

Technische Universiteit Delft  
Faculteit Elektrotechniek, Wiskunde en Informatica  
Delft Institute of Applied Mathematics

**Het detecteren van veranderingen in de prestatie van  
een wielrenner**

**(Engelse titel: The detection of changes in the  
performance of pro cyclists)**

Verslag ten behoeve van het  
Delft Institute of Applied Mathematics  
als onderdeel ter verkrijging

van de graad van

**BACHELOR OF SCIENCE  
in  
TECHNISCHE WISKUNDE**

door

**MARISKA POST-VREUGDENHIL**

**Delft, Nederland, juni 2016**

---

<sup>1</sup>Referentie afbeelding voorblad: <http://teamgiantalpecin.com/wallpapers>



## **BSc verslag TECHNISCHE WISKUNDE**

**“Het detecteren van veranderingen in de prestatie van een wielrenner”  
(Engelse titel: “The detection of changes in the performance of pro cyclists”)**

MARISKA POST-VREUGDENHIL

**Technische Universiteit Delft**

### **Begeleiders**

Dr. A.J. Cabo

A.H.M. van Erp, MSc

### **Overige commissieleden**

Dr. B. van den Dries

Dr.ir. M.C. Veraar

Juni, 2016

Delft



# Inhoudsopgave

<b>Samenvatting</b>	<b>vii</b>
<b>Voorwoord</b>	<b>ix</b>
<b>1 Inleiding</b>	<b>1</b>
<b>2 Het beschrijven van de data</b>	<b>3</b>
<b>3 Achtergrond - Energiesystemen</b>	<b>7</b>
3.1 ATP + CrP systeem . . . . .	7
3.2 Anaëroob lactisch systeem . . . . .	7
3.3 Aëroob oxidatief systeem . . . . .	7
<b>4 Het bepalen van een maat van prestatie</b>	<b>9</b>
4.1 Critical Power . . . . .	9
4.2 W'balans . . . . .	15
<b>5 Veranderingen detecteren - de theorie</b>	<b>21</b>
5.1 Changepoint Analyse . . . . .	21
5.2 Taut string methode . . . . .	23
5.3 Het vergelijken van de twee methoden . . . . .	26
5.4 Uitwerken van de methoden voor een kleine dataset . . . . .	30
<b>6 Veranderingen detecteren - de toepassing</b>	<b>33</b>
<b>7 Conclusies en aanbevelingen</b>	<b>39</b>
7.1 Conclusies . . . . .	39
7.2 Aanbevelingen . . . . .	40
<b>Referenties</b>	<b>41</b>
<b>A Bijlagen</b>	<b>43</b>
A.1 Algemene matlab code . . . . .	43
A.2 Matlab code voor hoofdstuk 2 . . . . .	45
A.3 Matlab code voor hoofdstuk 4 . . . . .	47
A.4 R code voor hoofdstuk 5 . . . . .	56
A.5 R code voor hoofdstuk 6 . . . . .	57



## Samenvatting

Het betrekken van de fysiologie bij de wiskundige modellering van prestatiedata van een wielrenner kan een zeer waardevolle stap zijn om veranderingen te detecteren in de prestatie van de wielrenner. De fysiologie beschrijft modellen omtrent verschillende energiesystemen in het menselijk lichaam, die in verschillende stadia van inspanning gebruikt worden. De anaërobe arbeidscapaciteit, een bepaalde energievoorraad die opgeslagen zit in het lichaam, wordt verbruikt op het moment dat er inspanning wordt geleverd boven de zogeheten critical power. Wanneer er inspanning wordt geleverd met een vermogen beneden deze critical power, vult deze voorraad energie in het lichaam zich weer aan. Het verloop van deze energievoorraad kan weergegeven worden door de  $W'$ balans. Door deze  $W'$ balans terug te modelleren over de gefietste vermogensdata voor een bepaalde dag, kan een maat van prestatie bepaald worden: de gefietste critical power,  $GCP$ . Deze gefietste critical power kan voor iedere dag waarvan data beschikbaar is, bepaald worden en op de tijdreeks van deze data kunnen methoden toegepast worden om veranderingen in deze maat van prestatie te detecteren. Methoden die deze veranderingen kunnen detecteren zijn de changepoint analyse en taut string methode. De veranderingen die door de methoden gedetecteerd zijn, kunnen vervolgens naast het logboek van de renner gelegd worden om te kijken of deze veranderingen verklaard kunnen worden. De methode die de veranderingen het nauwkeurigst meet, blijkt de changepoint analyse te zijn.





## Voorwoord

Dit verslag is geschreven als onderdeel van het Bachelor Eindproject van de studie Technische Wiskunde aan de TU Delft. Om eerlijk te zijn is het niet zo dat ik heel mijn bachelor al uit keek naar de dag dat dit project zou gaan beginnen. Het leek me lastig om tien weken lang met één onderwerp bezig te zijn. Aan het einde van deze tien weken kan ik het tegendeel echter bewijzen. Ik heb met veel plezier gewerkt aan mijn BEP en kan zo nog een aantal dingen bedenken die ik verder uit zou kunnen diepen. Het feit dat wielrennen een van mijn interesses is, heeft zeker bijgedragen aan mijn enthousiasme gedurende deze tien weken. Ik vond het heel erg leuk om de wiskunde die ik gedurende de opleiding heb geleerd, toe te passen en een model te ontwikkelen dat ook daadwerkelijk gebruikt kan worden.

Ik wil Annoesjka Cabo bedanken voor de begeleiding, ik vond het erg fijn om samen in de lastige stukjes te duiken, hier weer wijzer van te worden en iemand te hebben die waar nodig wat sturing gaf. Geurt Jongbloed, bedankt voor het opzetten van het project, de contactlegging met Team Giant-Alpecin en het meedenken. Ik ben Team Giant-Alpecin erg dankbaar voor het beschikbaar stellen van de data en Teun van Erp voor de intensieve begeleiding vanuit de ploeg. Daarnaast wil ik Mark Veraar en Bart van den Dries bedanken voor het aanvullen van mijn beoordelingscommissie en Plop voor het doorlezen van het concept en het geven van de feedback hierop. Tenslotte bedank ik natuurlijk Mark voor alle kopjes thee en de mentale ondersteuning gedurende het project.

Ik wens een ieder veel leesplezier.

Mariska Post-Vreugdenhil  
Delft, juni 2016



# 1 Inleiding

Sinds de afgelopen Giro d'Italia kent heel Nederland hem: Steven Kruijswijk. 5 dagen reed hij in de roze trui, de leiderstrui, en er was een mogelijkheid geweest om de ronde te winnen. Kruijswijk reed zo goed als nooit tevoren.

Toch zal een wielerkenner weten dat Kruijswijk vorig jaar in de ronde van Italië ook uitmuntend presteerde. In de eerste week verloor hij echter veel tijd, waardoor hij geen kandidaat meer was voor de eindzege, maar in de zware etappes kon hij met de besten omhoog fietsen. Was deze renner dit jaar echt zoveel beter dan vorig jaar?

Hetzelfde kunnen we ons afvragen bij een andere renner, Warren Barguil. In 2014 reed hij de ronde van Spanje, de Vuelta d'España, waar hij 8<sup>e</sup> werd in het eindklassement. De Tour de France die hij daarna, in 2015 reed, eindigde hij als 14<sup>e</sup>. Was hij tijdens de Tour de France slechter dan tijdens de Vuelta d'España? Of was de concurrentie beter? De bovenstaande anekdotes geven weer dat rituitslagen niet altijd een goed beeld geven van hoe goed een renner is op een bepaald moment.

In het geval van de bovenstaande renners kan het erg waardevol zijn om meer informatie te vergaren over de prestatie en de eventuele verandering die daarin is opgetreden. In de afgelopen tien jaar is er veel onderzoek gedaan naar de koppeling met de fysiologie in het kijken naar de data die iets kan zeggen over de prestatie van een wielrenner, denk aan hartslag, vermogen en snelheid. Recentelijk is het  $W'$ balans model gepresenteerd [24], dat het verloop van een bepaalde energievoorraad in het lichaam beschrijft. In het kader van dat model is de volgende onderzoeksvraag opgesteld:

Is het mogelijk om aan de hand van het  $W'$ balans model een maat van prestatie te bepalen voor een wielrenner en zijn er veranderingen te detecteren in deze maat van prestatie in verloop van de tijd?

Om antwoord te geven op deze vraag, wordt deze opgedeeld in deelonderwerpen die allen in een apart hoofdstuk aan bod komen. Hoofdstuk 2 beschrijft de data die wordt gemeten door de wielerploeg en de wijze waarop dit wordt gemeten. Ook wordt er meer statistisch naar de data gekeken, er worden eigenschappen bekeken en de data wordt getransformeerd om deze symmetrisch te maken. Het volgende hoofdstuk neemt een uitstapje naar de fysiologie. Om een goed beeld te krijgen van de wiskundige modellen die opgesteld kunnen worden omtrent vermogen, is het nodig om kennis te hebben van de drie verschillende energiesystemen in het lichaam, het ATP + CrP systeem, het anaëroob lactisch systeem en het aëroob oxidatief systeem. In hoofdstuk 4 wordt de maat van prestatie bepaald. Dit door eerst naar eigenschappen van plot die de tijd uitzet tegen het maximaal vermogen te kijken en de critical power en de anaërobe arbeid capaciteit te introduceren. Vervolgens wordt de de anaërobe arbeidscapaciteit voorgesteld als een batterij die kan leeglopen en opladen. Het verloop van deze batterij,  $W'$ balans, kan gemodelleerd worden. Het terugrekenen van dit model met behulp van de dagelijks beschikbare prestatiedata levert een maat van prestatie op, de gefietste critical power,  $GCP$ . Er zijn verschillende methoden die veranderingen in data kunnen detecteren. Deze methoden worden in hoofdstuk 5 beschreven. Zowel de changepoint analyse als de taut string methode worden uitgelegd en deze worden met elkaar vergeleken. Tenslotte worden in hoofdstuk 6 de methoden uit hoofdstuk 5 toegepast op de in hoofdstuk 4 bepaalde  $GCP$ 's om veranderingen te detecteren. In dit hoofdstuk wordt ook de koppeling gemaakt met de gegevens uit het logboek van de renner om inzicht te krijgen in de veranderingen die gedetecteerd zijn.



## 2 Het beschrijven van de data

De data voor dit project is aangeleverd door Team Giant-Alpecin, een Nederlands-Duitse wielploeg, waar onder andere Tom Dumolin, John Degenkolb, Laurens ten Dam en Warren Barguil voor rijden. Bij deze ploeg wordt veel data verzameld. Voor dit project is de data verkregen van vijf wielrenners, die in het vervolg renner 1, renner 2, renner 3, renner 4 en renner 5 genoemd worden.

Voor iedere dag waarop er gefietst is, is er per renner een dataset met een tabel daarin. Deze dataset heeft de naam van de renner, gevolgd door het nummer van de dag, met dag 1 is gelijk aan 30 oktober 2011. Iedere rij in de tabel stelt een meting voor en iedere kolom stelt een variable voor die gemeten wordt. In elke rij bevindt zich dus één output van elke variabele. Hieronder volgt een lijst met de 9 variabelen die gemeten worden, met bij iedere variabele een kleine uitleg.

De dataset bevat de volgende variabelen:

1. Tijdseenheid, de tijd van de meting, gemeten vanaf het moment dat het meetkastje aangezet werd, in minuten.
2. Torque, het krachtmoment van de kracht op de pedalen, in Newtonmeter.
3. Snelheid, de snelheid van de renner in kilometer per uur.
4. Power, het vermogen dat de renner levert, in Watt.
5. Afstand, de afgelegde afstand, in kilometers.
6. Cadans, de trapfrequentie, het aantal omwentelingen per minuut.
7. Hartslag, de hartslag van de renner, in slagen per minuut.
8. Intervallen, een kolom die nu nog niet gebruikt wordt, maar hier kan gemeten worden of de renner aan intervaltraining deed.
9. Hoogtemeters, de hoogte van de meting, in meters boven zeeniveau.

Iedere renner heeft op zijn fiets een kastje dat deze variabelen meet. Wanneer het kastje wordt aangezet begint deze met meten. De hartslag wordt gemeten met een hartslagmeter, die bevestigd is onder de borst van de wielrenner.

Rond dag 1138 is overgegaan op een ander meetsysteem. Als gevolg hiervan verloopt de meting van de tijd op een andere wijze. Voor de overgang werd de tijd gemeten in minuten, na de overgang in dagen. Na de overgang moet de eerste kolom dus vermenigvuldigd worden met  $24 \cdot 60$  om dezelfde tijdseenheid te krijgen als op de dagen voor de wissel in het meetsysteem. Tabel 2.1 toont een overzicht per renner van welke dag de eerste is binnen het nieuwe meetsysteem. De tijd tussen de verschillende metingen, de samplefrequentie, is in de meeste gevallen 1 seconde, ofwel 0,0167 minuten. Dit betekent dat er iedere seconde een meting wordt gedaan. Voor twee renners, renner 1 en renner 5 zijn er voor de eerste files afwijkende samplefrequenties, namelijk 0.5. In tabel 2.1 staan ook de nummers van de files waar de samplefrequentie voor het eerst 1 is.

	Renner 1	Renner 2	Renner 3	Renner 4	Renner 5
Eerste dag nieuw systeem	1138	1139	1143	1166	1138
Eerste dag sample rate = 1	777	-	-	-	375

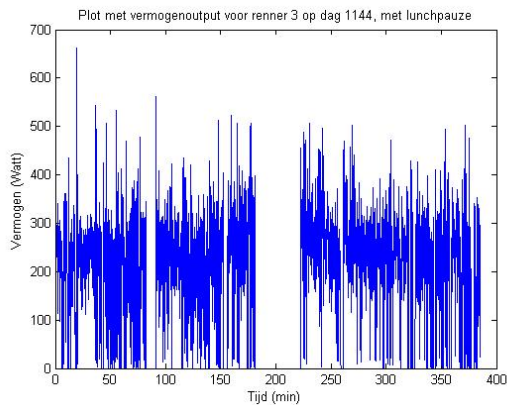
Tabel 2.1: Nummers van de datafiles waarvoor de verandering optreedt.

In het vervolg van dit verslag gaat er veel gewerkt worden met de vermogensoutput. Daarom zal hier meer over worden uiteenzet.

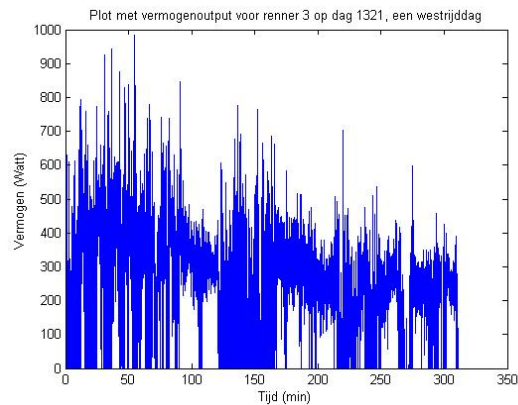
Het vermogen is een grootheid waarbij de eenheid Watt hoort, ook wel Joule per seconde. Het vermogen is dus een grootheid die aangeeft hoeveel energie er per seconde geleverd wordt. Iedere wielrenner heeft een vermogensmeter op de fiets waarmee dit vermogen bepaald wordt. Figuur 2.1 toont een plot van het vermogen voor een bepaalde dag. Omdat er heel veel datapunten zijn - 23024 punten voor deze plot - en er grote en sterke fluctuaties zijn in het vermogen, is het lastig om de grafiek af te lezen. Wel geeft de

grafiek een globaal beeld van hoe de renner heeft gefietst die dag.

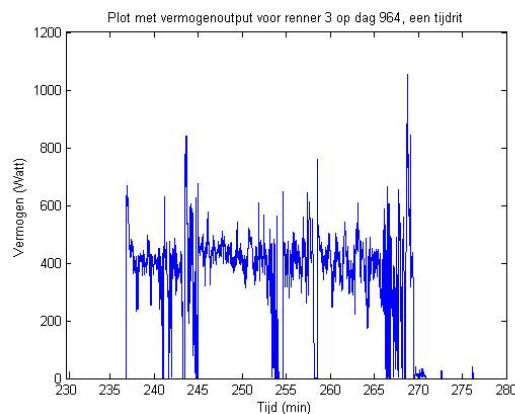
Iedere dataset bevat de data van één dag en dit kan dus meerdere trainingen bevatten. Het komt echter niet vaak voor dat er twee trainingen op één dag zijn. Wel houden de renners soms een koffie- of lunchpauze. Dit is zichtbaar in de data, het vermogen is dan voor langere tijd 0 Watt. Figuur 2.1 toont een plot van de tijd - vermogensoutput, waarin een lunchpauze zichtbaar is. Merk op dat er voor de lunchpauze ook een andere korte stop te zien is in de data.



Figuur 2.1: De vermogensoutput van een training.



Figuur 2.2: De vermogensoutput van een wedstrijd.



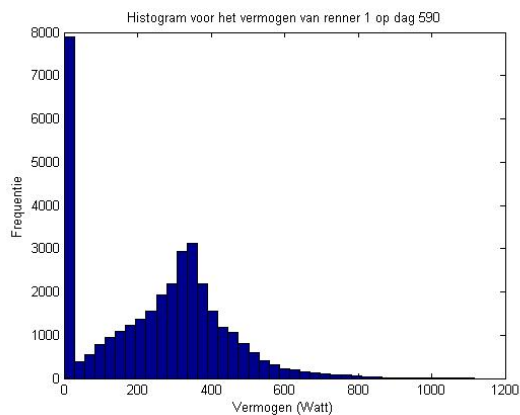
Figuur 2.3: De vermogensoutput van een tijdrit.

De plots van de vermogensoutput vertonen verschillende kenmerken. De plot in Figuur 2.1 toont een training, het gemiddelde vermogen ligt rond de 250 Watt. Wanneer er naar een wedstrijd gekeken wordt, zal het gemiddelde vermogen hoger liggen. Dit komt omdat de renners tijdens trainingen eigenlijk nooit helemaal voluit gaan, terwijl dat in wedstrijden wel het geval is, zoals te zien is in figuur 2.2. Ook individuele tijdritten zijn goed te herkennen binnen de data, hier geldt namelijk een erg constante vermogensoutput, zoals te zien is in figuur 2.3. Merk op dat de renner op de dag van de tijdrit al eerder die dag heeft gefietst, dit verklaart waarom de tijd niet bij 0 begint.

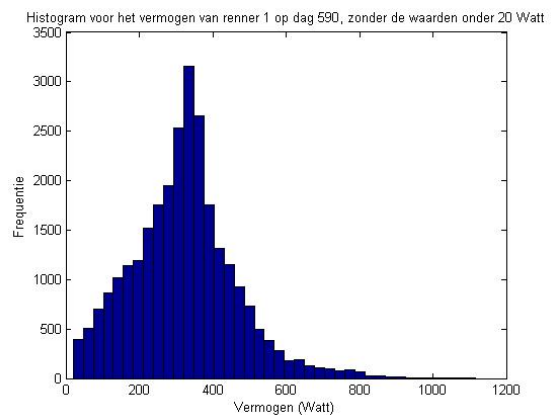
Een ander belangrijk detail om te noemen is het feit dat renners soms door de bergen rijden. Dan heeft het gewicht veel invloed op het vermogen. Een renner van 85 kilogram zal meer arbeid nodig hebben om dezelfde berg te beklimmen dan een renner van 65 kilogram. Iedere kilo moet namelijk tegen de zwaartekracht in megedragen worden de berg over. Voor bergetappes kan het goed zijn om niet naar het vermogen te kijken, maar naar het vermogen per kilogram. Op deze manier zijn renners te vergelijken. Op het vlakke heeft het gewicht geen invloed. Hier kan gewoon naar de vermogensdata gekeken worden.

Ook kan er op een andere manier naar de vermogensdata gekeken worden. Beschouw hierbij de vermogens-

output als een dataset  $y = (y_1, \dots, y_n)$ . Het feit dat de data tijdgeordend is, blijft hier buiten beschouwing.



