgoritme op basis van de werkelijke metingen
6. Evaluatie.(bespreking van de resultaten met de opdrachtgevers)
7. Training van de opdrachtgevers tijdens de 2<sup>e</sup> uitvoering van metingen bij BTN atleten

#### Fase 1 ,2 en 3.

De *forceplate* in het lab moest nog in gebruik genomen worden. Hetgeen maakte dat er nog wat hardware en software problemen moesten worden opgelost voordat er daadwerkelijk gemeten kon worden. De bekabeling van de *forceplate* was nog incompleet en de bij de *forceplate* geleverde software moest nog worden geïnstalleerd. De zogenaamde free measurement mode van de *forceplate* software is gebruikt om een registratie van de verticale GRF bij een sprong te verkrijgen. De eerste pilotmeting van een sprong is vervolgens gebruikt om de gewenste uitkomstmaten in matlab te kunnen berekenen.



#### Fase 4.

In overleg met de opdrachtgevers is het gewenste protocol voor de spronganalyse ontwikkeld. Om aan te sluiten bij reeds bestaande analyses is er gekozen voor een serie van 5 CMJ sprongen met een pauze van 20 seconde na elke sprong. De sprongen werden uitgevoerd met een stok die in de nek werd geplaatst en aan weerszijde werd vastgehouden. De tijdsduur van de pauzes werd zichtbaar gemaakt mbv een smartphone voor de atleet. Er zijn 18 atleten gemeten (7 mannen en 11 vrouwen). De CSV files van deze spronggen zijn vervolgens gebruikt om het basisalgoritme verder door te ontwikkelen.

#### Fase 5.

Het basisscript dat geschreven was op een enkele sprong is uitgebreid naar een serie van 5 sprongen. Tevens is de export naar excel toegevoegd. Hieronder wordt het algoritme nader toegelicht

Het proces van de analyse van de sprongen kent twee stappen:

- Het uitvoeren van de meting
- De analyse van de uitkomsten

Bij het uitvoeren van de meting dient gebruik gemaakt te worden van de MARS-software welke is bijgeleverd bij de *forceplate*. Allereerst dient een gebruikers-id aangemaakt te worden per atleet. Hiermee koppelt de MARS-software de data verkregen met de sprong aan een gebruiker (atleet). Er wordt vervolgens gekozen voor de *free measure*. Er is vooraf ingesteld dat bij deze *free measure* alleen wordt gekeken naar de verticale component van de GRF. Alle andere mogelijkheden van de *forceplate* worden bij deze meting niet gebruikt om de omvang van het bestand zo klein mogelijk te houden. Bij aanvang van de meting dient de gebruiker op de *forceplate* te staan alvorens de meting wordt gestart. Nadat het protocol doorlopen is wordt de meting gestopt en direct geëxporteerd naar een .csv file. De data kan nu worden geanalyseerd.

Bij de analyse van de .csv file in Matlab worden de onderstaande stappen doorlopen:

- a. Inlezen data
- b. Selecteren van de sprongen
- a. c. Bepalen van het lichaamsgewicht en de zweeffase
- d. Bepalen van de afzettijd
- b. Berekenen spronghoogte en  $RSI_{mod}$
- c. Berekening gemiddelde snelheden, kracht en vermogen
- d. Berekenen standaarddeviaties en schrijven naar Excel

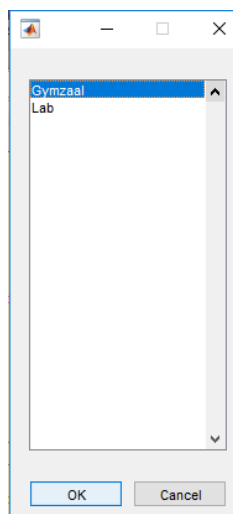
Het algoritme is zodanig geschreven dat het uitvoeren van de metingen bij de gebruiker minimale kennis van de Matlab-software vereist. Een gebruiker hoeft slechts het algoritme te openen, te starten en een data set te laden.

#### Ad a. Inlezen data

In de komende toelichting op het algoritme wordt steeds verwezen naar de diverse secties in de code (Bijlage 1). Deze secties zijn in het script weergegeven met procent-teken met daartussen een nummer. Het eerste deel van de code, sectie 1, geeft een menu weer waar de gebruiker een keuze moet maken van de *forceplate* die gebruikt is, die in het laboratorium of die in de gymzaal Figuur 2.

De reden hiervoor is dat het inlezen van de gegenereerde data per meetlocatie verschillend is vanwege het feit dat er in de gymzaal twee aan elkaar gekoppelde *forceplates* liggen. Als de

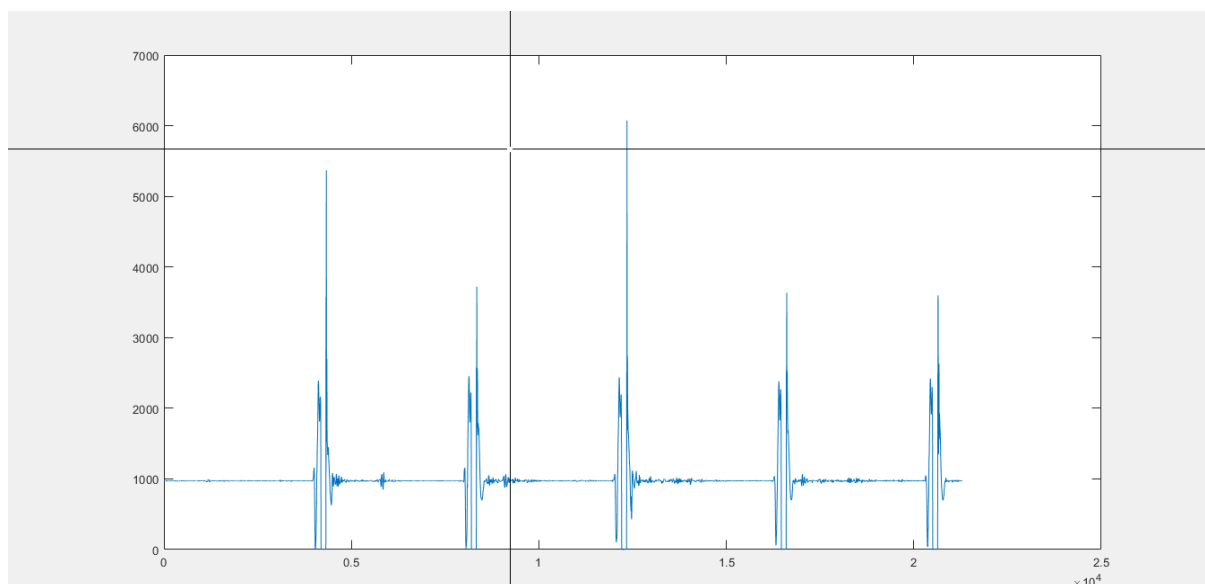
meetlocatie is aangeklikt gaat het programma verder met de vraag welk file (met sprong data per atleet) geladen moet worden.



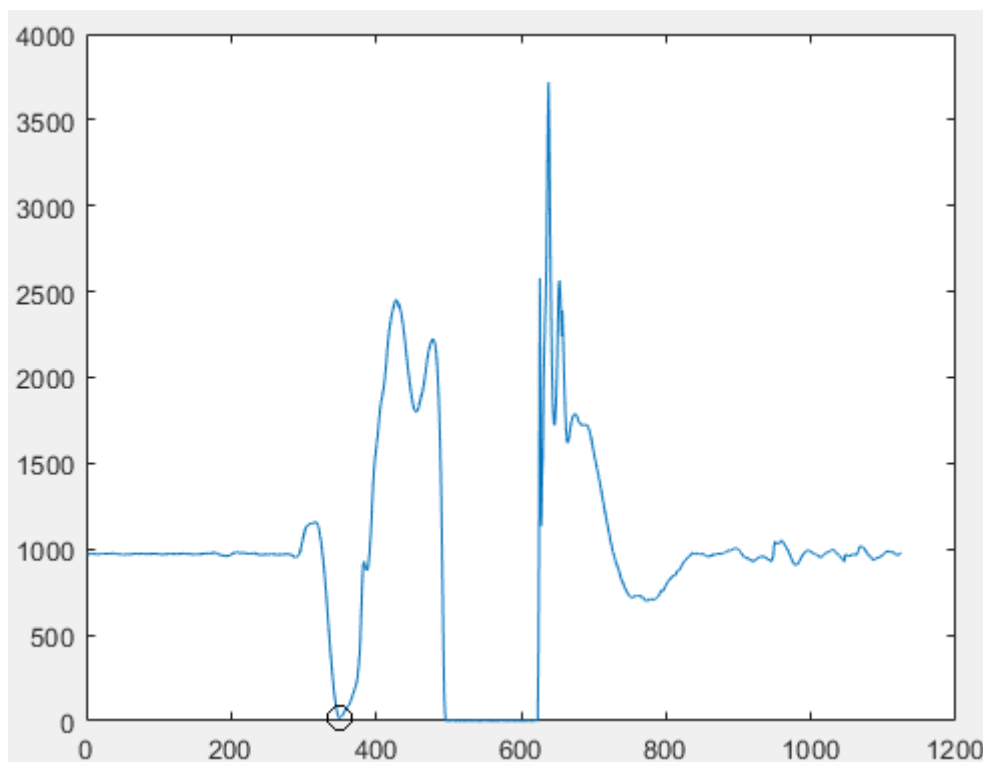
*Figuur 2 Weergave van het keuzemenu voor de testlocatie*

#### **Ad b.** Selecteren van de sprongen

Wanneer de file is geladen komt er een pop-up scherm (Figuur 3). Met behulp van de functie 'ginput' komt er een cursor in beeld waarmee de vijf sprongen geselecteerd dienen te worden (sectie 2). De sprongen worden geselecteerd door eerst vlak voor en vervolgens vlak na de sprong te klikken. Met behulp van een *for-loop* worden de samples van de sprongen opgeslagen. Zodat deze los van elkaar benaderd kunnen worden. Figuur 4 is een voorbeeld van een van de sprongen die geselecteerd is.



*Figuur 3 Weergave van het scherm waarbij de gebruiker de sprongen moet selecteren met behulp van de cursor*



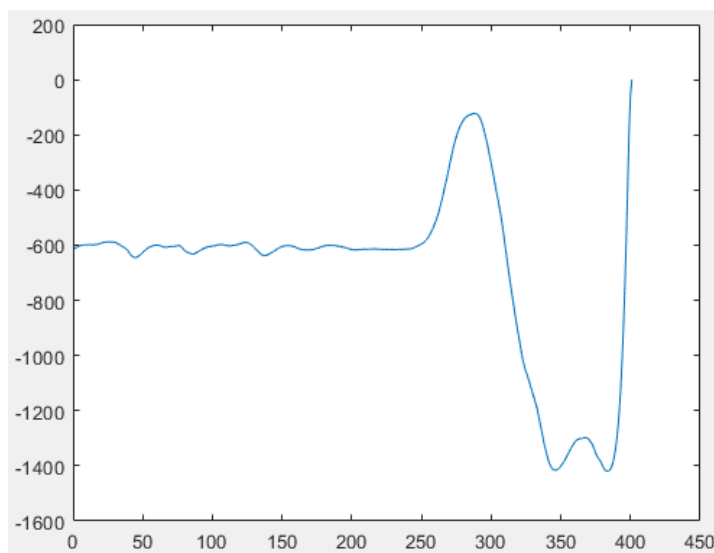
Figuur 4 Weergave van 1 CMJ signaal, zwarte cirkel is eerste dal.

#### Ad c. Bepalen van het lichaamsgewicht en de zweeffase

Nu de sprongen gedetecteerd zijn, wordt bepaald wanneer de springer in de lucht is tijdens de sprong. De detectie van deze zweeffase (sectie 3) wordt bepaald door middel van een grenswaarde. In het totale, gemeten signaal wordt vervolgens gekeken welk deel hiervan het meest bruikbaar is om het lichaamsgewicht mee te bepalen (sectie 4). Het script berekent per 200 samples wat de standaarddeviatie is. Het 200-samples-lange signaal met de laagste standaarddeviatie geeft dus de minste fluctuatie. Hiervan wordt het gemiddelde berekend. Dit wordt beschouwd als de meest nauwkeurige benadering van het lichaamsgewicht.

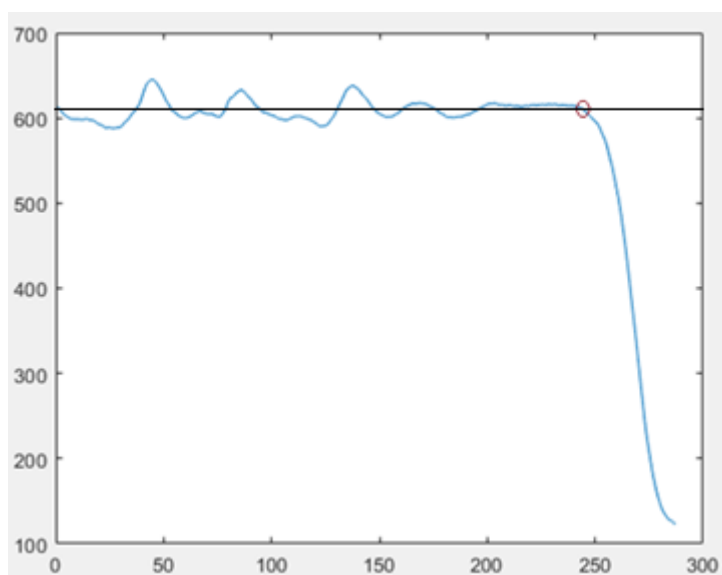
#### Ad d. Bepalen van de afzettijd

Nu het moment van *toe-off* bekend is moet de start van de sprong nog worden gezocht. Hiervoor wordt eerst gezocht naar het samplenummer waar zich het eerste dal bevindt (zwarte cirkel in Figuur 4). Doordat Matlab met behulp van de functie *findpeaks* geen dalen kan detecteren, maar alleen toppen is het totale signaal met -1 vermenigvuldigd om alles om de X-as te draaien zoals te zien in Figuur 5. Om andere pieken uit het signaal uit te sluiten zijn er grenswaarden gesteld voor de piekdetectie (sectie 5).



Figuur 5 Het signaal geknipt op TO en vermenigvuldigd met -1.

Na de piekdetectie in het signaal uit Figuur 5 is het signaal hierop geknipt (sectie 6). Het signaal, wanneer het weer met -1 is vermenigvuldigd, ziet er nu uit zoals weergegeven in Figuur 5. De start van de afzetfase wordt gedefinieerd als de laatste keer dat het signaal groter is dan lichaamsgewicht (Cirkel Figuur 6).



Figuur 6 Weergave van het signaal geknipt op het dal voor de benadering van de start van de afzetfase, cirkel. Zwarte lijn is het lichaamsgewicht.

#### Ad e. Berekenen spronghoogte en $RSI_{mod}$

Het beginpunt en het eindpunt van de afzetfase is nu bekend. Daarmee kan de duur van de afzetfase bepaald worden, de afzettijd. De spronghoogte is vervolgens bepaald met de impulsmethode (sectie 7). Met de resulterende kracht in de afzetfase en de tijdsduur waarbinnen deze kracht wordt geleverd, kan de verticale snelheid berekend worden door die twee met elkaar te vermenigvuldigen. Want het product van kracht en tijdsduur is gelijk aan de massa van de springer maal de snelheid waarmee de springer de grond verlaat (impuls) Formule 1 (Linthorn, 2001). Door de verticale

snelheid te kwadrateren en vervolgens te delen door twee keer de valversnelling wordt de spronghoogte berekend (formule 2). Door de spronghoogte op zijn beurt te delen door de afzettijd wordt de  $RSI_{mod}$  bepaald.

Formules:

$$F \cdot t = m \cdot v \quad (1)$$

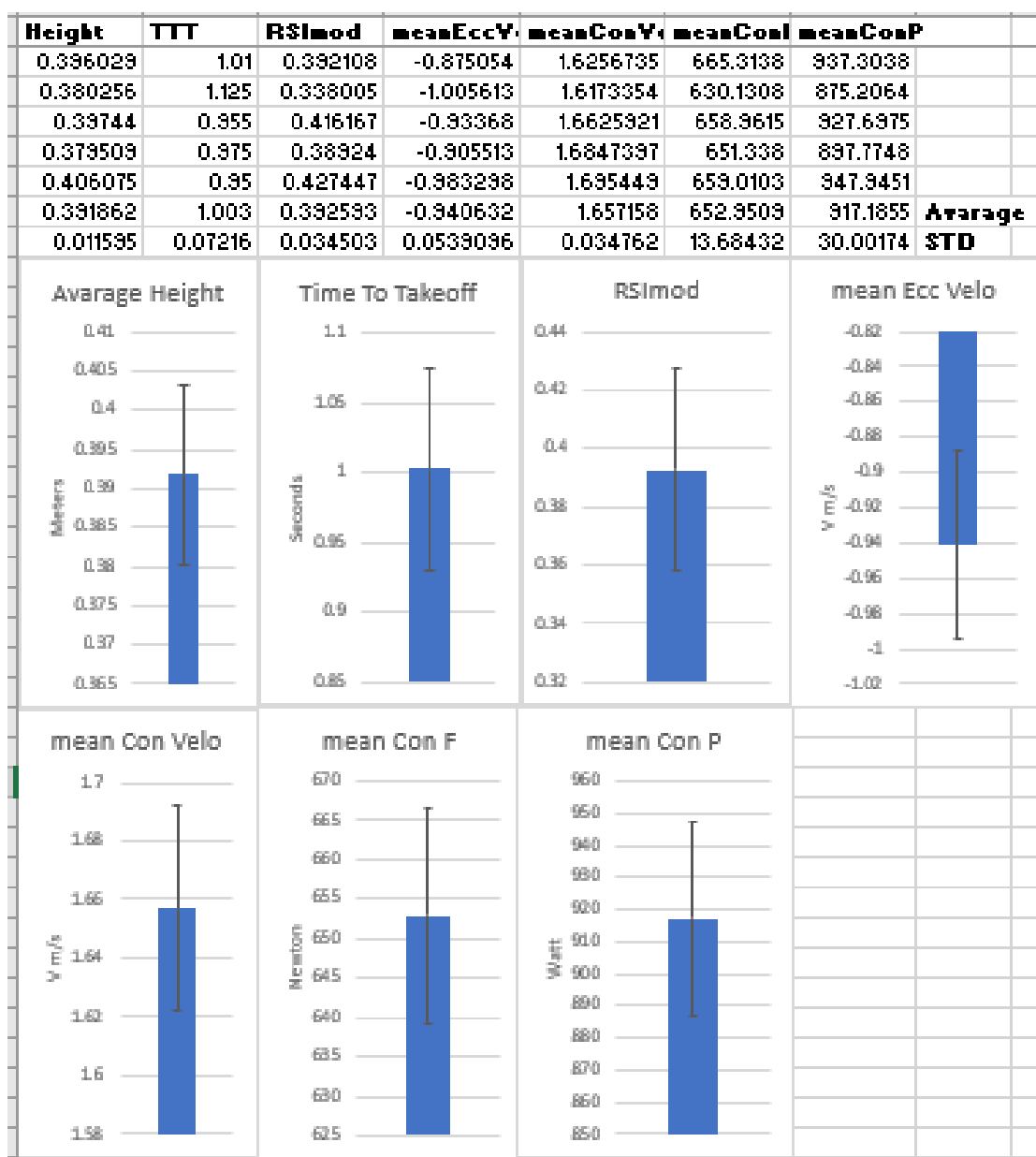
$$y_{flight} = y_{peak} - y_{to} = \frac{v_{to}^2}{2g} \quad (2)$$

#### **Ad f.** Berekening gemiddelde snelheden, kracht en vermogen

Naast de spronghoogte, afzettijd en de  $RSI_{mod}$  wil BTN ook de gemiddelde snelheid, kracht en vermogen in de concentrische fase van de sprong weten, en ook de gemiddelde snelheid in de excentrische fase (sectie 8). De snelheid (sectie 9) wordt bepaald door de versnelling te integreren, en met de functie *mean* wordt het gemiddelde per fase per sprong bepaald. Het vermogen en de kracht zijn beide gecorrigeerd voor het lichaamsgewicht. Het effect van het lichaamsgewicht is van het totale signaal afgetrokken om de resulterende kracht en het daardoor geleverde vermogen te kunnen bepalen.