Figuur 2.4: Histogram van de vermogensoutput.

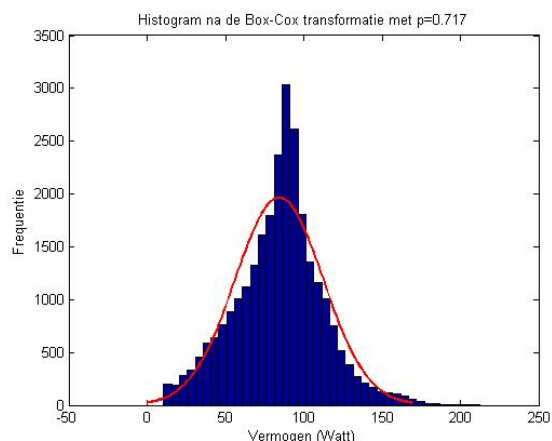


Figuur 2.5: Histogram van de vermogensoutput, zonder de waarden onder 20 Watt.

De histogram in figuur 2.4 laat zien dat de renner vaak een vermogen van 0 Watt levert. Dit is te verklaren door het feit dat renners niet altijd trappen. De renners houden bijvoorbeeld hun benen stil voor een bocht of tijdens het afdalen. Voor alle dagen en alle renners geldt dat de data een piek vertoont rond de 0 Watt. Figuur 2.5 toont de histogram van die waarden met een vermogen van meer dan 20 Watt. Deze histogram heeft nog slechts één piek. Dit verschijnsel geldt voor zowel wedstrijden als trainingen, maar het verschil is dat voor wedstrijden de locatie van de piek verder naar rechts ligt. Merk ook de asymmetrie op binnen de data. Dit verschijnsel vertoont zich voor meerdere dagen en meerdere renners. Door de data te transformeren kan deze asymmetrie omgezet worden in symmetrie. Een methode om data te transformeren is de Box-Cox transformatie

$$X \rightarrow X^{(p)} = \frac{X^p - 1}{p}. \quad (2.1)$$

Het transformeren van de data uit figuur 2.5 met  $p = 0.717$  geeft een enigszins symmetrisch beeld, zie figuur 2.6. Voor iedere dag en renner kan de data door middel van een Box-Cox transformatie getransformeerd worden tot een min of meer symmetrische dataset. De waarden van de verschillende  $p$ 's waarmee getransformeerd dient te worden, liggen wel veel uiteen. Deze waarden van  $p$  geven geen vermoeden voor een algemene  $p$  waarmee alle data getransformeerd kan worden.



Figuur 2.6: Histogram van de getransformeerde data.

Ook is het zo dat de data na de transformatie nog niet normaal verdeeld is, de rode lijn in figuur 2.6 geeft de vorm van de normale verdeling aan. De piek in de data is te groot om de data normaal verdeeld te laten zijn. De data lijkt uit gecombineerde verdelingen te komen, de normale verdeling en een verdeling die de piek beschrijft. Eventueel verder onderzoek zou in kunnen gaan op de verdeling van de vermogensoutput van wielrenners.



## 3 Achtergrond - Energiesystemen

Om een juiste basis te leggen en de wiskunde goed te kunnen toepassen, is het nodig om een kijkje te nemen in de fysiologie<sup>2</sup>.

Om te kunnen bewegen moeten de spieren samentrekken en daar is energie voor nodig. De benodigde energie komt vrij bij de splitsing van het stofje adenosine-trifosfaat (ATP) in adenosine-difosfaat (ADP) en fosfaat (P). De hoeveelheid ATP rond iedere spiervezel is echter zeer beperkt, iedere spier heeft maar voor een paar seconden energie op voorraad. Bij een inspanning zal het lichaam dus heel snel ATP aan moeten maken om een tekort te voorkomen. Het lichaam heeft drie verschillende energiesystemen die energiebronnen omzetten in ATP. Het soort energiebron dat daarvoor wordt gebruikt is afhankelijk van de tijdstuur en de intensiteit van de inspanning.

### 3.1 ATP + CrP systeem

In het lichaam is een kleine voorraad ATP en creatinefosfaat (CrP) aanwezig. Wanneer de kleine voorraad ATP op raakt, levert de splitsing van CrP in creatine (Cr) en P energie op om weer ATP te vormen uit de eerder gesplitste ADP en P. Het CrP in het lichaam kan slechts voor korte duur, zo'n 20 seconden, maar wel heel snel en efficiënt energie leveren. Vermoeidheid treedt dus snel op. Getrainde atleten hebben 3 tot 5 minuten rust nodig om de gehele ATP voorraad weer aan te vullen. Omdat dit proces geen zuurstof nodig heeft om ATP aan te maken, is dit een anaëroob proces, ofwel zuurstof onafhankelijk.

### 3.2 Anaëroob lactisch systeem

Voor een langere inspanning schakelt het lichaam over op een anaëroob systeem met voldoende energievoorraad voor 10 seconden tot 2 minuten. Omdat dit systeem, ook wel glycolyse genoemd, gebaseerd is op het omzetten van energie uit glucose (dat is aangemaakt uit gegeten koolhydraten) in ATP, reageert dit systeem minder snel dan het ATP + CrP systeem. Gedurende deze glycolyse worden koolhydraten in de vorm van bloedglucose (suiker) of spierglucose (de opgeslagen vorm van glucose) afgebroken door een serie van chemische reacties. Voor ieder molecuul glucose dat door dit proces afgebroken wordt, ontstaan er twee moleculen ATP. Er wordt niet heel veel energie tegelijk geproduceerd, maar daarentegen kun je de energie wel redelijk snel verkrijgen.

Een bijproduct van deze reactie is lactaat, wat via de lever weer omgezet wordt in glucose. Wanneer er echter meer lactaat wordt aangemaakt dan er wordt omgezet in de lever, ontstaat er een overschot aan waterstofionen in het bloed. Dit veroorzaakt een hoog zuurgehalte. Hierdoor kunnen de spieren zich niet meer samentrekken. Dit is het welbekende verschijnsel van verzuring. Deze ophoping van lactaat is ook de reden waarom er van dit energiesysteem niet oneindig lang gebruik gemaakt kan worden. Het punt waarop de verzuring net niet optreedt, dus het punt waarop er evenveel lactaat wordt afgebroken als aangemaakt, heet de lactaatdrempel.

### 3.3 Aëroob oxidatief systeem

Wanneer er een nog langere inspanning wordt geleverd, is het derde energiesysteem nodig. Dit is het aërobe systeem, het belangrijkste energiesysteem van ons lichaam. Dit systeem is afhankelijk van zuurstof en kan bij een matige intensiteit energie leveren. Dit systeem kan veel ATP leveren, maar het is wel de langzaamste methode van de drie. Het aërobe systeem gebruikt koolhydraten en vet als brandstoffen om ATP aan te maken. Als het de koolhydraten gebruikt, worden de glucose en glycogeen (de in het lichaam opgeslagen vorm van koolhydraten) omgezet door middel van glycolyse, wat we al tegen kwamen bij het anaërobe lactische systeem, tot pyrodruivenzuur. Dit pyrodruivenzuur wordt met voldoende zuurstof via een ingewikkeld proces omgezet tot ATP en water. Bij dit proces worden er voor ieder molecuul glucose 36 moleculen ATP geproduceerd.

Vet, wat opgeslagen is als triglyceride in het weefsel onder de huid en in de spieren, is de andere grote

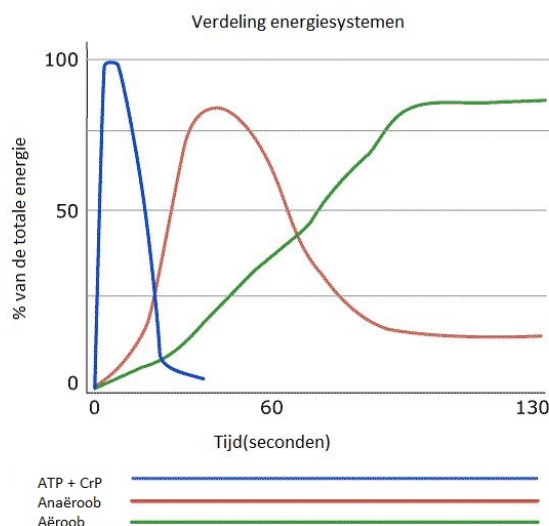
---

<sup>2</sup>Deze fysiologische informatie is ontleend aan Wilmore en Costill [28].

energiebron en heeft de grootste voorraad in het lichaam. Dit vet wordt eerst omgezet in drie vette zuren en glycerol, welke daarna allen op eenzelfde wijze als de glucose en glycereen omgezet worden tot ATP. Deze vette zuren kunnen veel meer ATP moleculen leveren dan de glucose of glycogeen. In totaal levert het omzetten van 1 molecuul vette zuren 129 moleculen ATP op.

Systeem	Snelheid ATP productie	Capaciteit ATP productie	Gebruikte brandstof
ATP + CrP	erg hoog	erg laag	opgeslagen ATP creatinefosfaat
anaëroob lac-tisch	hoog	laag	bloedglucose spierglucogeen
aëroob	laag	erg hoog	bloedglucose spierglucogeen vet

Tabel 3.1: Overzicht van de verschillende energiesystemen.



Tabel 3.1 geeft een samenvattende weergave van de verschillende systemen. Figuur 3.1 geeft de samenhang en verdeling van de systemen weer. Het zal ook zo zijn dat een wielrenner na vijf uur koersen nog wel in staat is om een energiepiek te leveren, denk aan een eindsprint. Op zo'n moment leveren alle drie de energiesystemen maximale energie om de renner zo hard mogelijk te laten gaan.

Figuur 3.1: De verdeling van de verschillende energiesystemen over de tijd<sup>3</sup>.

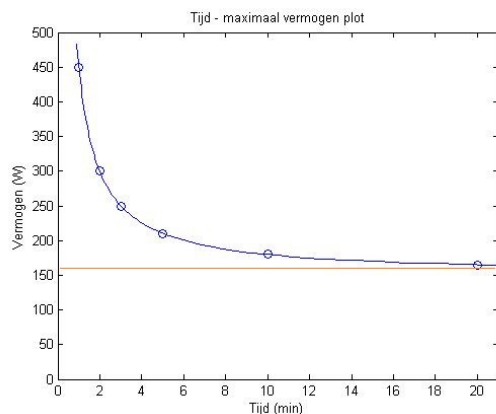
<sup>3</sup>Bewerking van <http://crossfitsouthie.com/three-metabolic-pathways/>

## 4 Het bepalen van een maat van prestatie

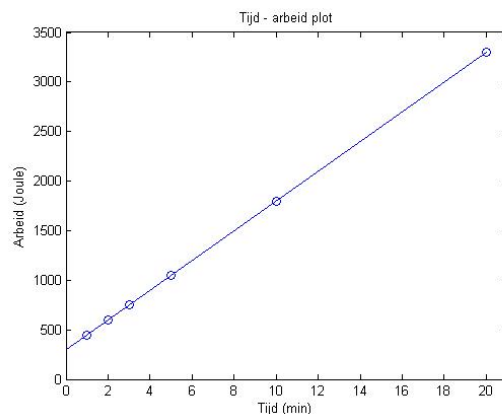
Nu bekend is hoe de energiesystemen in het lichaam werken, kan hiermee een maat van prestatie bepaald worden.

### 4.1 Critical Power

Op het moment dat er een erg grote inspanning geleverd moet worden, kan dit veel minder lang volgehouden worden dan het leveren van een veel lagere inspanning. Monod en Scherrer [19] waren de eersten die een dergelijke relatie tussen vermogensopbrengst en tijd tot uitputting van bepaalde spiergroepen gedurende 3, 4 of 5 repeterende sessies op de kaart zetten. Zij noteerden een hyperbolische relatie tussen de vermogensopbrengst en de tijd tot uitputting. Figuur 4.1 toont een dergelijke relatie.



Figuur 4.1: Het door Monod en Scherrer in kaart gebrachte verband.



Figuur 4.2: De getransformeerde relatie.

De hyperbolische relatie kan getransformeerd worden tot de lineaire relatie tussen de totaal verrichte arbeid (tijd · vermogen) in Joule en de tijd tot uitputting, zie figuur 4.2. De helling van deze lijn wordt gedefinieerd als het kritieke vermogen - vanaf nu de critical power ( $CP$ ) genoemd. De eenheid die hoort bij de critical power is Watt.

Deze critical power kan ook benoemd worden in figuur 4.1. Dit is namelijk de horizontale asymptoot. Fysisch is deze waarde te interpreteren als het maximale vermogen dat heel lang volgehouden kan worden, zonder moe te worden. Er vindt op dit vermogen dus geen verzuring plaats en de voorraad voor ATP of CrP raakt niet op. Monod en Scherrer [19] suggereerden dan ook dat er geen vermoeidheid op kan treden, zolang het opgelegde vermogen lager of gelijk aan de critical power is. Een vermogen van bijvoorbeeld 100 Watt zou volgens figuur 4.1 oneindig lang volgehouden moeten kunnen worden.

Moritani en zijn collega's [20] breidden het concept van de critical power uit en ontdekten dat dit ook geldt voor inspanningen die met het hele lichaam gedaan worden, denk aan voetbal, schaatsen of wielrennen. Wanneer in het vervolg gesproken wordt over het vermogen dat de wielrenner levert, gaat dit over het vermogen dat gebruikt wordt voor het fietsen. De wielrenner verbruikt ook energie om interne processen gaande te houden, dezelfde energie die verbruikt wordt als er stil op een stoel gezeten zou worden. Deze intern verbruikte energie wordt niet bekeken.

Moritani en co laten zien dat de critical power hoog gecorreleerd is met de anaërobe drempel, de eerdergenoemde lactaatrempel. Op het moment dat de energie voor dit vermogen geleverd wordt door het ATP + CrP systeem, of het anaërobe lactische systeem, zou de energievoorraad op een bepaald moment uitgeput zijn. Dit kan niet, de waarde van de critical power kan immers heel lang volgehouden worden. Het aërobe systeem moet hier dus wel aan te pas komen. Op het moment dat de lactaatrempel is bereikt en het lichaam voor de energietoevoer overgaat op de aërobe energie, wordt er gefietst met een vermogen van dat rond de critical power. Een derde bewijs vonden zij voor het feit dat de y-intercept voor de arbeid-tijd relatie, van anaërobe natuur is. Deze wordt niet beïnvloed door de afwezigheid van zuurstof. Zij noemden

deze intercept de anaërobe arbeidscapaciteit ( $AWC$  of  $W'$ ). Deze  $W'$  heeft als eenheid Joule en is gelijk aan de totale hoeveelheid energie die vrij kan komen vanuit het ATP + CrP systeem en het anaëroob lactisch systeem.

Met de kennis van  $CP$  en  $W'$  kunnen de vergelijkingen opgesteld worden bij figuur 4.1 en 4.2, respectievelijk

$$P = CP + \frac{W'}{t} \quad (4.1)$$

welke ook wel het nonlineaire 2-parameter model genoemd wordt, en

$$TW = W' + CP \cdot t \quad (4.2)$$

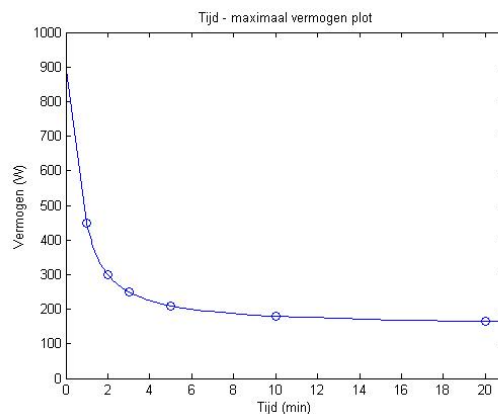
ook wel het lineaire TW model. In deze modellen is  $P$  het vermogen in Watt,  $t$  de tijd in seconden en  $TW$  de totale arbeid in Joule.

Eén van de kritische noten die gesteld werd bij het model uit vergelijking (4.1) is dat er volgens dit model op een heel klein tijdsinterval een heel groot vermogen geleverd kan worden. Fysisch is dit onmogelijk. Voor een wielrenner op topniveau zal gelden dat het maximaal mogelijke vermogen dat deze voor 1 seconde kan leveren onder de 2500 Watt ligt.

Deze observatie bracht Morton [21] tot een uitbreiding van het model, dat het nonlineaire 3-parameter model heet. Door figuur 4.1  $\tau$  naar links te verschuiven, ontstaat de grafiek bij de formule

$$P = CP + \frac{W'}{t + \tau} \quad (4.3)$$

Dit geeft een limiet op het vermogen, namelijk  $P_{max}$ . Figuur 4.3 toont een weergave hiervan.



Figuur 4.3: De grafiek bij de vergelijking  $P = CP + \frac{W'}{t + \tau}$ .

De constructie van deze modellen brengt een aantal aannames met zich mee [8].

#### Aannames omtrent de critical power

1. Slechts twee componenten dragen bij aan het energietoevoersysteem voor menselijke inspanning, deze componenten noemen we aëroob en anaëroob.
2. De aërobe toevoer is ongelimiteerd in tijdsduur, maar gelimiteerd in grootte. Deze limiterende factor heet de critical power.
3. De anaërobe toevoer is ongelimiteerd in grootte, maar de capaciteit is gelimiteerd door de parameter  $W'$ .
4. Uitputting, en dus het beëindigen van inspanning, treedt op wanneer de  $W'$  uitgeput is.
5.  $CP$  en  $W'$  zijn constanten, onafhankelijk van  $P$  en/of  $t$ , maar verschillen wel per persoon.

Onder deze aannames zijn in de afgelopen 30 jaar papers geschreven over het vinden van goede fits voor de data van de vergelijkingen (4.1), (4.2) en (4.3). Binnen zekere grenzen lukte dit altijd wel [22].

Met de van Team Giant-Alpecin ontvangen data, die onder andere het vermogen en de tijd bevat, kunnen de critical power  $CP$  en de anaërobisch arbeidscapaciteit  $W'$  berekend worden. Hiervoor zijn eerst de punten nodig die aangeven hoe lang een renner een bepaald vermogen vol kan houden, of het maximale vermogen dat een renner kan leveren gedurende een bepaalde tijd. Helaas doen de renners geen tests om te kijken hoe lang ze een bepaald vermogen vol kunnen houden. Wel is de tijd-vermogens data beschikbaar. Het maximale vermogen dat een renner in een bepaalde tijd kan leveren, kan berekend worden door te kijken naar de beste 1 seconde uit alle data, de beste 1 minuut, de beste 20 minuten en met beste wordt dan bedoeld: het hoogste gemiddelde vermogen.