#### **Ad g.** Berekenen standaarddeviaties en schrijven naar Excel

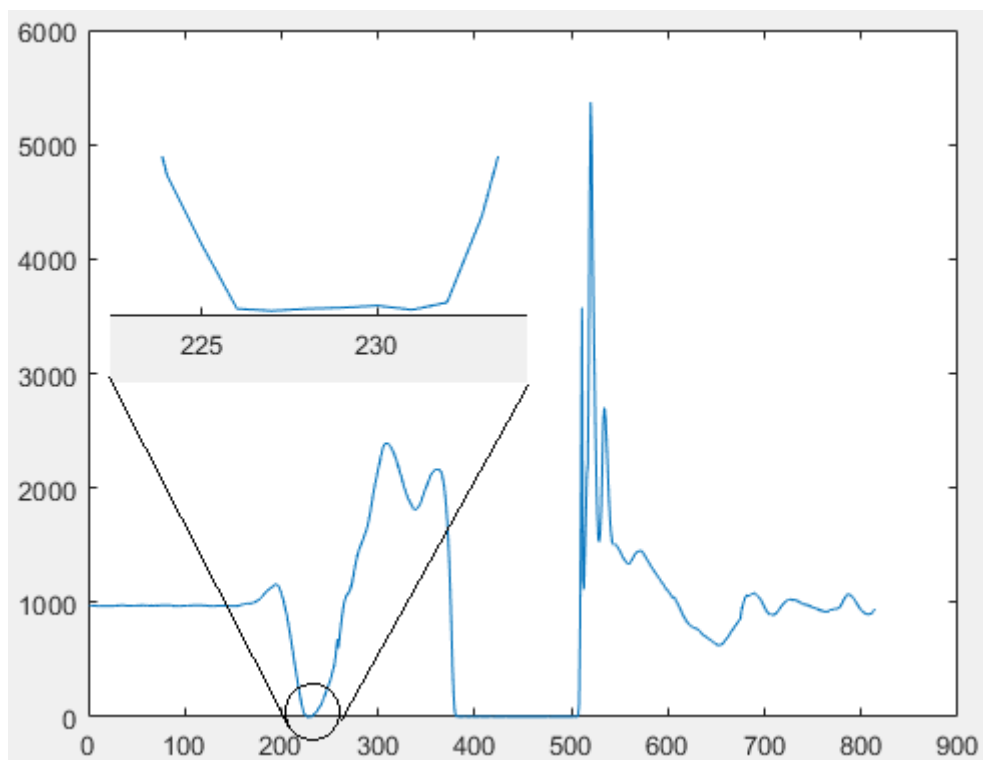
Tot slot worden de standaarddeviaties berekend van de parameters die verkregen zijn. Alle parameters worden geëxporteerd naar een tabel (sectie 10) welke vervolgens in een Excel template geladen wordt (Figuur 7). Elke keer nadat de analyse gedaan is, opent Excel met de uitkomst. Hierin zijn alle gegevens per sprong te zien, alsook de gemiddelden. Naast de waarden worden ook staafdiagrammen gemaakt van de parameters met de standaarddeviatie voor elke springer apart in een tabblad geladen.



Figuur 7 Weergave van de template waar de resultaten in zijn geladen.

## Fase 6. (Evaluatie)

Bij het verwerken van de metingen van de 18 BTN atleten bleek dat het algoritme niet bij alle spronguitvoering de correcte spronghoogte berekende. De vluchtfase detectie bleek een probleem op te leveren. Sommige springers wisten in de neerwaartse fase van de afzet de grondreactiekracht tot nul te reduceren. Er is vervolgens ingezoomd op het stuk uit het signaal waar dit optrad (zie figuur 8). Het algoritme zag dit ten onrechte als de start van de vluchtfase. Door aan de vluchtfase een eis te stellen van een minimale duur is deze onvolkomenheid uit het programma gehaald (zie tweede deel van de code in sectie 3).



Figuur 8 Weergave van een sprong waarbij de springer de grondreactiekracht tot nul weet te reduceren.

Ook bij springers die de GRF niet tot nul wisten te brengen in deze fase ontstond af en toe een probleem. Met de piekdetectie werden ook kleinere pieken gevonden dan de werkelijke piek van het dal. Door eisen (*peak prominence*) te stellen aan de piek is ook deze bug opgelost.

Na deze aanpassingen was het mogelijk om voor elke atleet de correcte uitkomsten te berekenen. Alleen de eis dat het algoritme voor BTN zelfstandig bruikbaar zou moeten zijn, was hiermee nog niet getest.

### Fase 7.

Een aantal weken later tijdens een tweede testdag is de werkwijze bij het uitvoeren en analyseren van de metingen uitgelegd aan de beoogde gebruikers van het script (de data analist van BTN en de krachttrainer). Aansluitend heeft de analist de meting uitgevoerd en de verwerking van de data gedaan met het algoritme. Met succes. Geconcludeerd kan daarom worden dat ook voldaan is aan de laatste eis. Tot slot zijn het algoritme en de daarbij horende extra bestanden overgedragen aan BTN en zijn inmiddels in gebruik genomen.

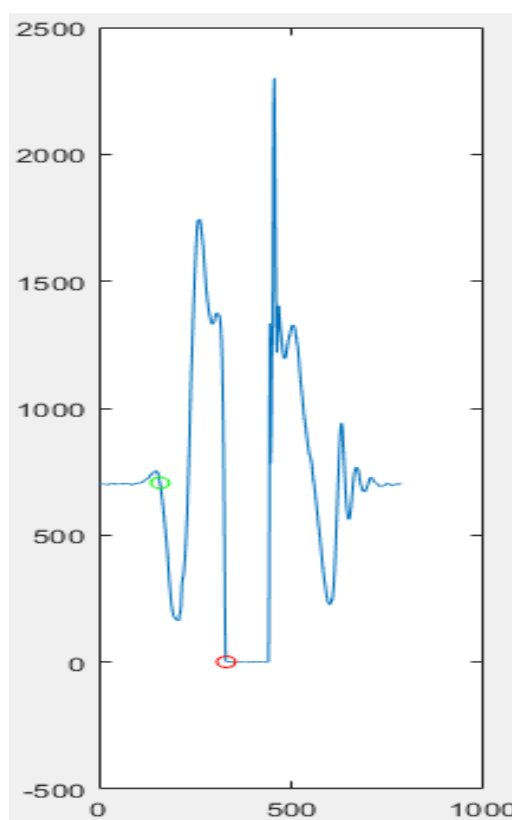
### 3.Theoretische verdieping

Om de gewenste aanbevelingen te kunnen formuleren is een studie verricht naar de achtergronden van de RSI en de SSC

#### 3.1 Stretch Shortening Cycle

Het vermogen van een individu om snel te veranderen van een excentrische naar een concentrische spiercontractie (de zogenaamde *Stretch Shortening Cycle*), wordt door Flanagan & Comyns (2008) beschouwd als maat van explosiviteit. Explosiviteit is een term die het vermogen beschrijft om maximale kracht te ontwikkelen in minimale tijd. Om het niveau van explosiviteit te kunnen kwantificeren wordt gebruik gemaakt van de  $RSI_{mod}$ . De  $RSI_{mod}$  wordt zoals eerder vermeld berekend door de spronghoogte (in meters) te delen door de tijdsduur van de afzetfase (in seconden) waarin de krachten worden gegenereerd voor de sprong. De duur van de afzettijd is bij sprongen waarbij wordt ingeveerd (zoals de *drop jump* en de *countermovement jump*) gelijk aan de tijdsduur van de SSC (Flanagan & Comyns (2008)).

De start van de afzetfase van de sprong wordt gedefinieerd als het punt in het signaal waar de curve voor het eerst onder lichaamsgewicht komt (groene cirkel in Figuur 9). Het einde van de afzetfase van de sprong is het punt waarbij de springer loskomt van de grond en de kracht op de *forceplate* nul wordt (rode cirkel in Figuur 9). Omdat de  $RSI_{mod}$  geïntroduceerd is om meer inzicht te krijgen in de wijze waarop de SSC van een atleet plaatsvindt, worden hieronder de mechanische en spierfysiologische werkingsprincipes van de SSC nader toegelicht. Deze inzichten worden later gebruikt om een advies uit te brengen met betrekking tot een verdere verdieping van de spronganalyses.



Figuur 9 Plot van een countermovement jump met start (groene cirkel) en eind (rode cirkel) van de afzetfase.



### 3.2 Mechanische en spierfysiologische werkingsprincipes van de SSC

Zowel bij een CMJ als een DJ is er sprake van een *countermovement*: een excentrische contractie van spieren rond de knie, heup en enkel tijdens de inveer-fase, gevolgd door een concentrische contractie in de fase waarin de springer weer opstrekt. Op spierniveau wordt dit ook wel een *stretch shortening cycle* (SSC) genoemd.

Het doorlopen van een SSC heeft een positief effect op de sprongprestatie. Met bijvoorbeeld een *squat jump* (sprong waarbij vanuit de *squat* houding – kniebuiging - zonder inverting gesprongen wordt) springt iemand lager dan met een CMJ (Linthorne, 2001, Bobbert, 1996). De tijdsduur van de SSC bij een DJ of een CMJ komt overeen met de duur van de afzetfase. Een kortere afzetfase komt dus overeen met een sneller doorlopen SSC.

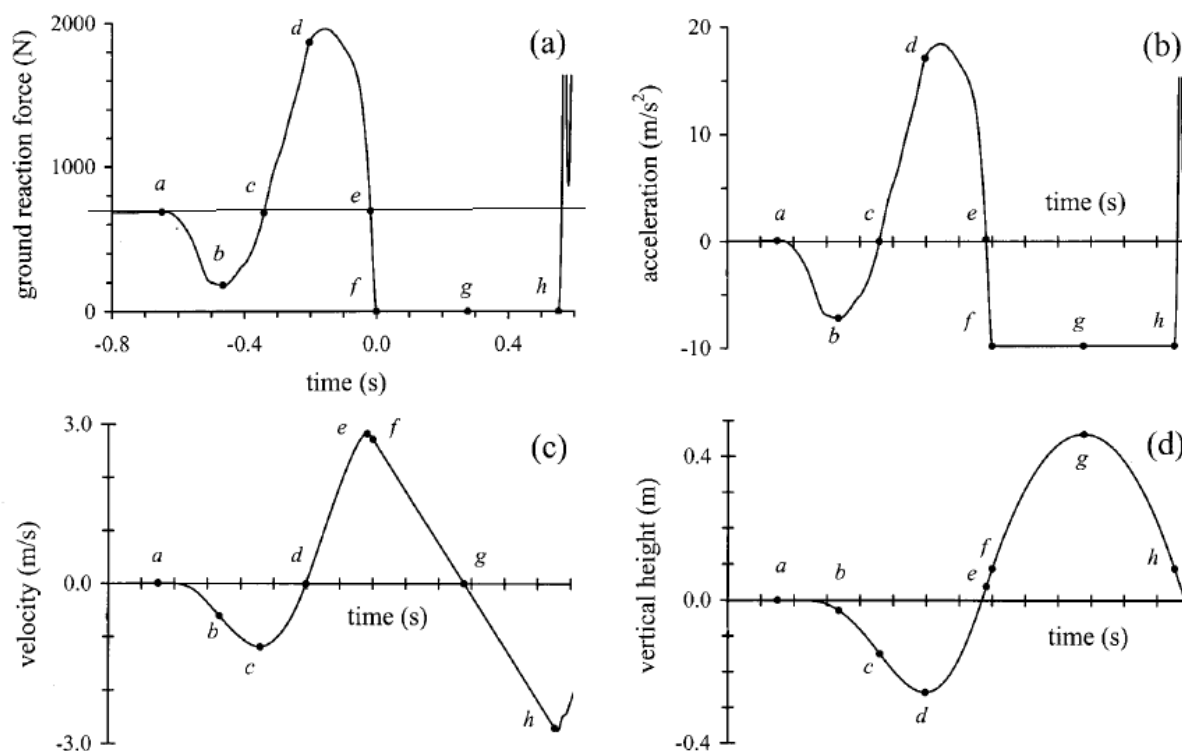
Flanagan & Comyns (2008) suggereren dat de SSC gecategoriseerd kan worden in een snelle SSC en een langzame SSC. De snelle SSC is te herkennen aan een korte contacttijd ( $< 0.25$  seconde) en een kleine hoekstandsverandering van de heup-, knie- en enkelgewrichten. Een DJ (kaatssprong) is een voorbeeld waar een snelle SSC van toepassing is. Een langzame SSC omvat langere contacttijden en grotere hoekstandsveranderingen van bovenstaande gewrichten. Deze komt voor bij bijvoorbeeld een CMJ.

Het feit dat een CMJ tot een hogere sprongprestatie leidt dan een *squat jump*, heeft twee oorzaken:

1. Verhoging van de voorspanning op de spieren (Bobbert, 1996)
2. Het effect van de *Residual Force Enhancement* (RFE) (Minozzo & Barbosa de Lira, 2013).

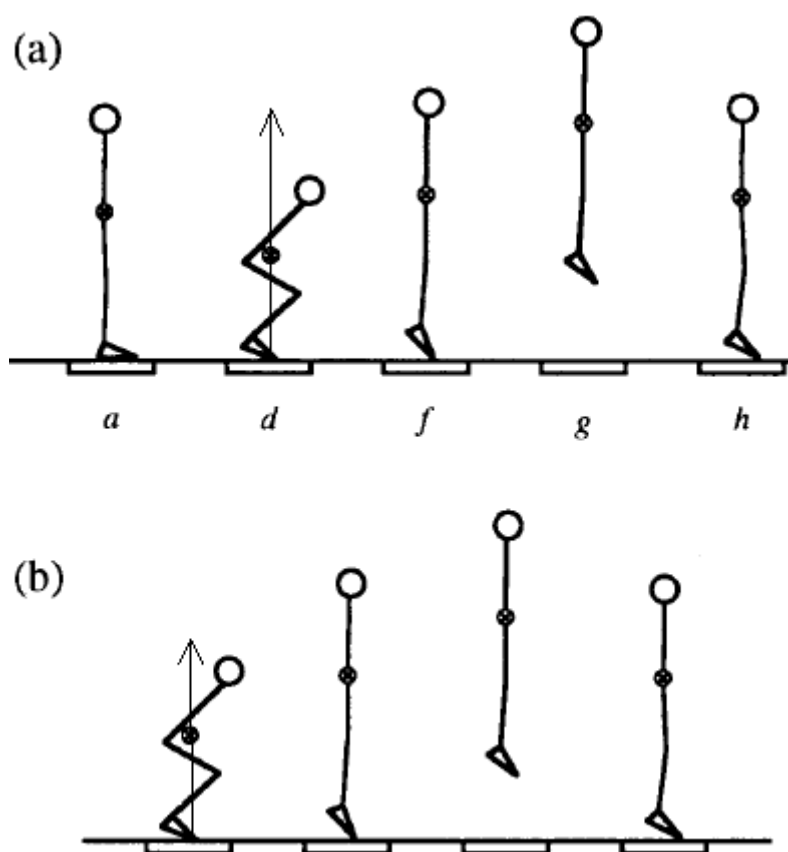
**ad 1** Het verhogen van de voorspanning op de spieren.

Tijdens de eerste fase van het inveren (van punt a tot punt c in Figuur 10a) is de resulterende kracht op het lichaam (*ground reaction force* min de zwaartekracht op het lichaam) neerwaarts gericht. Dit leidt tot een neerwaartse (negatieve) versnelling van het lichaam (van punt a t/m punt c in Figuur 10b). Zodra de resulterende kracht op het lichaam weer positief wordt (de GRF is nu dus groter dan de zwaartekracht op het lichaam), ondervindt de springer een opwaartse versnelling (vanaf punt c in Figuur 10a en 10b). Deze opwaartse versnelling zorgt echter niet direct voor een opwaartse beweging van het lichaam, maar leidt in eerste instantie tot een afname van de neerwaartse snelheid (zie Figuur 10c). Dit afremmen duurt voort totdat de springer het diepste punt van de inveerfase heeft bereikt (positie d in Figuur 10a,b c en d). Op het diepste punt van de inveerfase is er dus sprake van een GRF die aanzienlijk hoger is dan het lichaamsgewicht van de springer.



Figuur 10 Dynamische en kinematische curves van een CMJ. (a) grond reactiekracht curve, (b) versnelling-tijd curve, (c) snelheid-tijd curve, (d) verticale verplaatsing-tijd curve. Horizontale lijn 1a is de GRF door het lichaamsgewicht. Overgenomen uit: *Analysis of standing vertical jumps using a force platform*. Door Nicholas P. Linthorne, 2001, School of Exercise and Sport Science, The University of Sydney, Australia.

Stel dat de springer vanuit precies deze starthouding (eerste houding in Figuur 11b) een *squat jump* zou uitvoeren. In dat geval vertrekt hij vanuit stilstand en zal de GRF dus *gelijk* zijn aan zijn lichaamsgewicht. Bij een CMJ (Figuur 11a) is de GRF groter dan het lichaamsgewicht (verticale pijlen in Figuur 11). Dit leidt ertoe dat de snelheid, waarmee de gewrichten zullen gaan roteren zodra de springer zich opricht bij de uitvoering van een CMJ, direct veel hoger zal zijn dan bij de *squat jump*. De grotere kracht gaat gepaard met een groter moment rond de knie-, heup- en enkelgewrichten en dus een grotere voorspanning op de strekkers (Lagerberg & Faber, 2007).



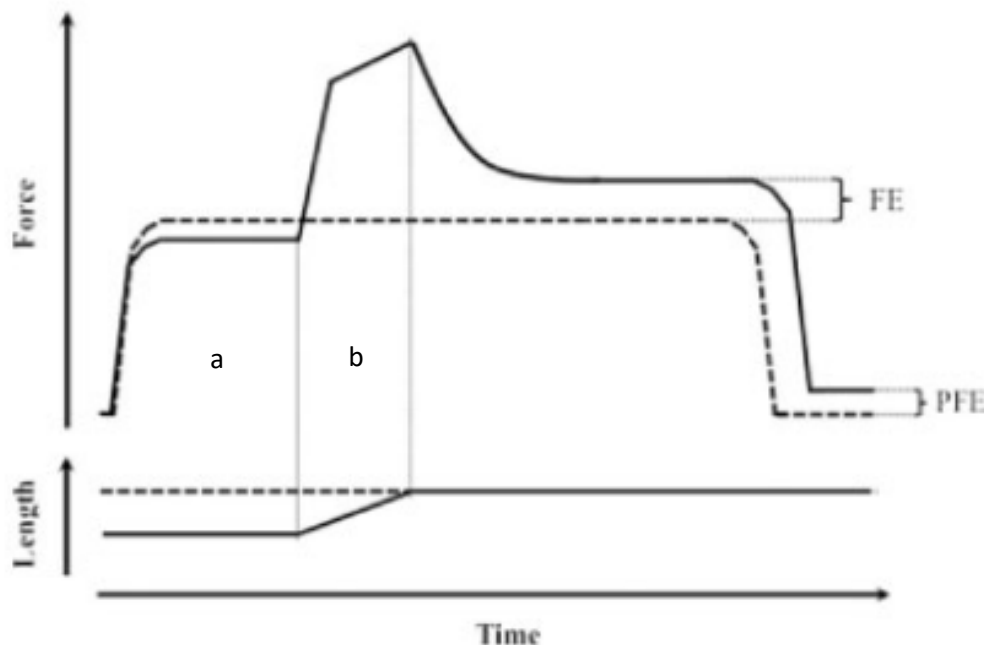
Figuur 11 Schematische weergave van een counter movement jump (a) en een squat jump (b). De letters onder de houding bij de weergave van de CMJ komen overeen met de positie in de curves van Figuur 10. De verticale lijn representeert de bijbehorende grondreactiekracht. Overgenomen uit: *Analysis of standing vertical jumps using a force platform*. Door Nicholas P. Linthorne, 2001

## ad 2 Het effect van *Residual Force Enhancement* (RFE).

Wanneer een actieve spier wordt verlengd en de spier vervolgens (terwijl de spier actief blijft) weer terugkeert naar zijn oorspronkelijke lengte, zal de kracht die de spier bij die lengte levert hoger zijn dan wanneer de spier bij deze lengte statisch wordt geactiveerd, dus zonder voorgaande verlenging (zie Figuur 12) (Minozzo & Barbosa de Lira, 2013).