Dit gemiddelde vermogen is analoog aan het voortschrijdend gemiddelde [25]. Het voortschrijdend gemiddelde is het gemiddelde van een vast aantal opeenvolgende elementen in een tijdreeks. Om het voortschrijdend gemiddelde over een periode  $N$  te vinden voor een tijdreeks  $(y_n)$  bekijk je het gemiddelde van de  $N - 1$  elementen voor  $y_n$  en  $y_n$  zelf. Dit wordt genoteerd door

$$VG_n = \frac{1}{N} \sum_{k=0}^{N-1} y_{n-k} \quad (4.4)$$

#### Aanname omtrent het voortschrijdend gemiddelde

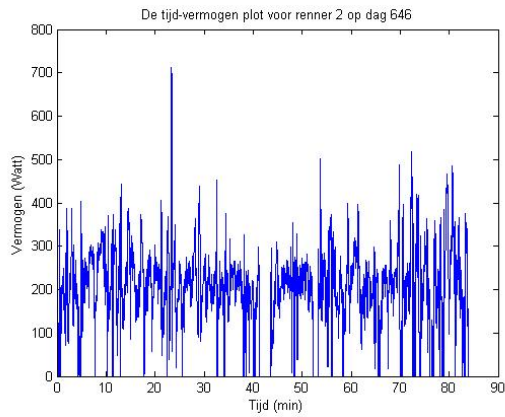
Het gemiddelde vermogen over  $N$  seconden is analoog aan het voortschrijdende gemiddelde

$$VG_n = \frac{1}{N} \sum_{k=0}^{N-1} y_{n-k}$$

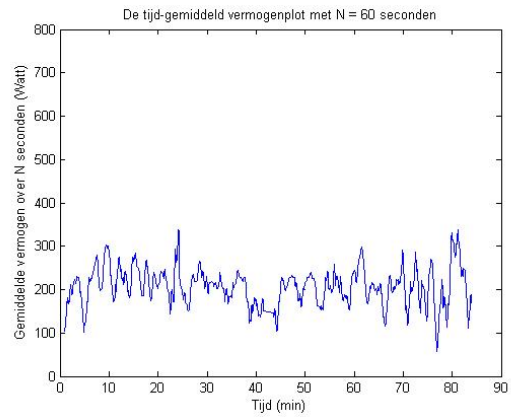
Op deze manier kan het voortschrijdend gemiddelde berekend worden over een bepaalde periode  $N$ . Figuur 4.4 toont afbeeldingen van een dag met verschillende afbeeldingen voor verschillende perioden waarover het voortschrijdende gemiddelde is berekend, namelijk  $N = 1$  seconde,  $N = 60$ ,  $N = 300$  en  $N = 1200$ .

Merk op dat de grafiek steeds verder afvlakt, naarmate  $N$  groter wordt. Dit heeft te maken met het feit dat het gemiddelde bij een grotere  $N$  over steeds meer waarden bepaald wordt, dus een extreem hoge of lage waarde heeft dan veel minder invloed.

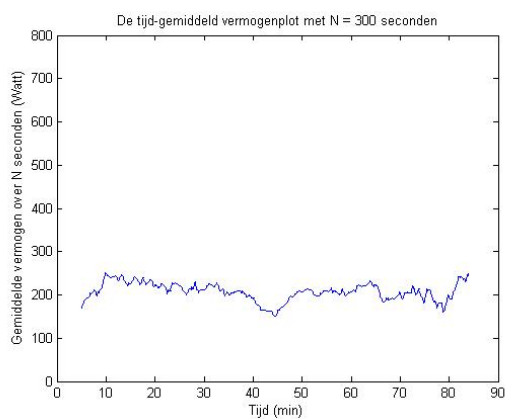
De waarden van het maximale vermogen die bepaald worden, zijn de waarden voor  $N = 1, 5, 10, 30, 60, 120, 300, 600, 900, 1200, 1500, 1800, 2400, 3000$  en  $3600$ , dus de verschillende tijden tussen 1 seconde en 1 uur. Gewenst is om het hoogste gemiddelde vermogen te vinden, dus wordt het maximum van het voorschrijdend gemiddelde over iedere dag bepaald, en vervolgens over alle dagen ook weer het maximum. Dit geeft de beste 1 seconde, beste 5 seconden, beste 10 seconden enzovoort die de renner ooit heeft gefietst. Tabel 4.1 toont deze maxima voor alle renners en figuur 4.5 toont een plot van de waarden voor renner 2. Nu deze waarden bekend zijn, kunnen met behulp van lineaire dan wel niet-lineaire regressie de waarden voor  $CP$  en  $W'$  bepaald worden volgens de verschillende modellen (4.1), (4.2) en (4.3). Deze  $CP$  en  $W'$  zijn bepaald over alle data, dit is dus geen waarde per jaar of per dag, maar één waarde voor één renner, berekend met alle beschikbare data.



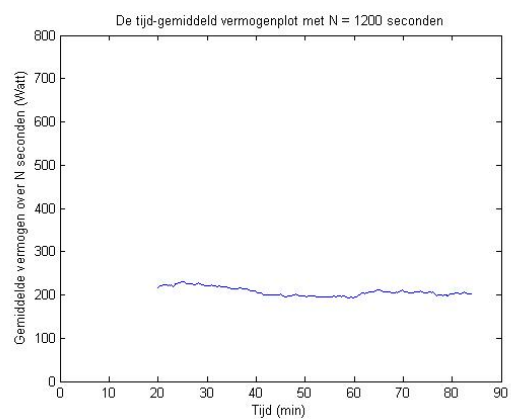
(a)  $N = 1$



(b)  $N = 60$



(c)  $N = 300$

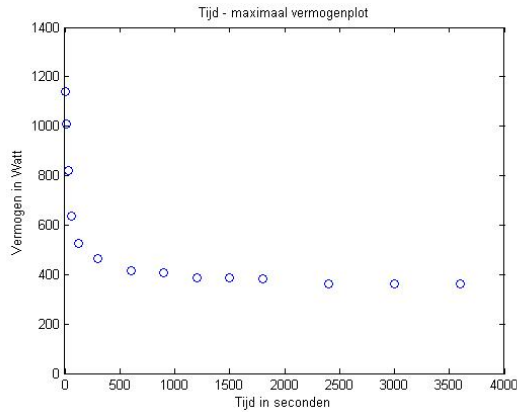


(d)  $N = 1200$

Figuur 4.4: De tijd-vermogen plots voor 1 renner op een bepaalde dag, met verschillende perioden  $N$  waarover het voorschrijdend gemiddelde is berekend.

Tijd (seconden)	Maximale vermogen (Watt)				
	Renner 1	Renner 2	Renner 3	Renner 4	Renner 5
1	2351	1982	2000	1218	1801
5	1422	1141	1363	998	1661
10	1076	1010	1170	875	1443
30	856	820	892	655	1042
60	685	638	671	590	721
120	570	525	590	493	579
300	486	467	472	455	499
600	458	414	436	392	447
900	448	405	435	374	414
1200	430	389	433	361	413
1500	425	386	401	348	389
1800	425	382	399	333	383
2400	395	362	377	331	364
3000	392	361	373	305	365
3600	386	360	373	2925	352

Tabel 4.1: De waarden die de basis vormen voor het bepalen van de critical power.



Figuur 4.5: De tijd - maximaal vermogen plot voor renner 2.

Voordat de regressie toepast kan worden, is het nodig nog eens goed naar de verschillende bepaalde tijdstippen te kijken. De maxima die bepaald zijn door het voortschrijdend gemiddelde, moeten ook daadwerkelijk maxima zijn. Hiertoe moeten de wielrenners dus 20, 30 of zelfs 60 minuten voluit gegaan zijn. In praktijk gebeurt dit niet, noch tijdens wedstrijden noch tijdens trainingen. Wel is het zo dat de renners jaarlijks minimaal éénmaal een 20 minuten maximum test doen. Dit houdt in dat zij 20 minuten lang zo hard mogelijk fietsen en dus het maximale vermogen leveren. Dit is dus een zeer goede indicatie voor het 20 minuten maximale vermogen. Voor de tijden langer dan 20 minuten zijn er geen goede benaderingen. Daarom worden die waarden niet meegenomen in de lineaire regressie. Ook de waarden die de sprint betreffen, 1, 5, 10 en 30 seconden, worden niet meegenomen in de regressie. Dit vanwege het feit dat er voor dergelijke sprints geen gebruik wordt gemaakt van het aëroob oxidatief systeem en dat deze waarde de regressie ernstig zullen beïnvloeden.

De regressie wordt uitgevoerd met Matlab, met de functie `lsqcurvefit()`. Het model achter deze regressie is de volgende

$$y = f(X, \beta) + \epsilon \quad (4.5)$$

met  $y$  een  $n$  bij 1 vector die de observaties van de responsvariabele bevat,  $f$  een functie van  $X$  en  $\beta$  die voor iedere  $X$  met  $\beta$  een voorspelling doet voor de waarde van de bijbehorende  $y$ ,  $X$  is de  $n$  bij 1 vector van verklarende variabelen,  $\beta$  is de vector die de te schatten parameters bevat en  $\epsilon$  is de  $n$  bij 1 vector die de onafhankelijke, identiek verdeelde random fouten bevat. Merk op dat er voor nonlineaire regressie, integenstelling tot lineaire regressie, geen aannames worden gedaan omtrent de verdelingen van de  $X$  en  $\epsilon$  [17].

Voor het nonlineaire 2-parameter model (4.1) gold  $P = CP + \frac{W'}{t}$ . Hier wordt de non-lineaire regressie toegepast om de waarden van  $W'$  en  $CP$  te vinden door de  $CP$  en  $W'$  te fitten aan

$$P_i = CP_i + \frac{W'}{t_i} + \epsilon_i. \quad (4.6)$$

Tabel 4.2 geeft deze waarden weer. Tabel 4.3 geeft de waarde voor de gefitte  $CP$  en  $W'$  van

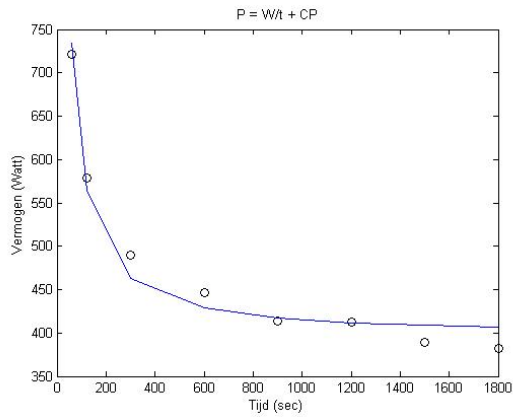
$$TW_i = W' + CP \cdot t_i + \epsilon_i \quad (4.7)$$

weer en tabel 4.4 geeft de waarden weer van het laatste model

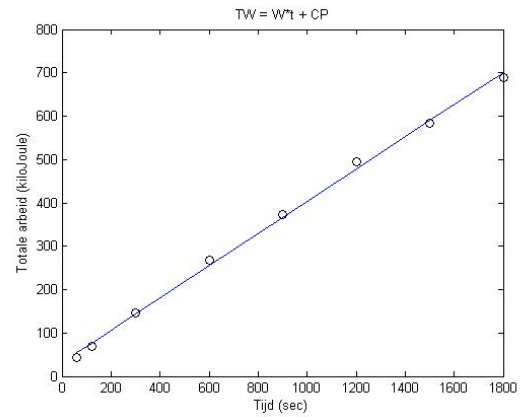
$$P_i = CP + \frac{W'}{t_i + \tau} + \epsilon_i. \quad (4.8)$$

Figuur 4.6 geeft een grafische weergave van de gefitte formules voor renner 5.

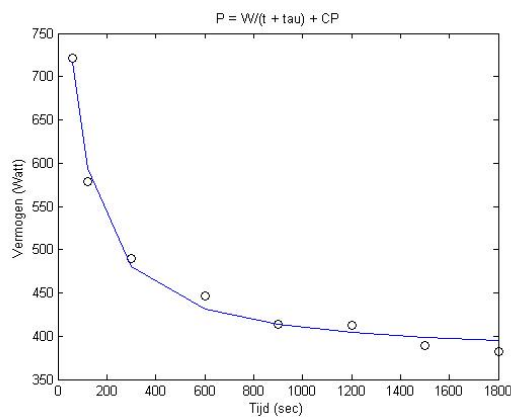
Met een drietal verschillende modellen is nu voor iedere renner een drietal verschillende waarden voor de  $CP$  en  $W'$  gevonden. Bergstrom et al. [4] beschrijven het feit dat er uit de verschillende modellen verschillende



(a)  $CP = 395, W' = 20337$



(b)  $CP = 371, W' = 32589$



(c)  $CP = 375, W' = 36380$

Figuur 4.6: De plots met de gefitte waarden voor  $CP$  en  $W'$  voor renner 5

	$CP$ (Watt)	95% betrouwbaarheidsinterval	$W'$ (Joule)	95% betrouwbaarheidsinterval	$r^2$
renner 1	424	[413;435]	16100	[14501;17700]	0.99020
renner 2	386	[369;402[	15683	[13301;18064]	0.97741
renner 3	409	[385;434]	16929	[13340;20520]	0.95688
renner 4	342	[322;362]	15386	[12473;18299]	0.96532
renner 5	395	[373;417]	20337	[17088;23586]	0.97506

Tabel 4.2: De bepaalde waarden voor  $CP$  en  $W'$  onder het nonlineaire 2 model.

	$CP$ (Watt)	95% betrouwbaarheidsinterval	$W'$ (Joule)	95% betrouwbaarheidsinterval	$r^2$
renner 1	414	[406;421]	21221	[13569;28873]	0.99967
renner 2	372	[364;379]	22365	[14516;30221]	0.99956
renner 3	391	[372;411]	26186	[6446.9;45924]	0.99752
renner 4	319	[301;337]	27485	[9354.7;64324]	0.99686
renner 5	371	[354;389]	32589	[15046;50132]	0.99782

Tabel 4.3: De bepaalde waarden voor  $CP$  en  $W'$  onder het lineaire TW model.



	$CP$ (Watt)	95% betrouw- baarheidsinter- val	$W'$ (Joule)	95% betrouw- baarheidsinter- val	$\tau$ (sec)	95% betrouw- baarheidsinter- val	$r^2$
renner 1	415	[405;424]	29211	[23328;35094]	26.3	[5.02;47.6]	0.99741
renner 2	371	[354;389]	27314	[14715;39915]	43.8	[-2.44;90.2]	0.99185
renner 3	387	[361;413]	35410	[14486;56335]	63.4	[-6.53;133.51]	0.98742
renner 4	327	[299;354]	28018	[7958.0;48079]	48.5	[-2.65;123.60]	0.98085
renner 5	375	[350;400]	36280	[18209;54352]	46.3	[-4.88;97.56]	0.99058

Tabel 4.4: De bepaalde waarden voor  $CP$ ,  $W'$  en  $\tau$  onder het nonlineaire 3 model.

waarden komen en vergeleken de modellen onderling met elkaar. Bevindingen uit dit onderzoek zijn onder andere dat het lineaire P model de waarde voor de  $CP$  overschat en het nonlineaire 2 model de waarde voor  $W'$  onderschat. De verschillende modellen vergelijkend, komen zij tot de conclusie dat het nonlineaire 3 model de nauwkeurigste schattingen geeft voor de  $CP$  en  $W'$ . Deze waarden zijn ook fysisch het aannemelijkst. Merk wel op dat de 95% betrouwbaarheidsintervallen voor de schattingen binnen het nonlineaire 3 model, zeker voor  $W'$  relatief groot zijn. Dit is wel iets wat genoemd dient te worden.

Literatuur kent meer modellen omtrent het bepalen van de  $CP$  en  $W'$ . Een aantal hiervan, namelijk het lineaire P-model [4], het EXP-model [4] en Wilkies model [26], zijn bepaald binnen dit onderzoek, maar deze gaven geen betere schattingen. Over andere modellen, zoals het extended model [5] is zeer weinig literatuur beschikbaar en om deze reden zijn deze modellen niet meegenomen in dit verslag. Voor een eventueel vervolg zouden hier mogelijkheden liggen.

## 4.2 $W'$ balans

Wanneer er een inspanning wordt geleverd met een vermogen boven de critical power, zal de energievoorraad van ATP en CrP uitputten en zal het zuurgehalte in de spieren toenemen. Dit kan weergegeven worden door een batterij. De volledige hoeveelheid energie die deze batterij kan bevatten, komt overeen met de  $W'$  die Moritani [20] meette. Als er inspanning verricht wordt met een vermogen boven de critical power, loopt de batterij leeg, wordt er inspanning verricht met een vermogen beneden de critical power, dan laadt deze weer op.

Skiba en zijn collega's [24] onderzochten het verloop van  $W'$  gedurende inspanning en wilden in hun onderzoek een dynamisch model weergeven dat het verbruik en het opladen van de batterij bijhoudt gedurende periodieke inspanning in het wielrennen. Met periodieke inspanning wordt het fietsen op intervallen gedaan, een bepaalde periode op een laag vermogen gevolgd door een periode op een hoog vermogen, en dit herhalend. Deze perioden hoeven niet allemaal gelijk te zijn. Denk bijvoorbeeld aan een bergetappe: wanneer er bergop gefietst wordt, kost dit veel inspanning. De afdaling kost veel minder inspanning. Ook op het vlakke doet een dergelijk patroon zich voort. Soms wordt er aangezet, soms wordt er een bepaalde tijd een lager tempo gefietst. In hun onderzoek vonden Skiba en collega's dat de energievoorraad in het lichaam exponentieel oplaadt op het moment dat er inspanning wordt geleverd beneden de critical power. Het verloop van deze energievoorraad noemen we  $W'$ balans, afgekort  $W'_{bal}$ . Er geldt dat  $W'_{bal} = W'$  op het moment dat er nog geen inspanning heeft plaatsgevonden, de batterij is dan nog helemaal vol. Het model voor het verloop van  $W'_{bal}$  is gebaseerd op de volgende aannames [24]:

### Aannames omtrent $W'_{bal}$

1. Het verbruik van  $W'$  begint op het moment dat de renner een vermogen levert boven de  $CP$ .
2. Het herstel van  $W'$  begint op het moment dat de renner een vermogen levert beneden de  $CP$ .
3. Het herstel van  $W'$  gaat op een exponentiële wijze.

en luidt als volgt

### Vergelijking $W'_{bal}$

$$W'_{bal}(t) = W' - \int_0^t W'_{exp}(u) \cdot e^{-\frac{(t-u)}{\tau_{W'}}} du \quad (4.9)$$

Hier is  $W'_{bal}(t)$  de balans van het overgebleven  $W'$  op een gegeven tijdstip  $t$ ,  $W'$  is gelijk aan de bekende  $W'$  van een wielrenner,  $W'_{exp}(u)$  is gelijk aan de verbruikte hoeveelheid energie op het tijdstip  $u$  en  $(t-u)$  is de tijd in seconden tussen die delen van de inspanning die resulteren in uitputting van  $W'$ , dus de momenten waar  $W'$  gereconstueerd wordt. De  $\tau_{W'}$  is de tijdsconstante die aangeeft hoe snel  $W'$  is gereconstrueerd. We kunnen zeggen dat de  $W'_{bal}$  op het tijdstip  $t$ , gelijk is aan de bekende batterij  $W'$  min de totale som aan joules van de batterij die verbruikt zijn tot tijdstip  $t$  gedurende de inspanning plus die joules die weer zijn toegevoegd aan de batterij door het herstel. Op de verdere precieze constructie en inhoud van deze vergelijking wordt in dit verslag niet ingegaan. Literatuur heeft de correctheid van deze vergelijking aangetoond [25, 26, 16], ook in de toepassing van het wielrennen met data van trainingen en wedstrijden [25]. Skiba bepaalde ook de  $\tau_{W'}$ , te weten

$$\tau_{W'} = 546e^{(-0.01D_{CP})} + 316 \quad (4.10)$$

Hierin is  $D_{CP}$  gelijk aan het verschil tussen de critical power en het gemiddelde herstelvermogen. Dit gemiddelde herstelvermogen is gelijk aan het gemiddelde van alle vermogens beneden de CP [25]. De  $\tau_{W'}$  is dus rennerafhankelijk. Sommige renners zullen sneller herstellen dan anderen.