De onderste grafiek in Figuur 12 geeft de lengte van de spier weer, de bovenste geeft de kracht weer. Beide boven elkaar afgebeelde grafieken tonen een contractie van dezelfde spier onder verschillende omstandigheden. De gestippelde lijn representeert een spier die een isometrische contractie vertoont en dus een constante lengte en een constante kracht heeft. De doorgetrokken lijn geeft een spier weer die aanvankelijk bij een kortere lengte geactiveerd wordt (fase a in Figuur 12) en daarbij minder kracht levert (door een minder gunstige overlap tussen actine en myosine). Vervolgens wordt de spier verlengd (er is dus sprake van een excentrische contractie). In deze excentrische fase (b in Figuur 12) levert de spier meer kracht dan de isometrisch contraherende spier. De verlenging van de actieve spier stopt zodra de lengte is bereikt van de (gestippeld weergegeven) isometrisch contraherende spier. Dan gaat de excentrische contractie weer over in een statische contractie maar levert de spier toch meer kracht dan de statische contraherende spier die geen excentrische contractie heeft doorgemaakt (de gestippelde lijn). Dit verschijnsel heet de (*Residual*) *Force Enhancement* (FE). Ook wanneer de spier niet meer actief is en passief verlengd wordt blijkt dat de

weerstand hoger is, wanneer deze eerst actief excentrisch is verlengd. Dit verschijnsel wordt de *Passive Force Enhancement* (PFE) genoemd.



Figuur 12 Schematische weergave van de Residual Force Enhancement. Beide boven elkaar afgebeelde grafieken tonen een contractie van de zelfde spier wanneer deze voor het eerst wordt aangespannen, uitgerekt tot eenzelfde eindlengte, ontspannen (lijn) en vervolgens isometrisch gecontraheerd en op dezelfde uiteindelijke lengte gehouden voordat ze weer worden ontspannen (streepjes lijn). Bovenste grafiek: kracht vs tijd. Onderste grafiek lengte van de spier vs tijd. FE: Force Enhancement. PFE: Passive Force Enhancement. Overgenomen uit *Muscle Residual Force Enhancement: a brief review*. Door Minozzo & Barbosa de Lira, 2013, McGill University, Montreal, Canada.

Volgens Minozzo & Barbosa de Lira (2013) is er een drietal hypothesen om dit fenomeen te verklaren. In de eerste hypothese wordt het lengte verschil van sacromeren gebruikt om de *force enhancement* te verklaren. De sacromeren hebben niet allemaal dezelfde lengte. Wanneer een actieve spier verder opgerekt wordt, zullen de sacromeren verlengen. Sommige sacromeren verlengen meer en worden daardoor zwakker door een afname in filament overlap. Andere worden sterker door toename van filament overlap. De zwakke sacromeren blijven opgerekt worden tot ze op een punt helemaal geen *cross-bridges* meer hebben en dus geen kracht meer kunnen leveren. Binnen deze theorie verklaart men de *force enhancement* doordat de excentrische verlenging zou zorgen voor meer lange sacromeren met een betere overlap en dus een sterkere spier. De tweede hypothese berust op de toename in sterkte van passieve elastische elementen in de spier (titine). Titine is de verbinding tussen het myosine-uiteinde en de actine. Naarmate sacromeren verder zijn uitgerekt wordt de spanning in de titine groter wat mogelijk stijfheid veroorzaakt. Dit zorgt dus mogelijk voor een FE. De laatste hypothese om de RFE te verklaren is een toename van het aantal gevormde *cross-bridges* en de duur hiervan. Joumaa et al (2008) zeggen echter dat de eerste hypothese onwaarschijnlijk is aan de hand van onderzoek dat ze hebben gedaan. Daaruit bleek dat de *force enhancement* niet afhankelijk is van een lengte verschil van sacromeren. Zij zeggen dat titine de verklaring vormt voor de *force enhancement*.

Bovenstaande inzichten in de mechanismen achter de ssc worden in het hoofdstuk advies betrokken bij de aanbevelingen

### 3.3 RSI<sub>mod</sub> of vermogen?

De RSI en de RSI<sub>mod</sub> zijn beide uitkomstmaten die verwantschap vertonen met het geleverde vermogen tijdens een sprong. Dit valt als volgt in te zien:

Naast de impulsmethode die in het eerste deel van dit rapport is besproken kan er ook op een tweede manier de spronghoogte worden bepaald. Deze tweede methode is het berekenen van de arbeid die de springer levert (Formule 3). De arbeid voortkomend uit de *Ground Reaction Force* (GRF) wordt berekend door het oppervlakte onder de kracht-verplaatsingscurve te bepalen door numeriek te integreren. Vervolgens wordt de arbeid van het lichaamsgewicht daarvan afgetrokken. Daarmee ontstaat Formule 4. Nu de snelheid op *toe-off* bekend is, kan deze weer worden ingevuld in Formule 2 om de spronghoogte te berekenen.

Formules

$$\int_{y_i}^{y_{to}} F_{GRF} dy - \int_{y_i}^{y_{to}} mg dy = W_{GRF} - W_{BW} = \frac{1}{2}mv_{to}^2 \quad (3)$$

$$\sqrt{\frac{W}{\frac{1}{2}m}} = v_{to} \quad (4)$$

$$P = F \cdot v \quad (5)$$

Aan die laatste methode is te zien dat de spronghoogte dus bepaald wordt door de hoeveelheid arbeid die de springer levert en niet door de snelheid waarmee hij die arbeid levert (vermogen).

Bij de RSI wordt de spronghoogte (die dus afhankelijk is van de geleverde arbeid tijdens de afzet) gedeeld door de tijd dat de afzet duurt. Die uitkomst is dus verwant aan het vermogen dat de springer levert. Als inzicht gewenst is in de explosiviteit van een springer (en dat is de bedoeling van de RSI) lijkt het berekenen van het vermogen en daarmee samenhangende uitkomsten als piek- en gemiddeld vermogen dus een veel directere maat dan de RSI. Het voordeel van de RSI is echter wel dat hij het met slechts een contactmat bepaald kan worden, maar voor de RSI<sub>mod</sub> is een *forceplate* nodig.

Het vermogen dat gegenereerd wordt tijdens een sprong staat gelijk aan het product van de gegenereerde kracht en de snelheid waarmee de kracht gegenereerd wordt, zoals te zien in Formule 5. Het vermogen is ook te berekenen als de eerste afgeleide van de geleverde arbeid. Indien de kracht in Formule 5 de GRF is, wordt het werkelijk geleverde vermogen berekend. Om atleten onderling beter te kunnen vergelijken heeft echter het voor lichaamsgewicht gecorrigeerde vermogen de voorkeur. Bij beschikbaarheid van een *forceplate* kan het geleverde (piek-)vermogen tijdens een sprong nauwkeurig worden bepaald.

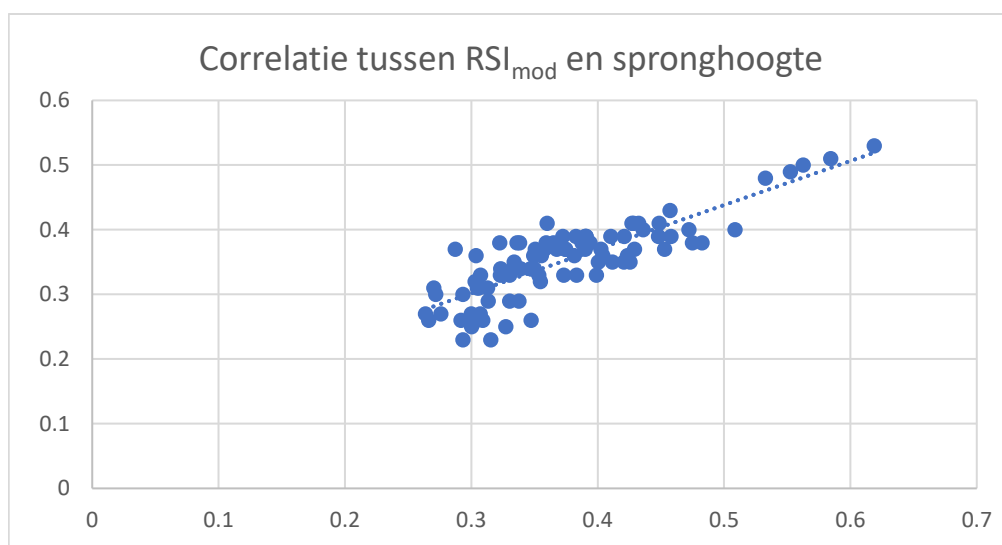
### 3.4 De meerwaarde van RSI<sub>mod</sub> of vermogensbepalingen

Een andere vraag die gesteld kan worden is welke extra informatie haalt de trainer of coach uit de berekende RSI<sub>mod</sub>. Indien de uitkomst van de spronghoogte sterk zou correleren met de berekende RSI geeft het de trainer weinig mogelijkheden tot het maken van een onderscheid tussen de diverse atleten en daarmee ook weinig aanknopingspunten voor specifieke training.

Omdat de RSI<sub>mod</sub> zoals zojuist genoemd verwant is met het vermogen, is er in het script eens stuk code toegevoegd waarmee de voor het lichaamsgewicht gecorrigeerde gemiddelde vermogen berekend wordt. De reden dat het voor het lichaamsgewicht gecorrigeerd wordt, is dat de atleten nu met elkaar vergeleken kunnen worden.

Om dit te onderzoeken zijn de op de resultaten van de op de eerste testdag uitgevoerde sprongen gebruikt. Om deze resultaten visueel te krijgen zijn de resultaten van de gemiddeldes en standaarddeviaties van alle sprongen (90) van alle atleten (18) weergegeven in Tabel 1. Bijlage 2 geeft de informatie over alle (vijf) sprongen per atleet.

Uit Tabel 1 blijkt dat een hoge  $RSI_{mod}$  vaak met een hoge spronghoogte en een lage  $RSI_{mod}$  vaak met een lage spronghoogte gepaard gaat. Hierdoor lijkt het dat de  $RSI_{mod}$  weinig nieuwe informatie geeft ten opzichte van de spronghoogte. Om hier meer inzicht in te krijgen, is de correlatie bepaald tussen de spronghoogte en de  $RSI_{mod}$ . Deze correlatie is met behulp van een scatterplot weergegeven in Figuur 13,  $r = 0.84$ . Bij deze correlatie zijn alle spronghoogtes en  $RSI_{mod}$  waarden betrokken (vijf per atleet in totaal 90 sprongen).

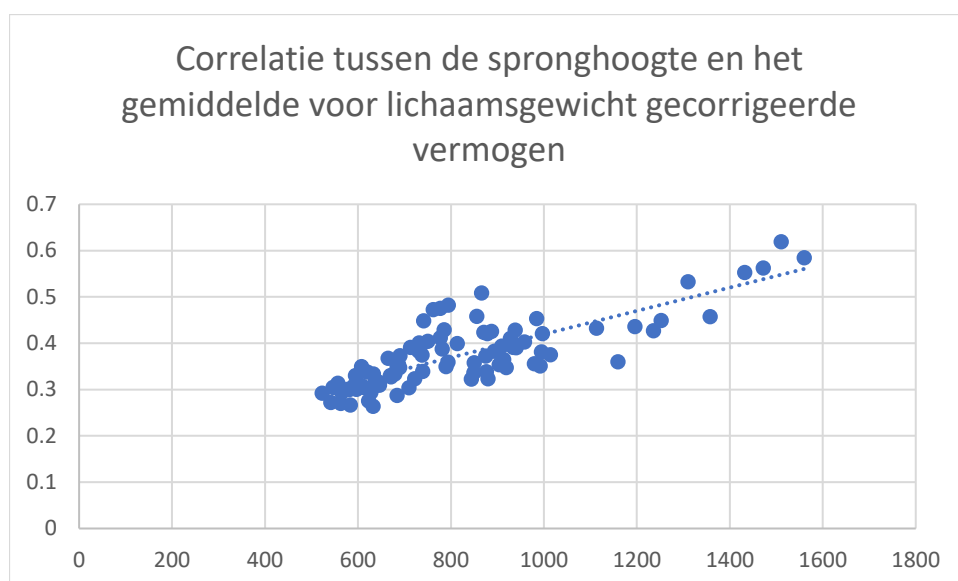


Figuur 13 Scatterplot die de correlatie weergeeft tussen de  $RSI_{mod}$  en de spronghoogte.  $r = 0.84$

Uit de correlatie van 0.84 blijkt dat de spronghoogte en de  $RSI_{mod}$  een sterk verband hebben met elkaar. De  $RSI_{mod}$  levert derhalve niet zoveel extra informatie op ten opzichte van de spronghoogte. Dit laatste geldt overigens ook voor het gemiddelde geleverde vermogen (zie figuur 14) de correlatie met spronghoogte is 0.8.

Ook deze correlatie is dus behoorlijk hoog. En de informatie die de  $RSI_{mod}$  bevat over de SSC blijft beperkt tot de uitvoeringssnelheid.

Het nadeel van de  $RSI_{mod}$  en ook van het gemiddelde concentrische vermogen, is dat het geen inzicht geeft in de effectiviteit van de *countermovement* tijdens sprongprestatie. De beschikbaarheid van een *forceplate* geeft hier echter wel aanvullende mogelijkheden die in het onderstaande advies worden toegelicht.



Figuur 14 Scatterplot die de correlatie weergeeft tussen de spronghoogte en het gemiddelde voor lichaamsgewicht gecorrigeerde vermogen.  $r = 0.80$

### 3.5 Het advies

Zoals hiervoor werd uitgelegd is voorspanning één van de elementen die bijdraagt aan het effect van de SSC van de CMJ. Met behulp van een forceplate kan deze voorspanning berekend worden. Door de GRF te bepalen op het diepste punt van het inveren, kan de procentuele toename van de GRF (ten opzichte van het lichaamsgewicht) berekend worden. Met die uitkomstmaat kan de mate waarin een springer voorspanning kan genereren onderling vergeleken worden. (Formule 6). Door de procentuele toename te gebruiken in de analyse wordt voor het verschil in lichaamsgewicht tussen de atleten gecorrigeerd.

Formule:

$$\text{Procentuele toename GRF} = \frac{\text{GRF}_{\text{diepstepunt}}}{\text{BW} \cdot g} \cdot 100 \quad (6)$$

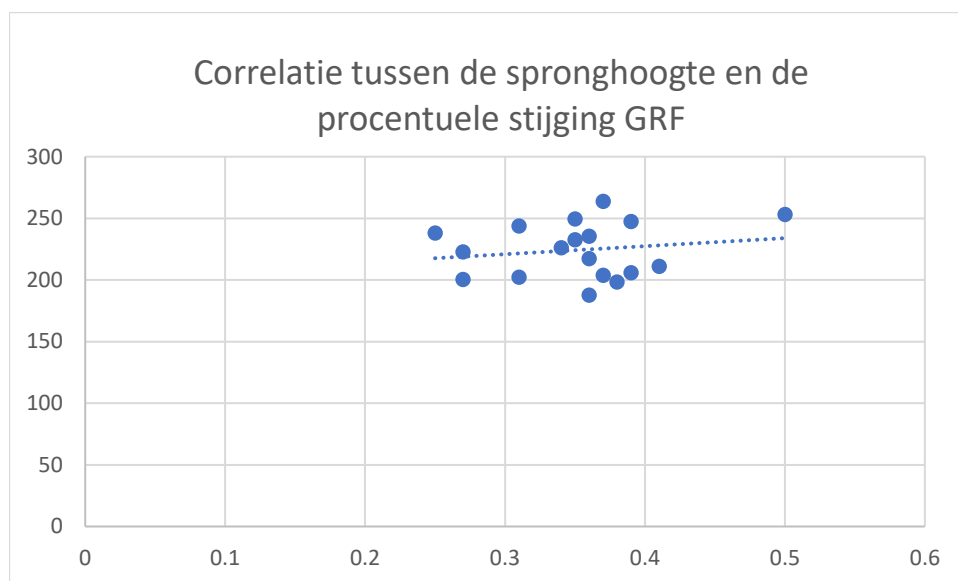
Voor de sprongen die tijdens de eerst BTN meting gedaan zijn, is de procentuele toename bepaald zoals weergegeven in Tabel 2. Hiervoor is een extra stuk code geschreven dat terug te vinden is in sectie 9b van het script.

Zoals uit Tabel 2 blijkt zijn er grote verschillen in de voorspanning die de springers genereren. Deze voorspanning blijkt niet sterk gecorreleerd aan de spronghoogte ( $r = 0.17$ , Figuur 15). Maar omdat de voorspanning iets zegt over de uitvoering van de *countermovement* en dus de SSC, is het wel te verwachten dat voorspanning zal correleren met het verschil in spronghoogte tussen een SJ en CMJ. Want de verwachting is dat dit verschil groter zal zijn naarmate de scc effectiever wordt doorlopen.

Springer	Spronghoogte (m)	%stijging GRF
1	0.35±0.03	249.49
2	0.36±0.01	187.67
3	0.36±0.03	217.13
4	0.39±0.01	205.72
5	0.38±0.01	198.2
6	0.50±0.02	252.96
7	0.37±0.03	263.74
8	0.37±0.02	203.65
9	0.41±0.01	210.98
10	0.36±0.01	235.42
11	0.25±0.01	237.86
12	0.35±0.02	232.52
13	0.27±0.01	200.24
14	0.27±0.02	222.74
15	0.34±0.01	226.11
16	0.31±0.01	202.09
17	0.31±0.02	243.57
18	0.39±0.01	247.22

Tabel 1 Weergave van de gemiddelde spronghoogte en procentuele stijging GRF.





Figuur 15 Scatterplot die de correlatie weergeeft tussen de procentuele stijging in GRF en de spronghoogte.  $r = 0.17$

De volgende stap in de vervolgstudie moet dan zijn het bepalen in hoeverre deze voorspanning correleert met de toename van de spronghoogte (CMJ versus SJ). Als deze correlatie sterk blijkt kan met behulp van een *forceplate* een feedbacktool worden ontwikkeld waarmee specifiek getraind kan worden op het verhogen van de bijdrage van de *countermovement*.

Door een applicatie te ontwikkelen die live de data kan analyseren zou na elke sprong de procentuele stijging van de GRF weergegeven kunnen worden. Hiermee kan tijdens de meting, direct na elke sprong, feedback gegeven worden of een sprong goed (hoge procentuele stijging GRF) of minder goed (minder hoge procentuele stijging GRF) was. Daarmee krijgt de atleet direct inzicht in welke sprong goed of minder goed was. Vervolgens kan wellicht een individueel trainingsprogramma worden opgesteld om het rendement van de *countermovement* te vergroten.