Met een discretisatie<sup>4</sup> kan de integraal

$$I_t = \int_0^t W'_{exp}(u) \cdot e^{-\frac{(t-u)}{\tau_{W'}}} du \quad (4.11)$$

omgeschreven worden. De meetwaarden zijn namelijk discreet, dus het bepalen van de integraal is niet mogelijk. Omdat de totale tijdsduur  $t$  niet voor iedere file gelijk is aan het aantal datasamples  $N$ , wordt de samplerate  $T_s$  geïntroduceerd, met  $t = T_s \cdot N$ . Omdat  $\tau_{W'} \gg T_s$ , mag de integraal beschouwd worden als een Riemanssom. Sommeer van  $n = 1$  tot  $N$  en kies  $N$  zodanig dat dit gelijk is aan het aantal datasamples voor de desbetreffende dag. De  $\frac{(t-u)}{\tau_{W'}}$  in de exponent wordt dus vervangen door  $\frac{(N-n) \cdot T_s}{\tau_{W'}}$  en dit geeft

$$\begin{aligned} I_N &= \sum_{n=1}^N W'_{exp}(n) \cdot e^{-\frac{(N-n) \cdot T_s}{\tau_{W'}}} \\ &= e^{-\frac{N \cdot T_s}{\tau_{W'}}} \cdot \sum_{n=1}^N W'_{exp}(n) \cdot e^{\frac{n \cdot T_s}{\tau_{W'}}}, \end{aligned} \quad (4.12)$$

wat ontleed kan worden in

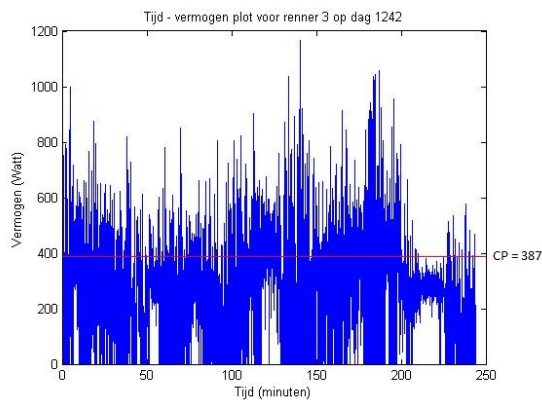
$$S_N = S_{N-1} + W'_{exp}(N) \cdot e^{\frac{N \cdot T_s}{\tau_{W'}}} \quad (4.13)$$

met

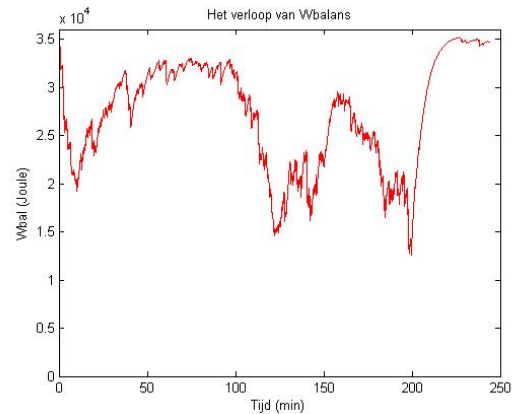
$$I_N = e^{-\frac{N \cdot T_s}{\tau_{W'}}} \cdot S_N \text{ en } S_0 = 0. \quad (4.14)$$

Figuur 4.7 toont een voorbeeld van de toepassing van dit model. Er is duidelijk te zien dat de batterij leegloopt om het moment dat de wielrenner een vermogen levert boven de CP. Op de stukjes waar de renner een lager vermogen trapt, laadt de batterij weer op.

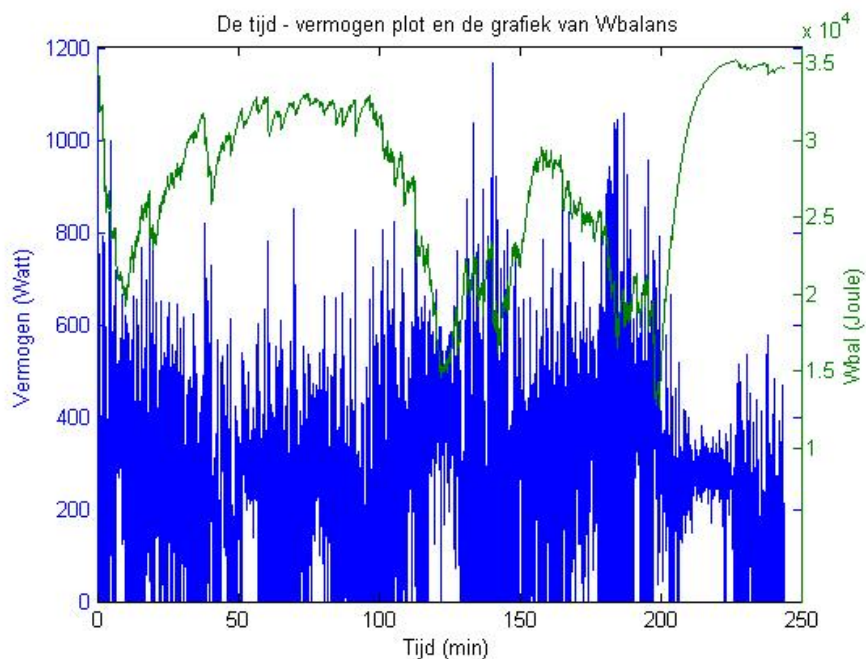
Dit model kan onder andere gebruikt worden om te zien hoe verschillende renners omgaan met hun energie gedurende een wedstrijd. Figuur 4.8 toont een plot van de  $W'_{bal}$  van vijf renners die dezelfde wedstrijd hebben gereden. In deze plot is te zien dat renner 1 rond de tachtigste minuut een enorme inspanning heeft geleverd in vergelijking met de andere vier renners. Het zou zo kunnen zijn dat deze renner toen kopwerk heeft gedaan of is gedemarreerd. Ook is te zien dat renner 1 en 2 aan het einde van de wedstrijd nog veel energie hebben geleverd. De andere renners hebben toen niet meer het uiterste uit de kan gehaald. Dit kan



(a) Het gefietste vermogen



(b) Het verloop van  $W'$ bal



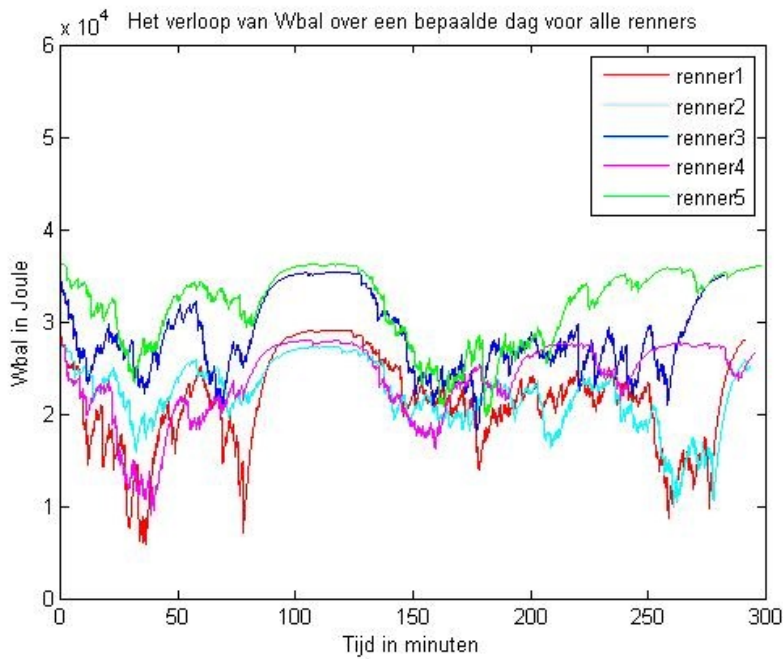
(c) Beide grafieken over elkaar heen

Figuur 4.7: De visuele weergave van de  $W'$ balans voor renner 3 op dag 1242.

natuurlijk vele factoren hebben. Een dergelijke plot kan naast het logboek van de renners gelegd worden om er vergelijkingen uit te halen.

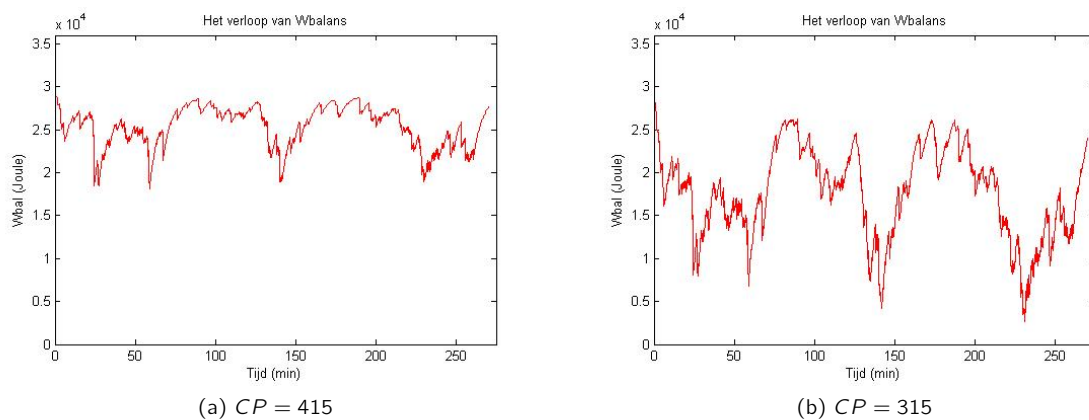
Nu het model omtrent  $W'_{\text{bal}}$  uiteengezet is, kan een maat van prestatie bepaald worden. Herinner het feit dat de waarde van  $W'_{\text{bal}}$  bepaald wordt aan de hand van de critical power. We kunnen het model terugrekenen om die waarde te vinden, met behulp van de bekende  $W'$  en de tijd-vermogen data voor een bepaalde dag. Als wielrenner is het qua energieverdeling het ideaalst als de batterij aan het einde van de rit leeg is, dan geldt  $W'_{\text{bal}} = 0$ . Denk aan sprinters in een eindsprint, die gebruiken al hun overige energie om de sprint te winnen, zij zullen een lege batterij hebben aan het einde van de rit. Dit geldt echter voor de meeste renners niet. Denk aan de renner die de sprint aantrekt, dit is een renner die heel erg hard op de kop van de groep gaat rijden, met de sprinter achter zich. Deze 'sprintaantrekker' geeft al zijn energie om de sprinter goed te lanceren in de sprint. Deze zal op dat moment het laagste niveau van  $W'_{\text{bal}}$  halen en de sprinter het af laten maken. Hetzelfde geldt voor de renner die een klimmer afzet onder aan een berg. Figuur 4.7

<sup>4</sup>Het idee voor deze discretisatie is genoteerd door Dave Waterworth [27].



Figuur 4.8: Het verloop van  $W'_{\text{bal}}$  voor vijf renners in dezelfde wedstrijd op dag 1240.

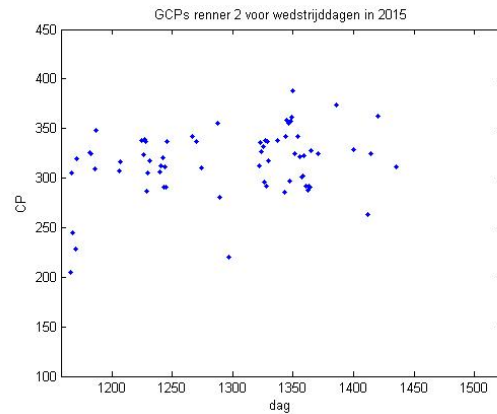
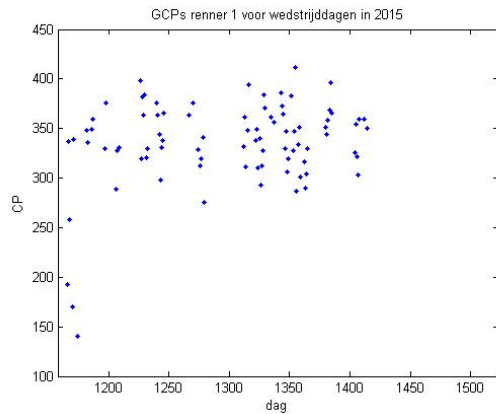
laat een dergelijke situatie zien. 50 minuten voor het einde van de wedstrijd heeft deze renner zijn grootste inspanning geleverd en fietst de wedstrijd uit op een lager vermogen. Het is dus niet zo dat alle renners hun minimale waarde voor  $W'_{\text{bal}}$  halen aan het einde van de rit. Wel mag aangenomen worden dat ze elke dag een minimum behalen. Theoretisch zouden renners op zo'n laag vermogen kunnen fietsen dat ze alle energie verkrijgen via het aëroë systeem, maar er wordt aangenomen dat dit in de gemeten data niet voorkomt. Met het model uit vergelijking (4.9) kan terugerekend worden voor welke  $CP$  geldt dat  $W'_{\text{bal}}$  precies één keer op die dag gelijk aan 0 wordt. De waarde die dit oplevert, wordt de gefietste critical power ( $GCP$ ) genoemd. Het is belangrijk te beseffen dat deze gefietste critical power per dag wordt bepaald. Als de renner die dag heeft gefietst, kan de  $GCP$  worden bepaald. De critical power echter is een constante. Deze waarde staat vast voor iedere renner. In de modellering wordt wel de vaste waarde voor  $W'$  gebruikt. De  $GCP$  is een indicatie voor hoe goed de renner heeft gereden die dag. Herinner dat het hebben van een hogere critical power,  $CP$ , inhoudt dat je gedurende een oneindige periode een hoger vermogen kunt fietsen dan bij een lagere critical power. Er kan dus meer vermogen geleverd worden zonder moe te worden. Ook



Figuur 4.9: De  $W'$ balans voor verschillende waarden van de  $CP$  voor renner 1 op dag 1407.

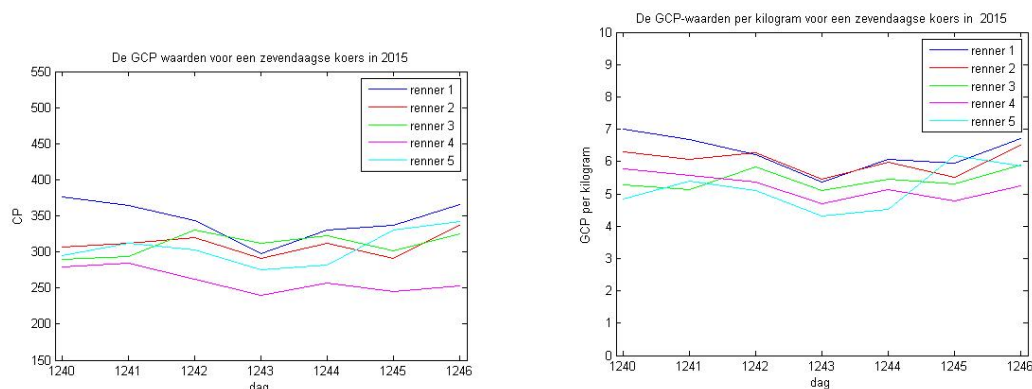
raakt bij eenzelfde vermogensoutput de batterij minder snel leeg op het moment dat de  $CP$  van een renner hoger ligt, figuur 4.9 laat dit zien.

Wanneer er met modelleren gevonden wordt dat de  $GCP$  hoger is dan de dagen ervoor, zou gesteld kunnen worden dat de renner beter reed dan de dagen ervoor. De  $GCP$  is een maat van prestatie. De  $GCP$  van renners kan bekeken worden over een langere tijd, of er kunnen renners vergeleken worden die dezelfde koers hebben gereden. Figuur 4.10 en 4.11 tonen de  $GCP$ 's voor renner 1 en 2 gedurende het jaar 2015.



Figuur 4.10: De  $GCP$ 's voor renner 1 in 2015.      Figuur 4.11: De  $GCP$ 's voor renner 2 in 2015.

Deze figuren laten zien dat de waarde van  $GCP$  erg kan verschillen per dag. Sommige dagen zit de waarde in de buurt van de  $CP$  van de renner, andere dagen hier ver onder. Er is zelfs een aantal uitschieters naar beneden te vinden. Dit zijn zeer waarschijnlijk dagen dat er niet hard gereden is in de wedstrijd. Wielerkenners zouden dit een wandeletappe noemen. Herinner dat de  $CP$  voor renner 2 371 is. Op dag 1350 is de  $GCP$  hoger is dan de  $CP$ , namelijk  $GCP = 387$ . Theoretisch zou dit onmogelijk zijn, een renner kan niet opeens meer energie hebben. In praktijk zouden de omstandigheden deze dag ideaal geweest kunnen zijn, kan er een kleine kalibratiefout in de metingen van het vermogen zijn geweest of er is sprake geweest van het gebruik van verboden middelen. Ook kan het zo zijn dat een renner op termijn beter wordt. Dit zijn de veranderingen waar wielploegen naar op zoek zijn. De critical power kan op de langere termijn met een kleine marge verbeteren. Dit is iets wat door het bepalen van de  $GCP$  gedetecteerd kan worden. Zoals al eerder aangegeven, kunnen ook de waarden voor  $GCP$  bepaald worden voor een meerdaagse wedstrijd die meerdere renners hebben gereden. Dan kunnen de renners vergeleken worden. Figuur 4.12a toont een dergelijke plot.



(a) De  $GCP$  waarden voor vijf renners in dezelfde koers      (b) De  $GCP$  per kilogram waarden voor vijf renners in dezelfde koers

Figuur 4.12:  $GCP$  waarden voor een koers.

Hier zijn de *GCP* waarden van vijf renners te zien, zij hebben dezelfde zevendaagse koers gereden. Merk op dat het lijkt alsof renner 4 minder goed is dan de rest. Zijn *GCP* waarden zijn immers lager dan die van de andere renners. Toch is dat niet het geval. Het is namelijk zo dat deze koers verreden is in de bergen. Er moet dus gekeken worden naar de *GCP* per kilogram. Deze is weergegeven in figuur 4.12b.

In de *GCP* data kan gekeken worden naar veranderingen, zodat er veranderingen gedetecteerd kunnen worden in de prestatie van de wielrenner. In het volgende hoofdstuk worden methoden besproken waarmee veranderingen gedetecteerd kunnen worden en in hoofdstuk 6 wordt dit toegepast op de verkregen *GCP*'s.

## 5 Veranderingen detecteren - de theorie

### 5.1 Changepoint Analyse

#### Wiskundige achtergrond

De changepoint (Nederlandse vertaling: Veranderingspunt) analyse is het proces van het detecteren van veranderingen in verdelingen binnen tijdsgoedende observaties. De changepoint analyse is belangrijk in veel verschillende gebieden, denk aan de bioinformatica waar changepoint analyse wordt gebruikt om genen te identificeren die geassocieerd worden met bepaalde ziekten of aan de signaalverwerking, waar de changepoint analyse wordt gebruikt om significante veranderingen te detecteren in een stroom van afbeeldingen [18].

Meer formeel, bekijk een geordende rij data,  $y_{1:n} = (y_1, \dots, y_n)$ . In deze dataset is er een changepoint als er een tijdstip  $\tau \in \{1, \dots, n-1\}$  bestaat zodanig dat de statistische eigenschappen van  $(y_1, \dots, y_\tau)$  en  $(y_{\tau+1}, \dots, y_n)$  op een bepaalde manier verschillend zijn. Als dit idee van één changepoint uitgebreid wordt naar meervoudige veranderingen, geeft dit  $m$  changepoints, met de posities  $\tau_{1:m} = \{\tau_1, \dots, \tau_m\}$ . Elk changepoint is een geheel getal tussen 1 en  $n-1$ . Definieer  $\tau_0 = 0$  en  $\tau_{m+1} = n$  en neem aan dat de changepoints geordend zijn zodanig dat  $\tau_i < \tau_j$  dan en slechts dan als  $i < j$ . De  $m$  changepoints delen de data op in  $m+1$  segmenten, waarbij het  $i^e$  segment de data  $y_{(\tau_{i-1}):\tau_i}$  bevat. Elk segment wordt samengevat door een set van parameters. Deze parameters worden geassocieerd met het  $i^e$  segment en worden genoteerd door  $(\theta_i, \varphi_i)$ .  $\varphi_i$  is de verzameling van nuisance (Nederlandse vertaling: hinder) parameters en  $\theta_i$  is de verzameling van parameters die de daadwerkelijke verandering bevatten.

Dat wat men bij de changepoint analyse wil weten is hoeveel changepoints een bepaalde verzameling bevat en wat de waarde is die bij ieder changepoint hoort. Deze waarde bevindt zich in de parameter  $\theta_i$ .

#### Enkelvoudige changepoint detectie

De detectie van een enkel changepoint kan beschouwd worden als een hypothesetoets. De nulhypothese  $H_0$  correspondeert met het geval dat er geen changepoint is,  $m = 0$ , en de alternatieve hypothese,  $H_1$ , met het geval dat er één changepoint is,  $m = 1$ . De mogelijkheid om de likelihood-ratiotest (ook wel aannemelijkheidsquotienttoets) te gebruiken om changepoints te detecteren is voorgesteld door Hinkley [9]. Door de test statistic te construeren kan beslist worden of er een verandering in de verdeling van de data heeft opgetreden. Deze likelihoodratio methode vereist het berekenen van de maximum log-likelihood onder zowel de nulhypothese als de alternatieve hypothese. Voor de nulhypothese geldt dat  $\text{lik}(\theta) = p(y_{1:n}|\theta)$  met  $p(\cdot)$  de kansdichtheidsfunctie geassocieerd met de verdeling van de data. Dit geeft voor de maximum log-likelihood  $\log p(y_{1:n}|\hat{\theta})$ , met  $p(\cdot)$  de kansdichtheidsfunctie geassocieerd met de verdeling van de data en  $\hat{\theta}$  de meest aannemelijke schatter (ofwel maximum-likelihood-schatter) van de parameter  $\theta$ . Onder de alternatieve hypothese geldt dat er een model wordt beschouwd met een changepoint op  $\tau_1$ , met  $\tau_1 \in \{1, 2, \dots, n-1\}$ . De likelihood voor een gegeven  $\tau_1$  is  $\text{lik}(\theta_1, \theta_2) = p(y_{1:\tau_1}|\theta_1) \cdot p(y_{(\tau_1+1):n}|\theta_2)$ . Dit geeft voor de maximum log-likelihood voor die gegeven  $\tau_1$

$$\begin{aligned} ML(\tau_1) &= \log[p(y_{1:\tau_1}|\hat{\theta}_1) \cdot p(y_{(\tau_1+1):n}|\hat{\theta}_2)] \\ &= \log(p(y_{1:\tau_1}|\hat{\theta}_1)) + \log(p(y_{(\tau_1+1):n}|\hat{\theta}_2)) \end{aligned} \quad (5.1)$$

Omdat de locatie van het changepoint een discrete waarde aanneemt, kan simpelweg gemaximaliseerd worden over alle mogelijke waarden van  $\tau_1$  om de maximale log-likelihood waarde onder de alternatieve hypothese te vinden. De log-likelihood statistic is gelijk aan min tweemaal likelihood onder de nulhypothese gedeeld door de maximale likelihood onder de alternatieve hypothese. Dit geeft dus voor de test statistic

$$\begin{aligned} \lambda &= -2 \log \left( \frac{p(y_{1:n}|\hat{\theta})}{\max_{\tau_1} [p(y_{1:\tau_1}|\hat{\theta}_1) \cdot p(y_{(\tau_1+1):n}|\hat{\theta}_2)]} \right) \\ &= -2 \log(p(y_{1:n}|\hat{\theta})) + 2 \max_{\tau_1} \log(p(y_{1:\tau_1}|\hat{\theta}_1) \cdot p(y_{(\tau_1+1):n}|\hat{\theta}_2)) \\ &= 2 \left[ \max_{\tau_1} ML(\tau_1) - \log p(y_{1:n}|\hat{\theta}) \right] \end{aligned} \quad (5.2)$$

Deze test leidt tot het kiezen van een drempel  $c$ , zodanig dat de nulhypothese verworpen wordt als  $\lambda > c$ . Deze  $c$  moet bepaald, omdat een heel klein verschil in gemiddelde, wat een lage test statistica zal geven, niet gewenst is. Het verschil moet significant zijn en daar zorgt deze  $c$  voor. Als de nulhypothese verworpen wordt, en er dus een changepoint gedetecteerd wordt, is  $\hat{\tau}_1$  de schatter voor de waarde van  $\tau_1$  die  $ML(\tau_1)$  maximaliseert. Voor het kiezen van een geschikte waarde voor  $c$  is nog geen concrete theorie [11]. Door trial-and-error kan deze waarde bepaald worden.

### Meervoudige changepoint detectie

De likelihood-ratiotest kan uitgebreid worden naar meervoudige changepoints door de likelihood simpelweg te sommeren over elk mogelijk aantal van  $m$  changepoints. Het probleem hierbij is dat het maximum van  $ML(\tau_1)$  bekeken moet worden van alle mogelijke combinaties  $\tau_{1:m}$  en dit zijn er erg veel. Dus zo simpel is dit nog niet. De vergelijking die geminimaliseerd dient te worden om dit probleem op te lossen, luidt

$$\sum_{i=1}^{m+1} [\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i})] + \beta f(m) \quad (5.3)$$

Hier is  $\mathcal{C}$  de kostenfunctie voor een segment en  $\beta f(m)$  de penalty, een meervoudige variant van de drempel  $c$  voor enkelvoudige changepoint detectie. In het geval van een enkel changepoint wordt voor  $\mathcal{C}$  tweemaal de min log-likelihood gekozen en ook in het meervoudige geval is dit een veelgebruikte functie [12]. Om deze reden wordt deze keuze nu ook gemaakt. Terugkomend op de keuze van de penalty, de meest voorkomende keuze is een penalty die lineair is in het aantal changepoints, dus  $\beta f(m) = \beta m$ . Dit geeft dus alleen nog een keuze voor  $\beta$ . Op de keuze van  $\beta$  wordt ingegaan in hoofdstuk 5.3.

Voor het berekenen van de changepoints in R wordt gebruik gemaakt van het package changepoint [13]. Dit package bevat de PELT-methode, die meervoudige changepoints kan detecteren. Deze methode is opgebouwd uit 3 andere methoden, en al deze vier methoden worden verder uitgewerkt.

De methode die het meeste gebruikt wordt om meervoudige changepoints te bepalen is de binary segmentation methode (Nederlandse vertaling: tweevoudige verdeling methode), ontwikkeld door Scott en Knott [23]. Deze methode begint met het bepalen van een enkel changepoint in de gehele dataset. Er wordt dus getest of er een  $\tau$  bestaat die voldoet aan

$$\mathcal{C}(y_{1:\tau}) + \mathcal{C}(y_{(\tau+1):n}) + \beta < \mathcal{C}(y_{1:n}). \quad (5.4)$$

Merk op dat deze ongelijkheid gelijk is aan vergelijking (5.2) voor  $\mathcal{C}(\cdot)$  is tweemaal de negatieve log-likelihood. De data wordt in twee segmenten gesplitst en de methode om een enkelvoudig changepoint te bepalen wordt toegepast op deze beide segmenten en op ieder verder ontstane segment, totdat er geen changepoints meer gedetecteerd kunnen worden [23]. Deze methode benadert de minimalisering van (5.3) met  $f(m) = m$ . Hoewel deze methode niet exact is, is deze wel erg snel, omdat deze maar  $2^{n-1}$  mogelijke oplossingen heeft, met een dataset van lengte  $n$ . De computationele complexiteit is  $\mathcal{O}(n \log n)$ . Merk op dat deze snelheid wel ten koste gaat van het feit dat deze methode een benadering geeft en geen exacte waarde.

Een tweede methode komt van Auger en Lawrence [1], is verder onderzocht door Bai en Perron [2] en heet de segment neighborhood algorithm (Nederlandse vertaling: segment burens algoritme). Dit is een exacte methode die alle mogelijke segmenten afgaat en op die manier een minimalisatie van (5.3) zoekt. Wel is een bovengrens voor het aantal segmenten een vereiste binnen deze methode. Een voordeel van deze methode is dat er de mogelijkheid is een penalty van de vorm  $\beta f(m)$  te gebruiken. Een consequentie echter van het feit dat deze methode alle mogelijke segmenten af moet gaan, is dat deze methode niet erg snel is, namelijk  $\mathcal{O}(n^3)$ .

De optimal partitioning methode (Nederlandse vertaling: optimale verdeling) is voorgesteld door Yao [29] en Jackson [10] en minimaliseert

$$\sum_{i=1}^{m+1} [\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i})] + \beta m. \quad (5.5)$$



Dit is gelijk aan vergelijking (5.3), maar dan met meest voorkomende keuze  $f(m) = m$ . Deze methode maakt gebruik van het feit dat een optimalisatie die vergaard is uit een deelverzameling van de informatie, nooit optimaler kan worden in het grote geheel. Deze methode begint met conditioneren op het laatste punt van verandering. De methode geeft de optimale waarde van de kostenfunctie voor de optimale verdeling tot dit laatste changepoint en de kosten voor het segment van het laatste changepoint tot het einde van de data. Herinner dat geldt  $\tau_0 = 0$  en  $\tau_{m_n+1} = n$ . Noem  $F(s)$  de minimalisatie van (5.5) voor de data  $y_{1:s}$  met  $\mathcal{T}_s = \{\tau \mid 0 = \tau_0 < \tau_1 < \dots < \tau_m < \tau_{m_s+1} = s\}$  de verzameling van mogelijke vectoren van changepoints. Noem  $F(0) = -\beta$ , dan volgt dat

$$\begin{aligned} F(s) &= \min_{\tau \in \mathcal{T}_s} \left\{ \sum_{i=1}^{m+1} [\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) + \beta] \right\} \\ &= \min_t \left\{ \min_{\tau \in \mathcal{T}_t} \sum_{i=1}^m [\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) + \beta] + \mathcal{C}(y_{(t+1):s}) + \beta \right\} \\ &= \min_t \{ F(t) + \mathcal{C}(y_{(t+1):s}) + \beta \} \end{aligned} \quad (5.6)$$

Dit geeft de minimale kosten voor de data  $y_{1:s}$  in termen van de minimale kosten voor de data  $y_{1:t}$  met  $t < s$ . Deze recursie kan een stuk sneller opgelost worden, namelijk met complexiteit  $\mathcal{O}(n^2)$ . Hoewel de snelheid van de optimal partitioning methode sneller is dan die van de neighborhood segmentation, zijn deze nog verre van even snel als de binary segmentation. De pruning exact linear time (PELT, Nederlandse vertaling: snoeien exact lineaire tijd) methode is een aanpassing van de optimal partitioning methode, die resulteert in een exacte minimalisatie, én met snelheid  $\mathcal{O}(n)$ . Dit algoritme minimaliseert nog steeds vergelijking (5.5). De essentie van het snoeien is om die waarden van  $\tau$  te verwijderen die nooit een minimum kunnen vormen voor de  $F(t)$  uit vergelijking (5.6). Onder aanname van de volgende stelling mag er gesnoeid worden:

#### Stelling

Neem aan dat het introduceren van een changepoint binnen een rij observaties de kosten  $\mathcal{C}$  van een rij reduceert, er bestaat dus een constante  $\beta$  zodanig dat voor alle  $t < s < T$  geldt

$$\mathcal{C}(y_{(t+1):s}) + \mathcal{C}(y_{(s+1):T}) + \beta \leq \mathcal{C}(y_{(t+1):T}) \quad (5.7)$$

Als dan op een later tijdstip  $T > s$  geldt dat

$$F(t) + \mathcal{C}(y_{(t+1):s}) + \beta \geq F(s) \quad (5.8)$$

kan  $t$  nooit het optimale laatste changepoint zijn dat voorafgaat aan  $T$ .

Het bewijs van deze stelling maakt goed duidelijk waarom de stelling geldt. Wanneer aangenomen wordt dat (5.8) geldt, dan geldt ook

$$F(t) + \mathcal{C}(y_{(t+1):s}) + \beta + \mathcal{C}(y_{(s+1):T}) \geq F(s) + \mathcal{C}(y_{(s+1):T}).$$

Wegens (5.7) geldt dat

$$F(t) + \mathcal{C}(y_{(t+1):T}) \geq F(s) + \mathcal{C}(y_{(s+1):T})$$

maar dan is er dus een optimaler changepoint  $s$  dat zich na  $t$  bevindt, want er gold  $0 < t < s < T$ . Maar dan kan  $t$  nooit het laatste optimale changepoint voor  $T$  zijn. Voor een uitgebreider bewijs is voor de geïnteresseerde lezer te vinden in de extra materialen bij de paper van Killick [12].

## 5.2 Taut string methode

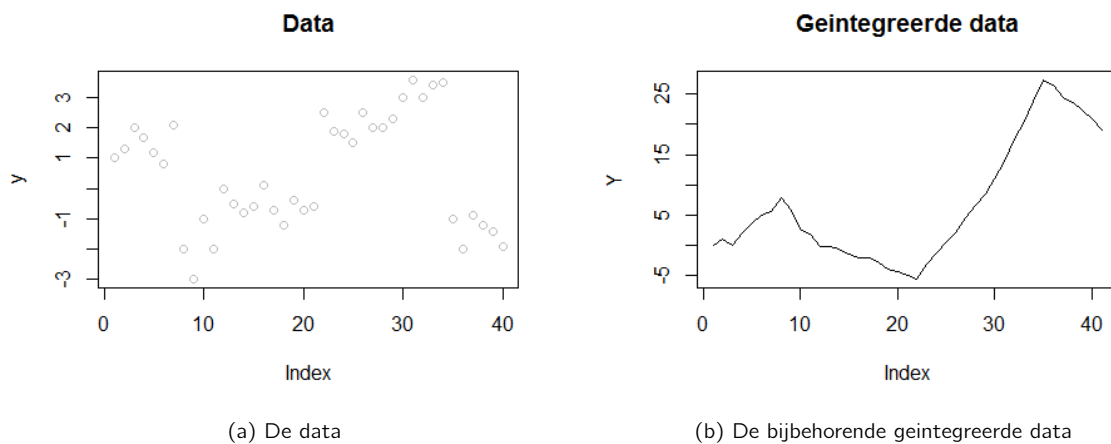
Een tweede methode om een verandering te kunnen detecteren is de taut string methode (Nederlandse vertaling: strak koord). Barlow et al. [3] waren de eersten die een dergelijke methode beschreven en

onder andere Davies en Kovac [7] ontwikkelden deze verder in de toepassing van de statistiek. Het idee van deze methode is om een stapsgewijs constante functie  $f$  te creëren welke de dataset beschrijft met behulp van een iteratief proces met lineaire interpolatie tussen zogeheten designpunten. Beschouw weer de rij met tijdgeordende data  $y_{1:n} = (y_1, \dots, y_n)$ . Er wordt gezocht naar een functie  $f$  die deze data beschrijft. Daartoe wordt eerste gekeken naar geïntegreerde data. De designpunten worden gedefinieerd door het nemen van de partiële sommen van  $y$ .

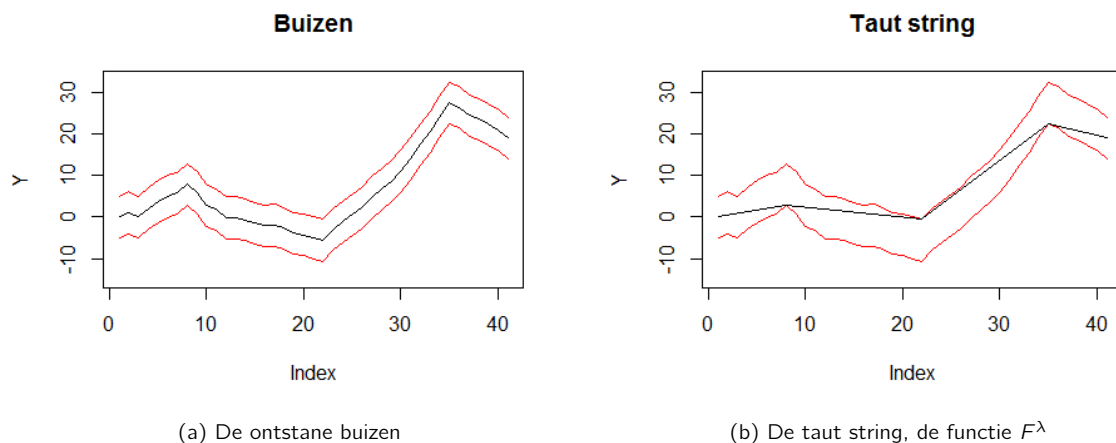
$$Y_0 = 0$$

$$Y_j = \sum_{i=1}^j y_i \quad (j = 1, \dots, n) \quad (5.9)$$

Deze designpunten creëren de functie  $Y$ . Figuur 5.1 toont een voorbeeld van de constructie van  $Y$ .  $Y$  is



Figuur 5.1: Het integreren van de data. Het linkerpaneel laat een kleine gesimuleerde dataset zien en het rechterpaneel toont de partiële sommen.



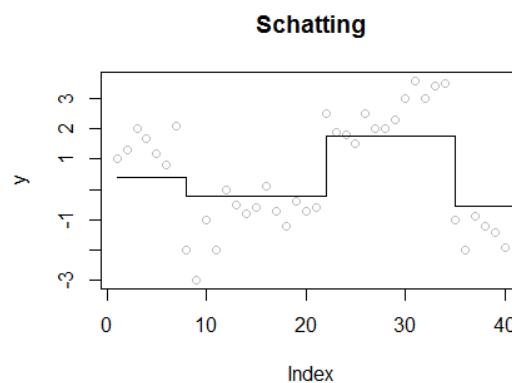
Figuur 5.2: Het linkerpaneel laat de ontstane buizen zien en het rechterpaneel de ontstane taut string, door het straktrekken van het koord.

de basis voor een buis door het introduceren van een  $\lambda$ . De buis ontstaat door de functies  $Y - \lambda$  en  $Y + \lambda$ , dit is te zien in figuur 5.2a. Nu volgt de kern van de taut string methode. Beschouw de lijn  $Y$  als een koord.

Trek nu aan de uiteinden aan dit koord, zodanig dat dit strak staat in de buis, zoals figuur 5.2b toont. Het koord kan analytisch gedefinieerd worden als de functie  $F^\lambda$  op het interval  $[a, b]$  zodanig dat de lengte van het koord minimaal is, dus

$$\text{lengte}(F^\lambda) = \int_a^b \sqrt{1 + ((F^\lambda)'(t))^2} dt \quad (5.10)$$

$F^\lambda$  is stuksgewijs lineair. Deze functie hangt af van  $\lambda$  omdat de grootte van  $\lambda$  invloed heeft op de functie, en wel op de hoeveelheid pieken. Hierover volgt later meer. Door deze  $F^\lambda$  te differentiëren ontstaat  $f^\lambda$ , en dat is de functie die een schatting geeft voor de verschillen in de gemiddelden van data. Deze functie is te zien in figuur 5.3.



Figuur 5.3: De schatting voor de functie  $f^\lambda$

Om functies als  $f^\lambda$  te creëren, moet de volgende vergelijking geminimaliseerd worden

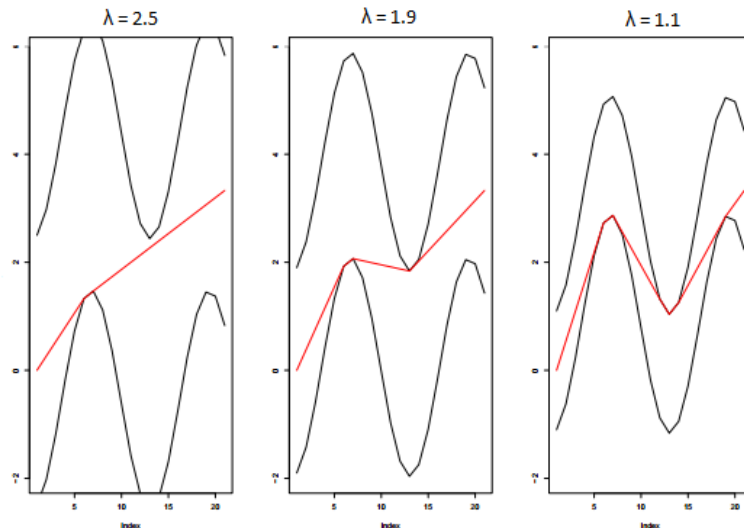
$$T(f) = \sum_{i=1}^n (y_i - f_i)^2 + \lambda \sum_{i=1}^{n-1} |f_{i+1} - f_i| \quad (5.11)$$

Het eerste gedeelte is de kleinste kwadratenmethode om de afstand tussen  $f_i$  en  $y_i$  te minimaliseren. De  $\lambda$  is gelijk aan de  $\lambda$  die de breedte van de buis beschrijft. Het tweede gedeelte van (5.11) minimaliseert de totale variatie van  $f$ . De totale variatie is de afstand die afgelegd is op de y-as als de functie gevolgd is over de x-as. De makkelijkste functie om de totale variatie van te berekenen is een rechte lijn, de totale variatie van zo'n functie is  $|f(x_{\max}) - f(x_{\min})|$ . Omdat het koord zo strak mogelijk moet staan, moeten de totale variatie ook geminimaliseerd worden. Uit de literatuur blijkt dat de taut string methode het snelste algoritme is om (5.11) op te lossen [14]. De geïnteresseerde lezer wordt doorverwezen naar de paper van Mammen en van der Geer [15], waar ook dieper ingegaan wordt op de link tussen de taut string en vergelijking (5.11).