## Referenties

Flanagan, E & Comyns, T. (2008). The use of contact time and the reactive strength index to optimize fast stretch-shortening cycle training. *National Strength and conditioning association*, 30(5).

Bobbert, M, Gerritsen, K, Litjens M & Van Soest, A. (1996). Why is countermovement jump height greater than squat jump height? *Med. Sci. Sports. Exerc.*, 28(11), pp.1402-1412.

Joumaa, V, Leonard, T & Herzog, W. (2008). Residual force enhancement in myofibrils and sarcomeres. *Proceedings of the royal society B*. 275(1641), DOI: 10.1098/rspb.2008.0142

Lagerberg, A & Faber, H. (2007). Springen en armzwaaien. *Versus, tijdschrift voor fysiotherapie*. 25(5), pp.235-255.

Linthorne, N. (2001). Analysis of standing vertical jumps using a force platform. *American association of physics teachers*, 69(11). DOI: 10.1119/1.1397460.

Martinez, D. (2016). From the Field-directed topic the use of reactive strength index, reactive strength index modified, and flight time:contraction time as monitoring tools. *Journal of Australian Strength and Conditioning*, 24(5), pp.37-41.

Minozzo, F & Barbosa de Lira, C. (2012). Muscle residual force enhancement: a brief review. *Clinics*, 68(2), pp.269-274. DOI:10.6061/clinics/2013(02)R01.



```
numbers = result.numbers;

% Detected commas in non-thousand locations.
invalidThousandsSeparator = false;
if numbers.contains(',')
    thousandsRegExp = '^\\d+?(\\.|\\d{3})*\\.\\{0,1}\\d*$';
    if isempty(regex(numbers, thousandsRegExp, 'once'))
        numbers = NaN;
        invalidThousandsSeparator = true;
    end
end

% Convert numeric text to numbers.
if ~invalidThousandsSeparator
    numbers = textscan(char(strrep(numbers, ',', '')), '%f');
    numericData(row, 1) = numbers{1};
    raw{row, 1} = numbers{1};
end
catch
    raw{row, 1} = rawData{row};
end
end

%% Replace non-numeric cells with NaN
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Find non-numeric cells
raw(R) = {NaN}; % Replace non-numeric cells

%% Create output variable

data = table;
data.General_Fz = cell2mat(raw(:, 1));
data = data.General_Fz(18:end);

else
    % Measure in Gymzaal van de toekomst
    %% Read columns of data as text:
    % For more information, see the TEXTSCAN documentation.
    formatSpec =
'%*s*s*s*s*s*s*s*s*s*s*s*s*s*s*s*s*s*s[s^\\n\\r]';

%% Open the text file.
fileID = fopen(fileName,'r');

%% Read columns of data according to the format.
% This call is based on the structure of the file used to generate this
% code. If an error occurs for a different file, try regenerating the code
% from the Import Tool.
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter,
'TextType', 'string', 'ReturnOnError', false);

%% Close the text file.
fclose(fileID);

%% Convert the contents of columns containing numeric text to numbers.
% Replace non-numeric text with NaN.
raw = repmat({''},length(dataArray{1}),length(dataArray)-1);
for col=1:length(dataArray)-1
```

```

        raw(1:length(dataArray{col}),col) = mat2cell(dataArray{col},
ones(length(dataArray{col}), 1));
end
numericData = NaN(size(dataArray{1},1),size(dataArray,2));

% Converts text in the input cell array to numbers. Replaced non-numeric
% text with NaN.
rawData = dataArray{1};
for row=1:size(rawData, 1)
    % Create a regular expression to detect and remove non-numeric prefixes
    and
    % suffixes.
    regexstr = '(?<prefix>.*?)(?<numbers>([-
]*(\d+[\,\,]*)+[\.\,]{0,1}\d*[eEdD]{0,1}[-+]*\d*[i]{0,1})|([-
]*(\d+[\,\,]*)*[\.\,]{1,1}\d+[eEdD]{0,1}[-+]*\d*[i]{0,1})) (?<suffix>.*?);
    try
        result = regexp(rawData(row), regexstr, 'names');
        numbers = result.numbers;

        % Detected commas in non-thousand locations.
        invalidThousandsSeparator = false;
        if numbers.contains(',')
            thousandsRegExp = '^(\d+?(\,\,\d{3}))*\.\{0,1\}\d*$';
            if isempty(regexp(numbers, thousandsRegExp, 'once'))
                numbers = NaN;
                invalidThousandsSeparator = true;
            end
        end
        % Convert numeric text to numbers.
        if ~invalidThousandsSeparator
            numbers = textscan(char(strrep(numbers, ',', '')), '%f');
            numericData(row, 1) = numbers{1};
            raw{row, 1} = numbers{1};
        end
    catch
        raw{row, 1} = rawData{row};
    end
end

%% Replace non-numeric cells with NaN
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Find non-numeric
cells
raw(R) = {NaN}; % Replace non-numeric cells

%% Create output variable
data = cell2mat(raw);
%% Clear temporary variables
clearvars filename delimiter formatSpec fileID dataArray ans raw col
numericData rawData row regexstr result numbers invalidThousandsSeparator
thousandsRegExp R;
%%
data = data(28:end);

end
%% Clear temporary variables
clearvars filename delimiter formatSpec fileID dataArray ans raw col
numericData rawData row regexstr result numbers invalidThousandsSeparator
thousandsRegExp R;
%% Variables

```

```

dt = 0.005;
g = 9.81;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Select jump's
H_Fig = figure(1);
set(H_Fig, 'units', 'normalized', 'outerposition', [0 0 1 1])
plot (data)
[x,y] = ginput(10);
x = round(x);
y = round(y);
close all
%% Plot selected jump's
%%
% sprong.trail{1}(4) inhoud van cell tussen {} en waarde van inhoud ().
j = 1;
for i = 1:2:9
    sprong.trail{j} = data(x(i):x(i+1));
    j=j+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% look for <4 because when in the air the forceplate will give a value
between 4 and minus 4
for i = 1:5
    sprong.flight{i} = find(sprong.trail{i}<4);

    for j = 1:length(sprong.flight{i})-1;
        verschil = sprong.flight{i}(j+1)-sprong.flight{i}(j);
    end

    sprong.To{i} = sprong.flight{i}(1);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Find motionless standing section
j = 1;
for i = 1:(1/dt):length(data)-(1/dt)
    standev(j) = std(data(i:i+(1/dt)));
    j=j+1;
end
[Valdev, Indexdev] = min(standev);

FBW = mean(data((Indexdev-1)*(1/dt)+1:Indexdev*(1/dt)));
BW = FBW/g;
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% **peakprominence: zijn de waarden naast de peak groot genoeg om het
juiste piekje te laten zien
%
for i = 1:5
    sprong.cut{i} = sprong.trail{i}(1:sprong.To{i});
    % Flip het signaal, en zoek de piek die minimaal 0,9* BW is. met
    % prominence zegt die dat het signaal links en rechts van de piek
    % minimaal 60 lager moet zijn
    [sprong.peakVal{i}, sprong.peakIndex{i}] = findpeaks(sprong.cut{i}*-
    1, 'MinPeakHeight', -FBW*0.9, 'MinPeakProminence', 60);
%

```

```

emptycell = isempty(sprong.peakIndex{i});
if emptycell == 1
    [Index,Val] = find(sprong.trail{i}<4);
    sprong.peakIndex{i} = Index(1);
    k = 1;
    for j = 1:length(Index)-1;
        difference = Index(j+1)-Index(j);
        %zoeken naar wanneer het aantal samples tussen het dal en de flight
        %fase groot genoeg is om de flight fase te herkennen
        if difference > 20
            different{k} = Index(j+1);
            k = k+1;
        end
    end
    sprong.To{i} = cell2mat(different(1));
    sprong.cut{i} = sprong.trail{i}(1:sprong.To{i});
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% knip signaal van begin tot eerste dal
for i = 1:5
    sprong.dalcut{i} = sprong.trail{i}(1:sprong.peakIndex{i}(1));

    sprong.grFBW{i} = find(sprong.dalcut{i}>FBW-1);
    sprong.start{i} = sprong.grFBW{i}(end);
    sprong.jumpsamples{i} = sprong.To{i} - sprong.start{i};
    sprong.TTT{i} = sprong.jumpsamples{i}*dt;
end

figure
for i = 1:5
    subplot(1,5,i)
        plot(sprong.trail{i})
        hold on
        % x coördinaat is sprong TO, y coördinaat is sprong trail op het punt
        % van to off

    line([sprong.To{i},sprong.To{i}],[sprong.trail{i}(sprong.To{i}),sprong.trail{i}(sprong.To{i})], 'marker', 'o', 'color', 'r')

    line([sprong.start{i}(1),sprong.start{i}(1)],[sprong.trail{i}(sprong.start{i}(1)),sprong.trail{i}(sprong.start{i}(1))], 'marker', 'o', 'color', 'g')

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%spronghoogte bepaling
for i = 1:5
    sprong.cutF{i} = sprong.cut{i}-FBW;
    sprong.inte{i} = cumtrapz(sprong.cutF{i})*dt;
    sprong.J{i} = sprong.inte{i}(end);
    sprong.vTO{i} = sprong.J{i}/(FBW/g);
    sprong.height{i} = sprong.vTO{i}^2/(2*g);
end

for i = 1:5
    sprong.RSImod{i} = sprong.height{i}/sprong.TTT{i};
end