De ontstane functie  $f^\lambda$  is stuksgewijs constant. De punten waar de functie 'overspringt' naar een ander gemiddelde, heten de knopen. Deze knopen zijn dus de punten waar de verandering optreedt. Het is voor de taut string methode mogelijk om een oplossing te bepalen in  $\mathcal{O}(n)$  stappen. Dit maakt deze methode een concurrent voor de PELT methode. In hoofdstuk 5.3 wordt verder ingegaan op de vergelijking tussen de twee methoden. Een andere eigenschap van de taut string methode is dat de modaliteit (het aantal pieken of relatieve minima/maxima) toeneemt als de buis kleiner wordt, dit is geïllustreerd in figuur 5.4.

Ook kan gesteld worden dat  $f^\lambda$  een lokaal maximum aanneemt op  $[t_{i+1}, t_j]$  als de taut string de bovengrens raakt in  $i$  en de ondergrens in  $j$  en tussendoor geen van de grenzen meer aanraakt en kan gesteld worden dat  $f^\lambda$  een lokaal minimum aanneemt op  $[t_{k+1}, t_m]$  als de taut string de ondergrens raakt in  $k$  en de bovengrens in  $m$  en tussendoor geen van de grenzen meer raakt.

De taut string is ook in R geïmplementeerd in R, in het package `ftnonpar` [6].

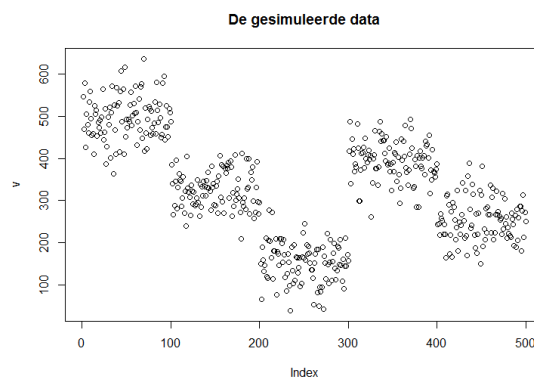


Figuur 5.4: Visuele weergave van het feit dat er meer pieken ontstaat op het moment dat de buisbreedte kleiner wordt<sup>5</sup>.

### 5.3 Het vergelijken van de twee methoden

Voordat de gegenereerde data van hoofdstuk 4.2 toegepast wordt op de modellen om verandering te detecteren, worden de twee methoden met elkaar vergeleken. Dit vergelijken wordt gedaan door het simuleren van een aantal datasets en hierop zowel de changepoint analyse als de taut string methode toe te passen en vervolgens te kijken wat iedere methode teruggeeft.

Voor de eerste simulatie wordt een dataset  $y$  van lengte 500 met waarden uit een normale verdeling genomen. Alle  $y_i, i = (1, \dots, 500)$  hebben  $\sigma = 50$ , alleen het gemiddelde is verschillend. Voor  $y_{1:100}$  geldt  $\mu = 500$ , voor  $y_{101:200}$  geldt  $\mu = 350$ , voor  $y_{201:300}$  geldt  $\mu = 150$ , voor  $y_{301:400}$  geldt  $\mu = 400$  en voor  $y_{401:500}$  geldt  $\mu = 250$ . Figuur 5.5 geeft een weergave van de gegenereerde data. Gezien het feit dat het gemiddelde van de verdelingen van deze data heel erg verschilt, valt te verwachten dat de methoden de punten van verandering terug zouden moeten geven.



Figuur 5.5: Gesimuleerde data,  $n = 500$ , voor een vergelijking tussen de twee methoden.

Bij het gebruik van de PELT methode in R, om changepoints te bepalen, moet waarde van  $\beta$  ingegeven worden, dit is de zogeheten penalty. Omdat er voor de keuze van de penalty geen literatuur beschikbaar is, gaat deze keuze via trial-and-error. Voor een penalty van 27.000 tot 1.000.000 is de methode in staat de

<sup>5</sup>Bewerking van <http://www.maths.bris.ac.uk/~maxak/ast.pdf>

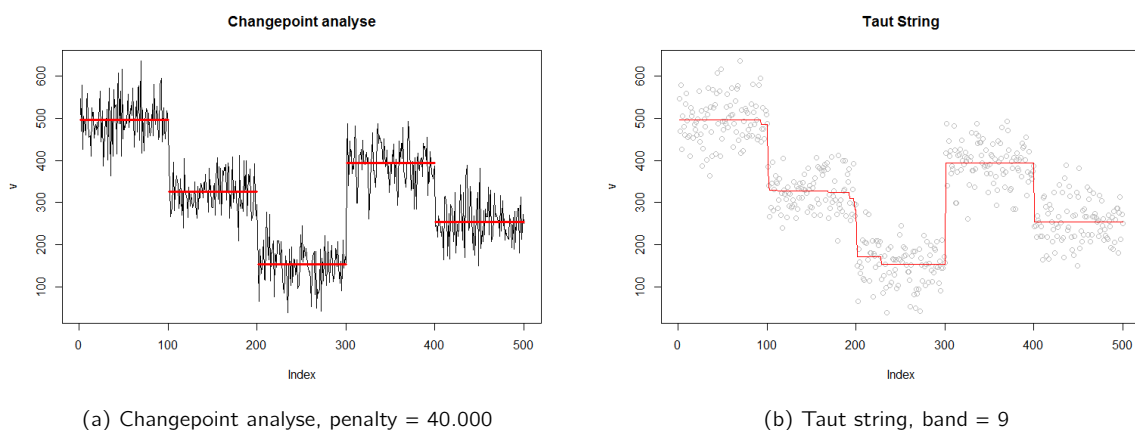
precieze changepoints te bepalen, de methode retourneert als locaties

```
> cpts(y.pelt)
[1] 100 200 300 400
```

met als waarden voor de gemiddelden

```
> coef(y.pelt)$mean
[1] 496.4667 325.1262 152.7062 394.4564 254.0693
```

Merk op dat deze gemiddelden netjes in de buurt liggen van de eerder gekozen  $\mu$ 's.



Figuur 5.6: De afbeeldingen behorende bij de twee verschillende manieren om veranderingen te detecteren.

De taut string methode kan in R bepaald worden met het package `ftnonpar` [6]. De waarde van de  $\lambda$  moet hierbij ook ingegeven worden. In dit package heet deze waarde de band. Voor de taut string methode is de keuze voor de band is 4 tot 9 de optimale. Dit geeft steeds eenzelfde aantal knopen, namelijk onder de indices

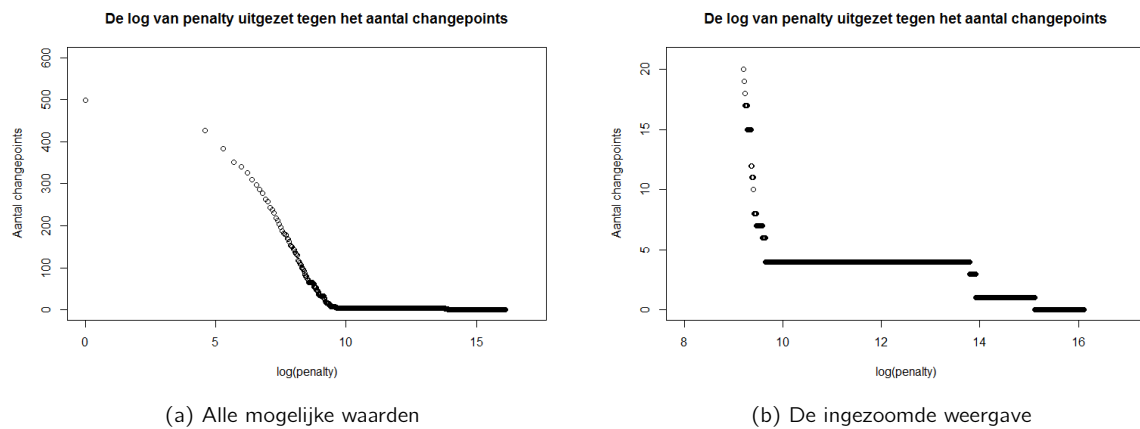
```
> tmp$knotsind
[1] 1 101 102 103 112 169 193 198 199 201 301
[12] 401 501
```

met als gemiddelden

```
> ygem
[1] 496.4667 384.1216 340.7796 329.5810 327.2895
[6] 323.1317 310.0739 296.9451 281.7570 172.0869
[11] 394.4564 254.0693
```

Figuur 5.6 toont de grafische weergave van het detecteren van de veranderingen op beide manieren. Een eerste observatie is dat de changepoint analyse de correcte indices teruggeeft. Ook voor andere keuzes van  $y$  geeft deze netjes de 4 punten terug. De taut string methode geeft de juiste indices terug, maar geeft ook nog tussenliggende waarden. Een reactie hierop zou zijn de band te vergroten, zodat er minder pieken ontstaan, dit gaat echter ten koste van het verschil in gemiddelde tussen de waarnemingen bij de indices 300 tot 500. Wanneer er voor de band meer dan 10 wordt gekozen, herkent de methode niet de knoop bij index 400. Het lijkt erop dat de taut string methode aan de ene kant gevoeliger is, maar aan de andere kant ook bepaalde missers maakt. Deze methode detecteert namelijk kleinere verschillen, maar is niet in staat de 4 gewenste knopen te retourneren.

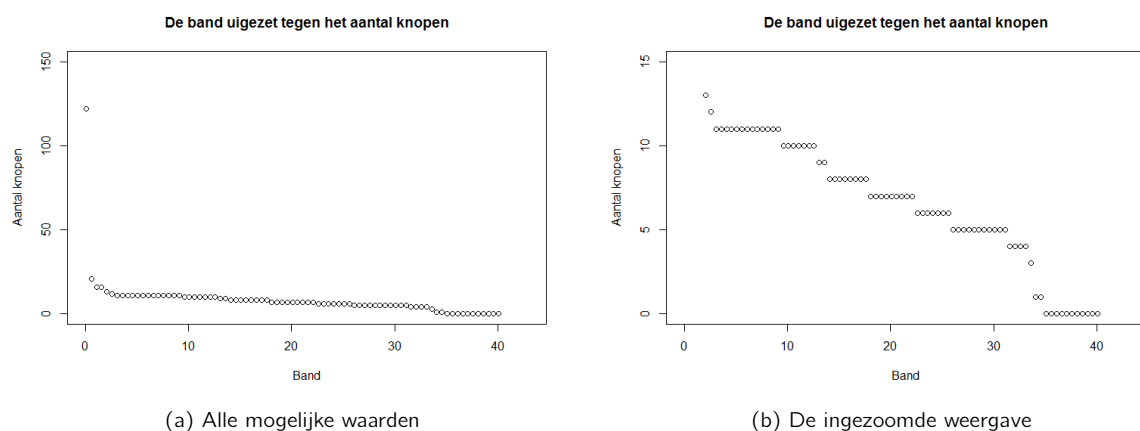
Het is goed om te onderzoeken hoe de grootte van de penalty dan wel band invloed heeft op het aantal changepoints dan wel knopen. Figuur 5.7 toont een plot van de  $\log(\text{penalty})$  uitgezet tegen het aantal changepoints voor de data, gebruikt bij de bovenstaande simulatie.



Figuur 5.7: Figuren die inzicht geven in het verband tussen de penalty en het aantal changepoints.

Merk op dat het interval aan penalties dat vier changepoints terug zal geven, erg groot is. Er is dus een brede range aan penalties die gekozen kunnen worden, zodanig dat de gewenste changepoints terug gegeven worden. Dit is waardevolle informatie voor de toepassing die nog volgt.

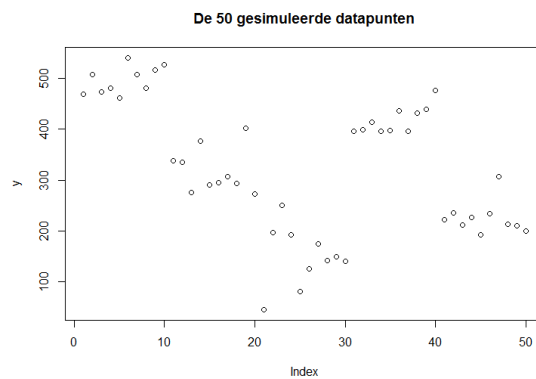
Voor de taut string kunnen eenzelfde soort plots gemaakt worden, zie figuur 5.8.



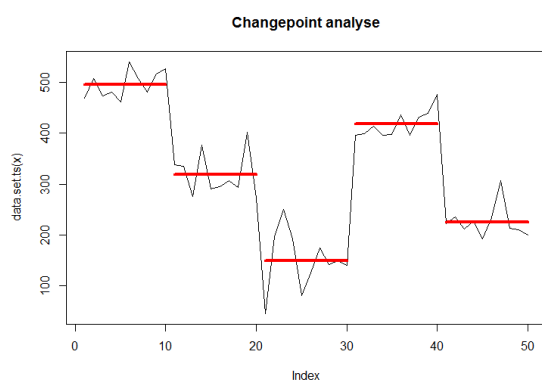
Figuur 5.8: Figuren die inzicht geven in het verband tussen de band en het aantal knopen.

Merk op dat er nu een minder duidelijk geheel is. In de wetenschap dat de keuze van een de band is 11 de knoop rond 400 niet meer detecteert, kunnen hier geen duidelijke conclusies getrokken worden. Wel is het interval  $[3,9]$  het grootste interval.

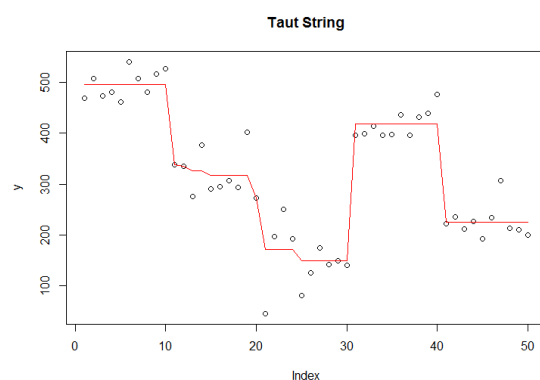
Een volgende vergelijking van de data vindt plaats door te kijken naar een kleinere dataset. In plaats van 100 random waarden voor iedere  $\mu$  te nemen, worden er nu 10 van elke gekozen, dus in totaal heeft de dataset lengte 50. Figuur 5.9 toont de dataset en de detectie van de veranderingen.



(a) De gegenereerde data, n=50



(b) Changepoint Analyse, penalty = 40.000



(c) Taut String, band = 9

Figuur 5.9: De afbeeldingen behorende bij de twee verschillende manieren om veranderingen te detecteren.

Merk ten eerste op dat de waarde van de penalty en de band onveranderd zijn gebleven. De grootte van de dataset heeft hier geen invloed op. Het is belangrijk om dit te weten. Verder retourneert de changepoint de locaties en waarden:

```
> cpts(y.pelt)
[1] 10 20 30 40
> coef(y.pelt)$mean
[1] 496.0862 318.8767 149.9226 418.0482 225.5913
```

en de taut string methode noteert voor de knopen en diens waarden retourneren

```
> tmp$knotsind
[1] 1 11 12 13 15 20 21 31 41 51
> ygem
[1] 469.40235 337.74338 334.81856 276.49298 291.31335 273.48520 45.42263
395.45800 221.78486
```

Wanneer ervoor gekozen wordt om de  $\mu$ 's van de trekkingen uit de normale verdelingen dicht bij elkaar te leggen, is het wel nodig om de penalty en band aan te passen. Zowel de penalty als de band dienen dan verlaagd te worden. Wanneer echter de  $\sigma$  dan weer verlaagd wordt, kies bijvoorbeeld 10, dan moet de waarde van band of penalty weer verhoogd worden.

Uit deze vergelijking kunnen een aantal belangrijke dingen geconcludeerd worden. Ten eerste het feit dat de taut string methode gevoeliger is voor het detecteren van veranderingen dan de changepoint analyse, maar

ook slordiger als het aankomt op het vinden van de juiste vier punten. Dit kan een nadeel zijn, de methode lijkt minder nauwkeurig, maar het kan ook een voordeel zijn, omdat we straks met data gaan werken die niet zo mooi verdeeld is als de data die hier nu gebruikt is. Daarnaast is de keuze van de penalty of band erg belangrijk. Er is gezien dat er altijd een interval voor de band of penalty is, welke dezelfde, en ook de beste waarden voor de punten waarop verandering optreedt teruggeeft. De grootte van de dataset heeft geen invloed op de waarde van de band of penalty, alleen de parameters die de verdeling beschrijven hebben hier invloed op.

## 5.4 Uitwerken van de methoden voor een kleine dataset

Om meer gevoel te krijgen bij de beide methoden, worden de minimalisaties voor beide methoden uitgevoerd op een kleine gegenereerde dataset met  $n = 10$ .

Deze 10 datapunten worden gegenereerd uit de normale verdeling met  $\sigma = 0.1$  en een verschillende  $\mu$ , namelijk voor drie punten  $\mu = 3$ , voor de volgende twee punten  $\mu = 2$ , dan twee punten met  $\mu = -1$  en de laatste drie punten met  $\mu = 0.5$ . Omdat het bij deze kleine datasets erg van invloed kan zijn hoe de dataset eruit ziet, is deze hieronder gegeven.

[1] 2.9625822 3.0490213 3.1548573 1.9292974 1.9723562  
 [6] -1.1469867 -0.8741497 0.4505882 0.2804609 0.3857803

Om de changepoint analyse toe te passen wordt de volgende vergelijking geminimaliseerd

$$\text{kosten} = \sum_{i=1}^{m+1} [\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i})] + \beta m, \quad (5.12)$$

met  $y_{1:n}$  de gegenereerde data,  $\mathcal{C}$  de kostenfunctie, welke tweemaal de min log-likelihood is,  $\beta$  de penalty en  $m$  het aantal changepoints.

De drie gewenste changepoints worden teruggegeven bij een penalty van 1. Door een penalty van 2 of 5 worden er twee dan wel één changepoint terug gegeven. Tabel 5.1 geeft een overzicht van de kosten uit vergelijking 5.12 en figuur 5.10 geeft een weergave van de drie verschillende waarden voor de penalty.

Penalty	Aantal changepoints	Kosten
1	3	31.19698
2	2	95.53412
5	1	143.8963

Tabel 5.1: Een overzicht van kosten, afhankelijk van de penalty.

In de tabel is te zien dat het heel erg veel uitmaakt hoeveel changepoints er gedetecteerd worden. De kosten lopen aanzienlijk op bij het missen van een changepoint.

De vergelijking die geminimaliseerd moet worden voor de taut string methode, was

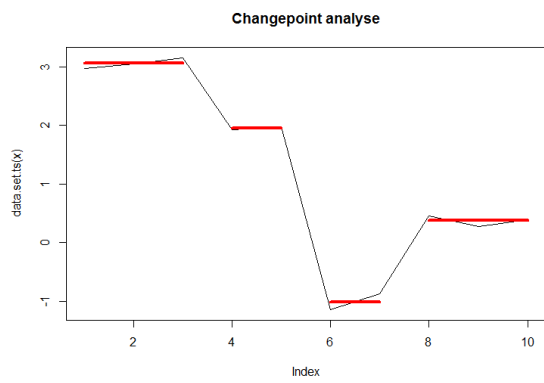
$$T(f) = \sum_{i=1}^n (y_i - f_i)^2 + \lambda \sum_{i=1}^{n-1} |f_{i+1} - f_i|, \quad (5.13)$$

met  $y_i, i \in (1, \dots, n)$  de gegeneerde data,  $\lambda$  de waarde van de band en  $f$  de stuksgewijs constante functie die bepaald diende te worden door middel van de methode.

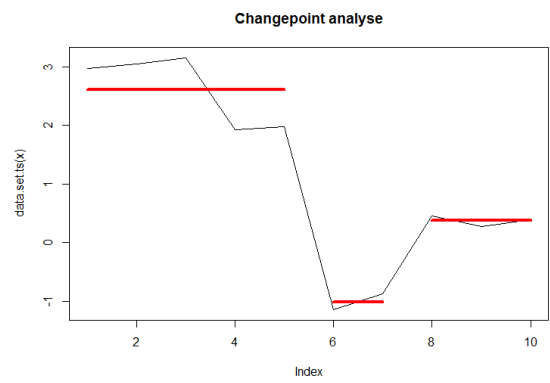
Band	Aantal knopen	Kosten
0.1	3	0.545072
0.2	2	2.366151
0.5	1	5.227722

Tabel 5.2: Een overzicht van kosten, afhankelijk van de band

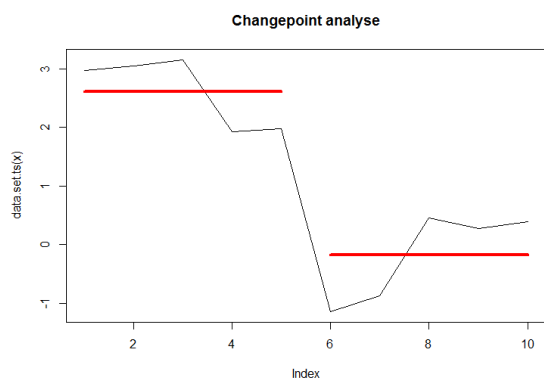




(a) penalty = 1



(b) penalty = 2

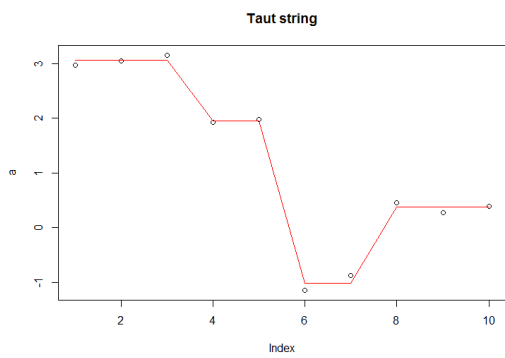


(c) penalty = 5

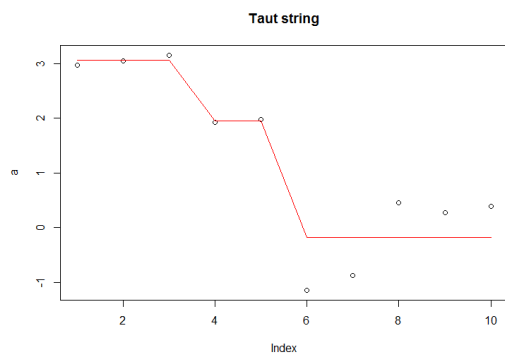
Figuur 5.10: Changepoint analyse voor verschillende waarden van de penalty.

Tabel 5.2 en figuur 5.11 geven weer een overzicht. Ook nu geven de kosten een goed beeld van het optimale aantal knopen.

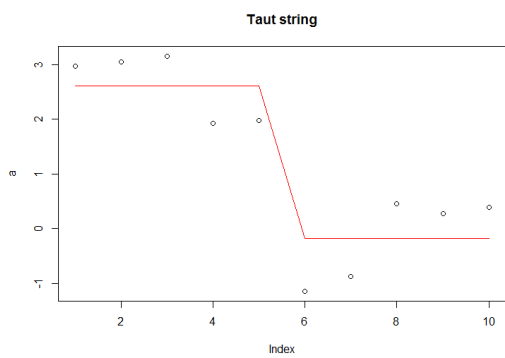
In het bovenstaande geval is de band precies een tiende van de penalty. Dit is toeval. Als er andere datasets gekozen worden, is dit verband niet aanwezig.



(a) band = 0.1



(b) band = 0.2



(c) band = 0.5

Figuur 5.11: Taut string methode voor verschillende waarden van de band

## 6 Veranderingen detecteren - de toepassing

Nu de kennis is vergaard over de twee verschillende methoden om veranderingen te detecteren, kan dit toepast worden op de data die verkregen is door het bepalen van de *GCP*'s.

Wanneer er gezocht wordt naar changepoints in de data, wordt er dus gezocht naar een verandering in statistische eigenschappen van de data. Er moet dus iets gezegd worden over de statistische eigenschappen van de *GCP*'s. Neem aan dat deze *GCP*'s allemaal uit eenzelfde soort verdeling komen, maar dat het gemiddelde (een parameter die de verdeling beschrijft) kan verschillen. Met de changepoint analyse wordt gezocht naar een verandering in het gemiddelde, dit is de  $\theta_i$ . De  $\varphi_i$ , de ruisparameter, wordt 0 gesteld. Het bepalen van de verdeling waaruit deze data komt, zou een onderwerp voor een vervolg kunnen zijn. Dit is namelijk geen simpel antwoord. Eerst dient daartoe de verdeling van het vermogen bepaald te worden, en met de toepassing van extreme waarden theorie en theorieën over het toepassen van functies op verdelingen, kan iets gezegd worden over de precieze verdeling van de *GCP*'s.

Omdat renners tijdens trainingen niet voluit gaan, en tijdens wedstrijden wel, wordt de keuze gemaakt om de *GCP*'s te bekijken die gaan over de wedstrijddagen. Het detecteren van de veranderingen kan naast met de twee methoden, ook nog over allerlei verschillende tijdsintervallen. Er kan gekeken worden naar een zomer, jaar of alle beschikbare data. Figuur 6.1 toont de changepointanalyses voor renner 4 over het jaar 2015. In dit jaar heeft deze renner 81 wedstrijddagen gereden, waarvan de data beschikbaar is. De keuze van de penalty is erg belangrijk, de figuur toont de afbeeldingen voor verschillende keuzes van de penalty.

Er geldt inderdaad dat hoe groter de penalty, hoe minder changepoints. Merk op dat de penalties 4000 en 6000 dezelfde changepoints teruggeven. In de afbeeldingen is te zien dat op de horizontale as de index staat. Dit komt omdat alle dagen waarop een wedstrijd is gereden als het ware achter elkaar zijn geplakt en hierop is de changepoint analyse toegepast. Door middel van het scriptje in bijlage A.5 kan achterhaald worden op welke dag dit changepoint zich bevindt.

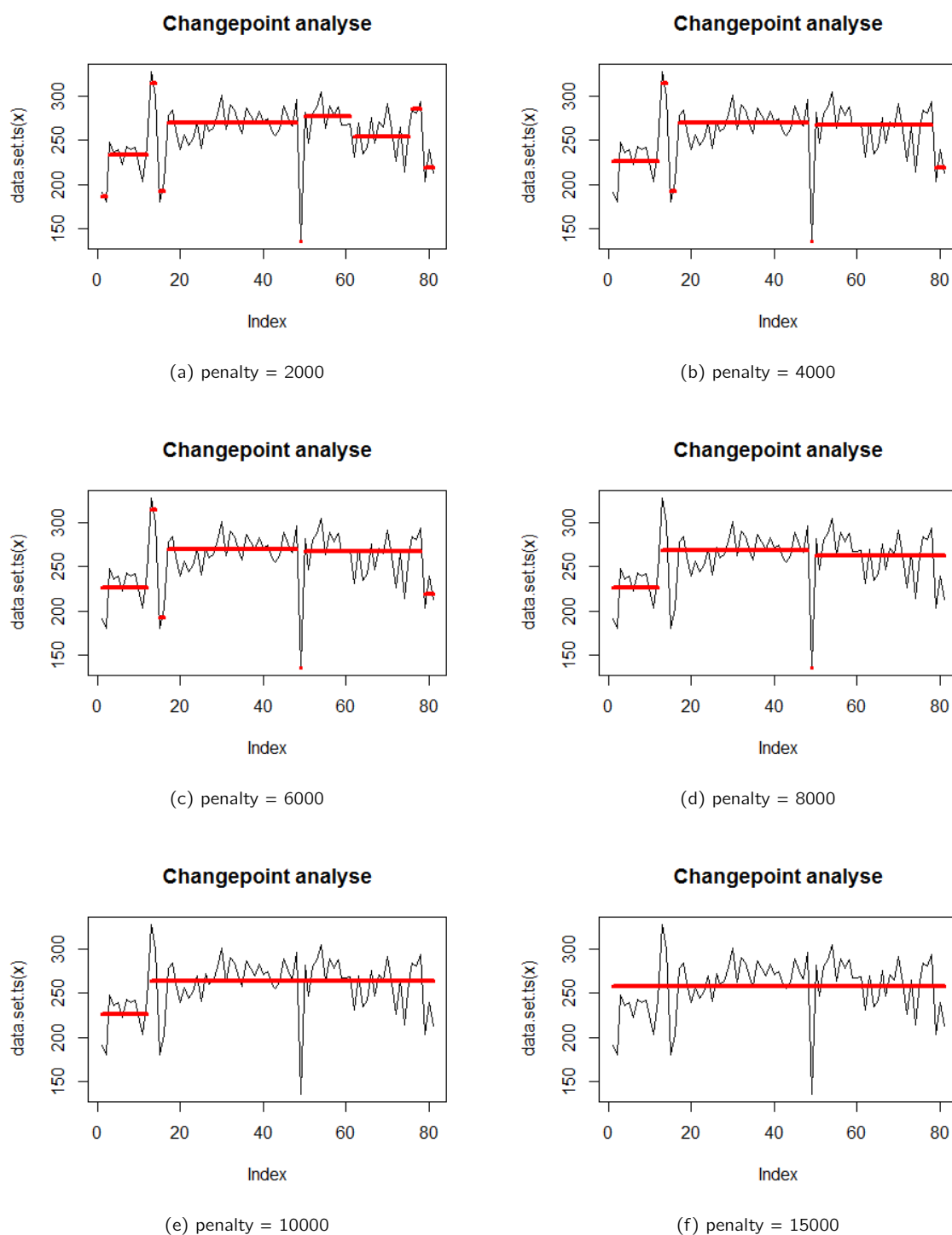
Het bepalen van alle *GCP*'s van de wedstrijden die renner 4 in zijn loopbaan bij de ploeg Giant-Alpecin heeft gereden, levert de plot in figuur 6.2 op. De waarden van 2015 zijn terug te vinden tussen index 150 en index 230. In deze plot zijn duidelijke verschillen te zien in de gemiddelde *GCP* over een bepaalde periode, een overzicht hiervan is te vinden in tabel 6.1

Periode	Index	Dagen	Datum
1	[9,74]	469-708	9 februari 2013-13 oktober 2013
2	[89,147]	835-1080	11 februari 2014 - 14 oktober 2014
3	[148,159]	1081-1203	15 oktober 2014 - 14 februari 2015
4	[163,195]	1217-1323	28 februari 2015 - 14 juni 2015
5	[197,255]	1325-1383	16 juni 2015 - 26 april 2016

Tabel 6.1: De data bij de gedetecteerde changepoints, toegepast op alle *GCP*'s voor de wedstrijden van renner 4, penalty = 6000.

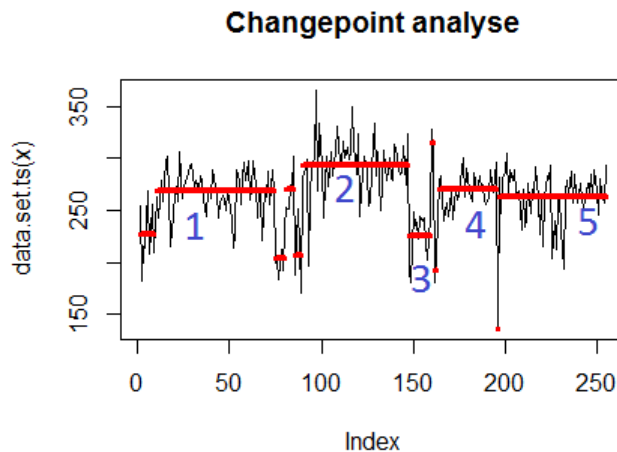
Merk op dat de perioden 1 en 2, van februari tot oktober ongeveer het wielerseizoen zijn. In het seizoen van 2014, periode 2, liggen de waarden voor de *GCP* hoger dan in het seizoen van 2013. Aan de hand van het logboek is dit verschil te verklaren. In deze periode reed de renner met een andere vermogensmeter, waarvan al vermoed werd dat deze te hoge waarde meette. Het resultaat van de changepoint analyse is hier een extra bevestiging van. Tussen de twee periodes 1 en 2 in werden er lagere waarden gehaald. Dit kan te maken hebben met vermoeidheid, of met het in vorm komen. Uit het logboek van de renner blijkt dat hij vanaf begin januari 2015 tot eind februari 2015, in periode 3 dus, last heeft gehad van ziekte en daarna oververmoeidheid richting een meerdaagse wedstrijd. De changepoint analyse haalt deze mindere periode er inderdaad uit. Voor de penalty is hier een keuze gemaakt voor 6000.

Op dezelfde data kan de taut string methode toegepast worden. Figuur 6.3 laat weer verschillende afbeeldingen zien voor de verschillende keuzes van de band. Merk het grote verschil op. De taut string methode detecteert de uitschieters naar beneden niet, maar geeft een constanter gemiddelde terug. In de detectie van prestatie van een wielrenner kan dit een voordeel zijn, omdat een slechte dag dan niet gelijk een knoop inhoudt, maar wanneer de afbeeldingen vergeleken worden, komt de changepoint overtuigender over, deze detecteert zichtbaardere verschillen. De varantie van de gemeten *GCP*'s is dusdanig groot dat de taut string



Figuur 6.1: Changepoint analyse toegepast op de *GCP*'s van 2015, van renner 4, met verschillende waarden voor de penalty.

methode hierdoor ook knopen detecteerd. Herinner figuur 5.4. Op de punten waar het koord als het ware 'het bochtje' om moet, geeft de methode een aantal knopen. Dit verklaart de waarom er niet één duidelijke knoop is, maar waarom er ook een aantal kunnen zijn. De changepoint analyse maakt ruwere keuzes dan de taut string methode.



Figuur 6.2: Changepoint analyse toegepast op alle *GCP*'s voor de wedstrijden van renner 4, penalty = 6000, perioden aangegeven.

De taut string kan ook weer toegepast worden op alle *GCP*'s van alle wedstrijden. Figuur 6.4 toont verschillende intervallen en tabel 6.2 geeft hiervan een overzicht.

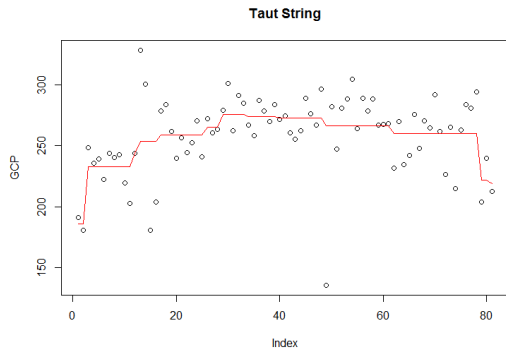
Periode	Index	Dagen	Datum
1	[13,75]	478-766	18 februari 2013-3 december 2014
2	[95,148]	838-1168	14 februari 2014-28 januari 2015
3	[149,164]	1187-1240	29 januari 2015-23 maart 2015
4	[173,256]	1256-1644	8 april 2015-30 april 2016

Tabel 6.2: De data bij de knopen in de taut string methode, toegepast op alle *GCP*'s voor de wedstrijden van renner 4, band = 0.8

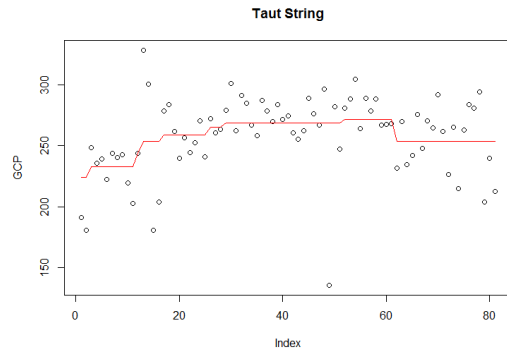
De knopen die de taut string methode terug geeft, zijn dan anders, de grote lijnen zijn wel weer hetzelfde. De periode van 2014 was deze renner beter dan de perioden ervoor en erna en is de dip in het voorjaar van 2015 zichtbaar.

De waarde van de penalty en de waarde van de band zijn rennerafhankelijk. Dit heeft te maken met het feit dat iedere renner verschillend is en dit verschil zie je terug in de waarden van de *GCP*'s. Bekijk figuur 6.5a en 6.5b. Hier zijn de plots te zien voor het jaar 2015 van renner 1 en renner 5. Renner 1 rijdt veel constantere *GCP*'s dan renner 5. Daarom zal een verandering bij renner 1 lastiger te detecteren zijn dan een verandering bij renner 5. De penalty zal bij renner 1 daarom lager moeten zijn dan bij renner 5 om eenzelfde aantal changepoints te kunnen bepalen.

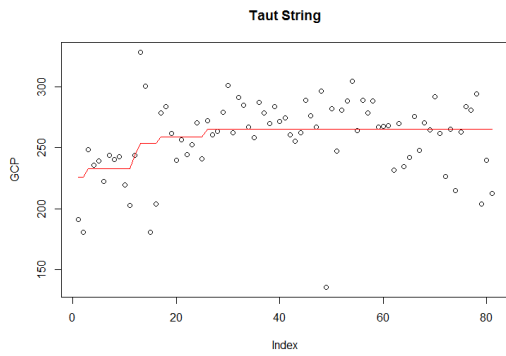
Voor iedere renner kan voor iedere gewenste periode een plot gemaakt worden om de veranderingen te detecteren. Gezien het feit dat de changepoint analyse de verschillen duidelijker aangeeft, is de aanbeveling om deze methode te gebruiken. Met de waarde van de penalty dient een beetje 'gespeeld' te worden, geheel afhankelijk van de renner, de tijdsperiode en het aantal changepoints dat gedetecteerd dient te worden. Wanneer gekeken wordt naar een korte periode, 1 tot 3 maanden, omdat er een dergelijk vermoeden is over de verandering van de prestatie van een renner, dient de penalty rond de 1000 te zijn. Op deze manier kunnen er veel changepoints gedetecteerd worden. Wanneer we echter kijken naar alle beschikbare data, dan is een penalty van rond de 6000 meer op zijn plaats, dit geeft een goede schatting van de verschillende prestatieniveaus over de verschillende seizoenen.