```

```

DataOut(i,1) = sprong.height{i}/sprong.TTT{i};
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Eccentric starts when V is at minimum and stops when V hits Zero, then
% concentric phase starts
for i = 1:5
sprong.acc{i} = sprong.cutF{i}/BW;
sprong.velo{i} = cumtrapz(sprong.acc{i}*dt);
%Vpeak is lowest point in v-curve// start of concentric phase
[sprong.VpeakVal{i},sprong.VpeakIndex{i}] = min(sprong.velo{i});
% Start Eccentric phase
sprong.startEcc{i} = sprong.VpeakIndex{i};
% End Eccentric phase
sprong.endEcc{i} = find(sprong.velo{i}<0);
sprong.endEcc{i} = sprong.endEcc{i}(end);
% Start Concentric
sprong.startCon{i} = sprong.endEcc{i}+1;
% Length of phase(samples)
sprong.lengthEcc{i} = (sprong.endEcc{i}(1)-sprong.startEcc{i}(1));
sprong.lengthCon{i} = (sprong.To{i}(1)-sprong.startCon{i}(1));
end
%%
for i=1:5
    sprong.trailcorr{i} = sprong.trail{i}-FBW;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 9a %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Mean velocity
for i = 1:5
sprong.meanEccVelo{i} =
mean(sprong.velo{i}(sprong.startEcc{i}:sprong.endEcc{i}));
sprong.meanConVelo{i} =
mean(sprong.velo{i}(sprong.startCon{i}:sprong.To{i}));
end
%% Mean Concentric Power (P=F*V) + Concentric Peak Power
for i = 1:5
    sprong.meanConP{i} =
mean(sprong.trailcorr{i}(sprong.startCon{i}:sprong.To{i})).*sprong.velo{i}(s
prong.startCon{i}:sprong.To{i}));
    sprong.ConP{i} =
sprong.trailcorr{i}(sprong.startCon{i}:sprong.To{i})).*sprong.velo{i}(sprong
.startCon{i}:sprong.To{i});
    sprong.ConPpeak{i} = max(sprong.ConP{i});
end

%% Mean Concentric Force
for i = 1:5
sprong.meanConF{i} =
mean(sprong.trailcorr{i}(sprong.startCon{i}:sprong.To{i}));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 9b %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% GRF at lowest point. en de berekening om de procentuele toenames
tenopzichte van BWGRF te bepalen
for i = 1:5
    GRF.lowestpoint{i} = sprong.trail{i}(sprong.endEcc{i});
    Procentueel.grf{i} = GRF.lowestpoint{i}/(BW*g/100);
end
for i = 1:5
    Average.ProcGRF{1} =
(Procentueel.grf{1}+Procentueel.grf{2}+Procentueel.grf{3}+Procentueel.grf{4
}+Procentueel.grf{5})/5
end

```



```

%% Averages
Average.Height{1} =
(sprong.height{1}+sprong.height{2}+sprong.height{3}+sprong.height{4}+sprong
.height{5})/5;
Average.TTT{1} =
(sprong.TTT{1}+sprong.TTT{2}+sprong.TTT{3}+sprong.TTT{4}+sprong.TTT{5})/5;
Average.RSiMod{1} =
(sprong.RSiMod{1}+sprong.RSiMod{2}+sprong.RSiMod{3}+sprong.RSiMod{4}+sprong
.RSiMod{5})/5;
Average.meanEccVelo{1} =
(sprong.meanEccVelo{1}+sprong.meanEccVelo{2}+sprong.meanEccVelo{3}+sprong.m
eanEccVelo{4}+sprong.meanEccVelo{5})/5;
Average.meanConVelo{1} =
(sprong.meanConVelo{1}+sprong.meanConVelo{2}+sprong.meanConVelo{3}+sprong.m
eanConVelo{4}+sprong.meanConVelo{5})/5;
Average.meanConF{1} =
(sprong.meanConF{1}+sprong.meanConF{2}+sprong.meanConF{3}+sprong.meanConF{4
}+sprong.meanConF{5})/5;
Average.meanConP{1} =
(sprong.meanConP{1}+sprong.meanConP{2}+sprong.meanConP{3}+sprong.meanConP{4
}+sprong.meanConP{5})/5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
for i = 1:5
DataOut(i,1) = round(sprong.height{i},2);
DataOut(i,2) = round(sprong.TTT{i},2);
DataOut(i,3) = round(sprong.RSiMod{i},2);
DataOut(i,4) = round(sprong.meanEccVelo{i},2);
DataOut(i,5) = round(sprong.meanConVelo{i},2);
DataOut(i,6) = round(sprong.meanConF{i},2);
DataOut(i,7) = round(sprong.meanConP{i},2);

end

Header = {'JumpHeight','Time to Takeoff','RsiMod','mean ecc V','mean con
V','mean con F','mean con P' };
xlswrite('RsiModResults.xlsx',Header,name,'A1');
xlswrite('RsiModResults.xlsx',DataOut,name,'A2');

Avg(1,1) = round(Average.Height{1},2);
Avg(1,2) = round(Average.TTT{1},2);
Avg(1,3) = round(Average.RSiMod{1},2);
Avg(1,4) = round(Average.meanEccVelo{1},2);
Avg(1,5) = round(Average.meanConVelo{1},2);
Avg(1,6) = round(Average.meanConF{1},2);
Avg(1,7) = round(Average.meanConP{1});

Header2 = {'Avg Height','Avg TTT','Avg RSiMod','Avg EccVelo','Avg
ConVelo','Avg ConF','Avg ConP'};
xlswrite('RsiModResults.xlsx',Header2,name,'A7');
xlswrite('RsiModResults.xlsx',Avg,name,'A8');

%% put al struct values in a table
T1 = table(sprong.height', sprong.TTT', sprong.RSiMod',
sprong.meanEccVelo', sprong.meanConVelo', sprong.meanConF',
sprong.meanConP');
T2 = table(Average.Height, Average.TTT, Average.RSiMod,
Average.meanEccVelo, Average.meanConVelo, Average.meanConF,
Average.meanConP);

```

```

T3 = table(std(cell2mat(sprong.height)), std(cell2mat(sprong.TTT)),
std(cell2mat(sprong.RSImod)), std(cell2mat(sprong.meanEccVelo)),
std(cell2mat(sprong.meanConVelo)), std(cell2mat(sprong.meanConF)),
std(cell2mat(sprong.meanConP)));
T = [T1;T2;T3];
T.Properties.VariableNames = {'Height' 'TTT' 'RSImod' 'meanEccVelo'
'meanConVelo' 'meanConF' 'meanConP' };

% row 7 is average row 8 is STD
fileName = strcat(name, '.xlsx');
writetable(T, fileName);
winopen(fileName)
fileName_template = 'results_template.xlsx';
copyfile(fileName_template, fileName);
writetable(T, fileName);
winopen(fileName);

```

## Bijlage 2: Sprongresultaten

springer	meanP	RSI	Hoogte
1	785.4301	0.429133	0.37
1	749.8041	0.40421	0.36
1	690.2509	0.373212	0.33
1	729.9962	0.383318	0.33
1	855.0526	0.458133	0.39
2	738.8601	0.33913	0.34
2	683.903	0.287324	0.37
2	781.0744	0.387612	0.38
2	709.6087	0.303829	0.36
2	737.5362	0.374592	0.37
3	632.9247	0.333966	0.35
3	664.4184	0.36776	0.37
3	713.1388	0.390912	0.39
3	546.2599	0.30499	0.31
3	732.9533	0.390124	0.37
4	934.179	0.390813	0.39
4	875.7201	0.338203	0.38
4	927.5613	0.410482	0.39
4	908.7811	0.393983	0.38
4	937.688	0.428605	0.41
5	849.6129	0.358149	0.37
5	940.1961	0.390275	0.39
5	912.9575	0.365052	0.38
5	849.4717	0.33627	0.38
5	892.9139	0.382857	0.39
6	1310.054	0.532829	0.48
6	1510.429	0.619035	0.53
6	1472.002	0.562626	0.5
6	1431.941	0.55254	0.49
6	1559.611	0.584541	0.51
7	957.6364	0.402603	0.37
7	996.5134	0.421061	0.39
7	1112.87	0.43244	0.41
7	903.6337	0.353626	0.33
7	879.5189	0.322986	0.33
8	843.517	0.322605	0.38
8	721.6042	0.323347	0.34
8	793.5582	0.359305	0.38
8	874.8396	0.372492	0.39
8	789.3051	0.349527	0.36
9	1235.243	0.427108	0.41
9	1195.665	0.436016	0.4
9	1159.039	0.35997	0.41
9	1252.161	0.448769	0.41
9	1357.694	0.45725	0.43
10	991.5808	0.350698	0.37

10	994.4989	0.381527	0.36
10	918.602	0.347225	0.34
10	1013.831	0.374894	0.37
10	979.5235	0.35553	0.36
11	646.1589	0.309152	0.26
11	627.4129	0.29358	0.23
11	645.401	0.315598	0.23
11	671.0955	0.327408	0.25
11	689.3107	0.347459	0.26
12	870.3716	0.423698	0.36
12	813.2191	0.399141	0.33
12	886.8822	0.425716	0.35
12	984.1464	0.452998	0.37
12	877.7892	0.420857	0.35
13	632.3664	0.26373	0.27
13	622.8947	0.27591	0.27
13	596.188	0.300351	0.25
13	583.1893	0.266414	0.26
13	609.3369	0.307116	0.27
14	619.9929	0.337854	0.29
14	522.7543	0.291921	0.26
14	594.1718	0.330511	0.29
14	580.8564	0.300143	0.27
14	610.18	0.304475	0.26
15	669.3224	0.330422	0.33
15	684.0965	0.347404	0.34
15	776.9417	0.411679	0.35
15	731.329	0.400659	0.35
15	686.3364	0.354872	0.32
16	564.4392	0.293389	0.3
16	557.6555	0.303096	0.32
16	562.6276	0.270425	0.31
16	588.738	0.306062	0.31
16	635.976	0.307393	0.33
17	541.1218	0.272006	0.3
17	607.5313	0.349885	0.34
17	556.6952	0.313391	0.29
17	635.4208	0.312899	0.31
17	679.1805	0.333468	0.34
18	865.6867	0.508871	0.4
18	776.6581	0.475132	0.38
18	793.7638	0.482585	0.38
18	761.7573	0.472374	0.4
18	741.3337	0.448359	0.39