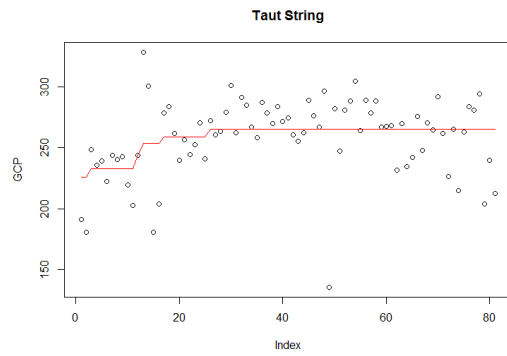
(a) band = 1



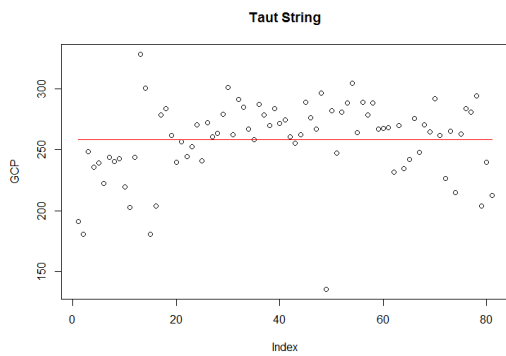
(b) band = 2



(c) band = 3

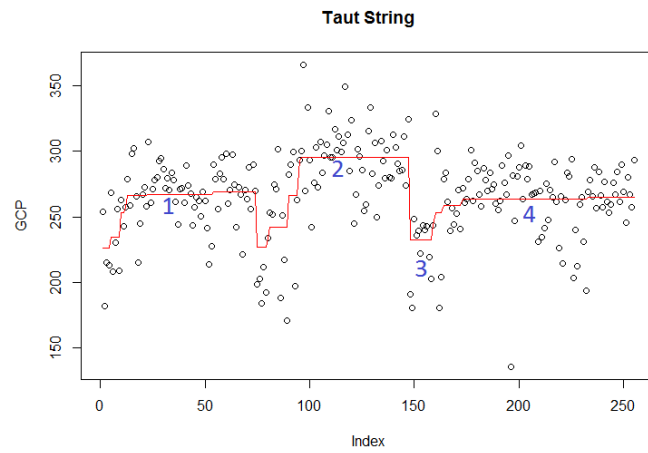


(d) band = 4

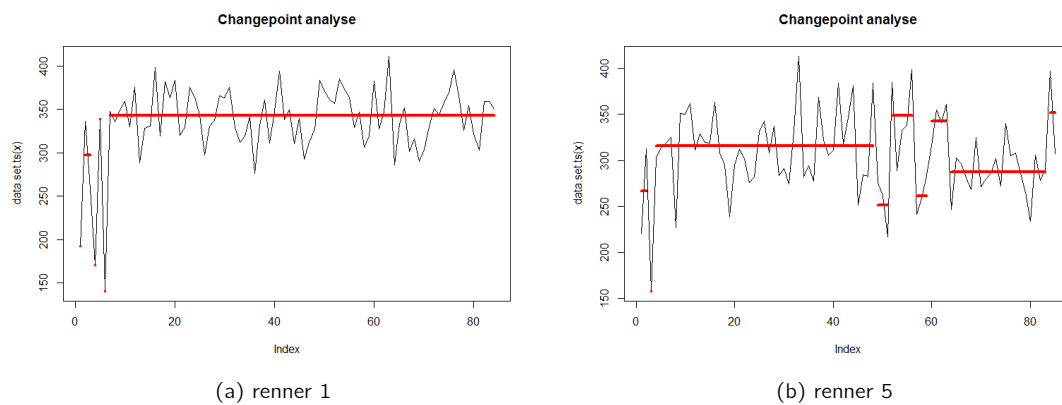


(e) band = 5

Figuur 6.3: Taut string toegepast op de *GCP*'s van 2015, van renner 4, met verschillende waarden voor de band



Figuur 6.4: Taut string toegepast op alle *GCP*'s voor de wedstrijden van renner 4, band = 0.8, perioden aangegeven.



Figuur 6.5: Changepoint analyse toegepast op de waarden van 2015, voor renner 1 en 5, penalty = 6000.





## 7 Conclusies en aanbevelingen

### 7.1 Conclusies

Het antwoord op de onderzoeksvraag, die luidde

Is het mogelijk om aan de hand van het  $W'$ balans model een maat van prestatie te bepalen en zijn er veranderingen te detecteren in deze maat van prestatie in verloop van de tijd?

luit ja. Dit antwoord is opgebouwd in de verschillende hoofdstukken. In hoofdstuk 2 werd duidelijk dat onder andere de tijd en het vermogen worden gemeten, iedere keer als de renner op de fiets zit. Dit vermogen kan beschouwd worden als een tijdsgeordende dataset  $y = (y_1, \dots, y_n)$ . Voor iedere gefietste dag is er een dergelijke dataset. De data in deze dataset komt niet uit een normale verdeling, daar heeft de dataset een te grote piek voor. Door middel van een Box-Cox transformatie kan de data enigszins symmetrisch gemaakt worden, maar voor iedere dataset is een andere maat van transformatie nodig. Voordat de data gebruikt kan worden in het modelleren, is het nodig om het een en ander af te weten van de verschillende energiesystemen in het lichaam. Het ATP + CrP systeem is een systeem dat voor korte duur, zo'n 20 seconde snel en efficiënt veel energie kan leveren. Dit proces is energieonafhankelijk en de verbruikte energie komt uit de spieren. Voor een langere inspanning, tot 2 minuten, maakt het lichaam gebruik van het anaëroob lactisch systeem. Dit systeem is ook zuurstofafhankelijk, kan minder energie leveren dan het ATP + CrP systeem, maar kan dit wel langer volhouden. Een bijproduct van dit systeem is de lactaatoefoping in het lichaam. Dit veroorzaakt een overschot aan waterstofionen in het bloed, wat leidt tot verzuring. Wegens deze reden kan dit systeem niet oneindig lang energie leveren. Voor inspanningen langer dan 2 minuten maakt het lichaam gebruik van het aëroob oxidatief systeem. Dit zuurstof afhankelijke proces kan energie blijven leveren, al is de snelheid laag.

Met deze kennis is de critical power ( $CP$ ) geïntroduceerd in hoofdstuk 4. De  $CP$  is de horizontale asymptoot in de hyperbolische relatie tussen de tijd en het maximale vermogen. Dit is dus een vermogen wat theoretisch gezien oneindig lang volgehouden moet kunnen worden. De formule die dit verband beschrijft is  $P = CP + \frac{W'}{t}$ , met  $W'$  de anaëroobe arbeidscapaciteit, wat beschouwd kan worden als de hoeveelheid energie die geleverd kan worden door het ATP + CrP systeem en het anaëroob lactisch systeem samen. Wanneer er met een vermogen gefietst wordt hoger dan de critical power, zal een deel van de energie uit dit systeem komen, wanneer er met een vermogen wordt gefietst lager dan de critical power, wordt deze energie weer aangevuld. Het verloop van dit leeglopen en aanvullen van energie kan weergegeven worden door de  $W'$ balans. Hierbij wordt aangenomen dat het opladen verloopt via een exponentieel verband. Door de waarde van  $W'$  vast te nemen en de waarde van de critical power te variëren, bereikt de renner voor een bepaalde waarde van  $CP$  dat  $W'_{\text{bal}}$  gelijk is aan 0. Deze specifieke  $CP$  heet de gefietste critical power  $GCP$  en deze geeft een maat van prestatie aan.

Het detecteren van veranderingen kan door middel van de changepoint analyse. De changepoint analyse gaat op zoek naar een verschil in de beschrijvende parameter van een bepaalde verdeling. Enkelvoudige changepoint detectie gaat door middel van de log-likelihoodratio. Meervoudige changepoint detectie, dus het detecteren van meerdere changepoints kan op verschillende manieren. Het PELT-algoritme kan de changepoints exact detecteren en kan dit met computationele complexiteit  $\mathcal{O}(n)$ , wat de methode een concurrent maakt voor de taut string methode. De taut string methode bepaalt de punten waar veranderingen optreedt door de datapunten te integreren en een buis te creëren door  $\lambda$  bij deze punten op te tellen en af te trekken. Vervolgens worden de oorspronkelijke punten als een koord beschouwd en wordt dit aangetrokken. Dit levert een stuksgewijs lineaire functie op, welke gedifferentieerd kan worden tot een stuksgewijs constante functie. Deze stuksgewijs constante functie bevat 'sprongen', de punten van verandering. Het vergelijken van de methoden levert op dat de changepoint analyse de veranderingen in de genereerde data netjes teruggeeft. De taut string methode kon dit niet precies. Ook in de toepassing op de  $GCP$ 's was dit een zichtbaar kenmerk. De changepoint analyse detecteerde veranderingen, die ook verklaard kunnen worden aan de hand van het logboek. De taut string methode detecteert wel veranderingen, maar deze is minder nauwkeurig dan de changepoint analyse.

Geconcludeerd geeft het werken met  $W'_{\text{bal}}$  een goede schatting voor de maat van prestatie en met behulp van de changepoint analyse zijn veranderingen duidelijk te detecteren.

## 7.2 Aanbevelingen

Een subdoel van dit onderzoek was om het model omtrent  $W'_{\text{bal}}$  uit te zoeken en aanbevelingen te doen voor een eventueel vervolgonderzoek. Vanuit dit verslag zie ik veel mogelijkheden voor verschillende vervolgonderzoeken en ook verscheidene toepassingen voor de  $GCP$ 's.

Vervolgonderzoek zou gedaan kunnen worden op het gebied van de vermogensdata. Hier kan meer analyse naar gedaan worden. De verdeling waaruit deze data komt kan bepaald worden en vervolgens kan er met de extreme waarden theorie gekeken worden wat dit inhoudt voor de verdeling van de waarden waarmee de critical power bepaald wordt. Evenzo zou ook naar de verdeling van de data van de  $GCP$  gekeken kunnen worden.

Om nauwkeurigere schattingen te verkrijgen voor de  $CP$  en  $W'$  kan er onderzoek worden gedaan naar uitgebreidere modellen om deze te bepalen. In hoofdstuk 4.1 is daar al even op ingegaan.

Een vervolgstap op dit onderzoek zou zijn om te kijken naar die verschillen die gedetecteerd worden door middel van de changepoint detectie. Wanneer zijn deze verschillen significant, wanneer kun je iets zeggen over een verandering in vorm en wellicht over het gebruik van verboden middelen? Dit is een zeer waardevol vervolg.

Ook is het goed om in een verder onderzoek de factor van vermoeidheid mee te nemen. Renners raken door het seizoen heen vermoeid. Welke invloed heeft dit op de prestaties? In lijn hiermee kan het goed zijn om ook de ervaringen van renners over bepaalde dagen te noteren. Als we bijvoorbeeld terugkijken naar figuur 6.2 en de negatieve piek tussen periode 4 en 5, kan het waardevol zijn om te weten hoe deze dag was voor de renner. Uit het logboek valt informatie te halen over ziekte en prestaties, maar een opinie van de renner zou een goede toevoeging zijn.

Een van de nieuwste ontwikkelingen in het wielrennen is de samenwerking tussen Velon en Infront Sports & Media. Velon is een initiatief van de 11 UCI World Tour wielrenploegen, de profploegen, die sport en fans dichter bij elkaar willen brengen. Infront Sports & Media houdt zich bezig met sportmarketing. Door deze samenwerking komt Velon onder andere met live data tijdens de wedstrijden. Deze live data bevat ook de vermogensdata van een aantal renners. Het  $W'_{\text{bal}}$  model zou dus tijdens de wedstrijd al toegepast kunnen worden om het verloop van de energie van de renner weer te geven. Deze kennis kan invloed hebben op de strategie van de ploeg. Als blijkt dat een renner nog een hoge  $W'_{\text{bal}}$  heeft, zou deze een aanval kunnen doen en als dit niet het geval is, kan een renner beter even rustig aan doen. Deze toepassing zou mijns inziens direct geïmplementeerd kunnen worden.

Mijn laatste aanbeveling past mooi bij het recente nieuwsbericht dat Team Giant-Alpecin een opleidingsploeg gaat opstarten. Renners die net beginnen met fietsen, maken veel ontwikkelingen door en zullen waarschijnlijk steeds beter gaan fietsen. Dit moet zichtbaar zijn in de waarden voor de  $GCP$ . Dit model kan dus ook een goede methode zijn om talent te herkennen.

## Referenties

- [1] I.E. Auger and C.E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of mathematical biology*, 51(1):39–54, 1989.
- [2] J. Bai and P. Perron. Estimating and testing linear models with multiple structural changes. *Econometrica*, pages 47–78, 1998.
- [3] R.E. Barlow, D.J. Bartholomew, J.M. Bremner, and H.D. Brunk. *Statistical inference under order restrictions: The theory and application of isotonic regression*. Wiley New York, 1972.
- [4] H.C. Bergstrom, T.J. Housh, J.M. Zuniga, D.A. Traylor, R.W. Lewis Jr, C.L. Camic, R.J. Schmidt, and G.O. Johnson. Differences among estimates of critical power and anaerobic work capacity derived from five mathematical models and the three-minute all-out test. *The Journal of Strength & Conditioning Research*, 28(3):592–600, 2014.
- [5] Golden Cheetah. Extended cp model, 2006. URL <https://github.com/GoldenCheetah/GoldenCheetah/blob/master/src/Metrics/ExtendedCriticalPower.cpp>. Geraadpleegd op 29-04-2016, zie ook <https://twitter.com/grauser/status/417056955451318273> © Daniel.
- [6] L. Davies and A. Kovac. *ftnonpar: Features and Strings for Nonparametric Regression*, 2012. URL <http://CRAN.R-project.org/package=ftnonpar>. R package version 0.1-88.
- [7] P.L. Davies and A. Kovac. Local extremes, runs, strings and multiresolution. *Annals of Statistics*, pages 1–48, 2001.
- [8] D.W. Hill. The critical power concept. *Sports Medicine*, 16(4):237–254, 1993.
- [9] D.V. Hinkley. Inference about the change-point in a sequence of random variables. *Biometrika*, 57(1): 1–17, 1970.
- [10] B. Jackson, J.D. Scargle, D. Barnes, S. Arabhi, A. Alt, P.R Gioumouisis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T.T. Tsai. An algorithm for optimal partitioning of data on an interval. *Signal Processing Letters, IEEE*, 12(2):105–108, 2005.
- [11] R. Killick and I.A. Eckley. changepoint: An r package for changepoint analysis. *Journal of Statistical Software*, 58(3):1–19, 2014.
- [12] R. Killick, P. Fearnhead, and I.A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.
- [13] R. Killick, K. Haynes, and I.A. Eckley. *changepoint: Methods for Changepoint Detection*, 2016. URL <http://CRAN.R-project.org/package=changepoint>. R package version 2.2.1.
- [14] A. Kovac. Local extreme values, 2006. URL <http://www.maths.bris.ac.uk/~maxak/ast.pdf>. Geraadpleegd op 31-05-2016.
- [15] E. Mammen, S. van de Geer, et al. Locally adaptive regression splines. *The Annals of Statistics*, 25 (1):387–413, 1997.
- [16] S. Mantica. Human performance modelling on wearable devices. 2015.
- [17] Mathworks. Nonlinear regression. URL <http://nl.mathworks.com/help/releases/R2016a/stats/nonlinear-regression-1.html>. Geraadpleegd op 16-06-2016.
- [18] D.S. Matteson and N.A. James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505):334–345, 2014.
- [19] H. Monod and J. Scherrer. The work capacity of a synergic muscular group. *Ergonomics*, 8(3): 329–338, 1965.

- [20] T. Moritani, A. Nagata, H.A. Devries, and M. Muro. Critical power as a measure of physical work capacity and anaerobic threshold. *Ergonomics*, 24(5):339–350, 1981.
- [21] H.R. Morton. A 3-parameter critical power model. *Ergonomics*, 39(4):611–619, 1996.
- [22] R.H. Morton. The critical power and related whole-body bioenergetic models. *European journal of applied physiology*, 96(4):339–354, 2006.
- [23] A.J. Scott and M. Knott. A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, pages 507–512, 1974.
- [24] P.F. Skiba, W. Chidnok, A. Vanhatalo, and A.M. Jones. Modeling the expenditure and reconstitution of work capacity above critical power. *Medicine and science in sports and exercise*, 44(8):1526–1532, 2012.
- [25] P.F. Skiba, D. Clarke, A. Vanhatalo, and A.M. Jones. Validation of a novel intermittent w' model for cycling using field data. *International Journal of Sports Physiology & Performance*, 9(6), 2014.
- [26] M.C. Tsai. *Revisiting the power-duration relationship and developing alternative protocols to estimate critical power parameters*. PhD thesis, University of Toronto, 2015.
- [27] D. Waterworth. W'bal optimization by a mathematician!, 2014. URL <http://markliversedge.blogspot.nl/2014/10/wbal-optimisation-by-mathematician.html>. Geraadpleegd op 25-04-2016.
- [28] J.H. Wilmore and D.L. Costill. *Physiology of sports and exercise*. Human Kinetics, Champaign, second edition, 1999.
- [29] Y.C. Yao. Estimation of a noisy discrete-time step function: Bayes and empirical bayes approaches. *The Annals of Statistics*, pages 1434–1447, 1984.

## A Bijlagen

### A.1 Algemene matlab code

De volgende code wordt gebruikt om de nummers van de datafiles om te zetten in een datum of de datum in het nummer van de file

```
numtodate.m

function [y] = numtodate(x)
% NUMTODATE schrijft de nummers om naar de juiste datum
% Dag 374 komt overeen met 6-11-2012, de datenum uit Matlab hierbij is
% 735179, vanaf 1 januari 2014 moet er nog een extra dag bij opgeteld worden.

if x < 795
    y = datestr(734805 + x)
else
    y = datestr(734806 + x)
end
end
```

```
datetinum.m

function [x] = datetinum(dag, maand, jaar)
% DATETINUM schrijft de datum om naar het bijbehorende nummer van de
% files. Als we verder zijn dan 1 januari 2014, moet er 1 afgehaald worden.

DateVector=[jaar,maand,dag,0,0,0];
x=datenum(DateVector)-734805;
if x>794
    x=x-1;
end
end
```

De volgende code wordt gebruikt om de nummers van de datefiles te bepalen waarop een renner heeft fietst.

```
VindDagen.m

function [dagen] = VindDagen(i)
%VINDDAGEN creert een vector die de nummers van alle dagen bevat waarvan er
%data beschikbaar is. Deze data bevindt zich in de file renneri.m.

renner = {'renner1','renner2','renner3','renner4','renner5'}
filename = sprintf('%s.mat',renner{i});
load(filename)
if i == 1
    dagen = renner1.alles(:,46);
elseif i == 2
    dagen = renner2.alles(:,46);
elseif i == 3
    dagen = renner3.alles(:,46);
elseif i ==4
    dagen = renner4.alles(:,46);
elseif i == 5
    dagen = renner5.alles(:,46);
```

```
end
end
```

De volgende code wordt gebruikt om de nummers van de datefiles te bepalen waarop een renner een wedstrijd heeft gereden.

```
VindWedstrijd.m

function [wedstrijd] = VindWedstrijd(i)
%VINDWEDSTRIJD creert een vector die de nummers van alle dagen bevat waarop de
%renner een wedstrijd gereden heeft. Deze data bevindt zich in de file renneri.m.

renner = {'renner1','renner2','renner3','renner4','renner5'}
filename = sprintf('%s.mat',renner{i});
load(filename)
if i == 1
    wedstrijd = renner1.alles(:,47);
    dagen = renner1.alles(:,46);
elseif i == 2
    wedstrijd = renner2.alles(:,47);
    dagen = renner2.alles(:,46);
elseif i == 3
    wedstrijd = renner3.alles(:,47);
    dagen = renner3.alles(:,46);
elseif i == 4
    wedstrijd = renner4.alles(:,47);
    dagen = renner4.alles(:,46);
elseif i == 5
    wedstrijd = renner5.alles(:,47);
    dagen = renner5.alles(:,46);
end

n = length(wedstrijd);
for j = 1:n
    if wedstrijd(j) == 3
        wedstrijd(j) = 0;
    end
end

wedstrijd = wedstrijd > 0;
wedstrijd = wedstrijd.*dagen;

end
```

De volgende code bepaalt de samplefrequentie van een datafile

```
function [sampletime] = sf(filename)
% SF bepaalt de samplefrequentie van een datafile

data=load(filename);
variables=fields(data);
data1=data.(variables{1});
sampletime=data1(2,1)-data1(1,1);

end
```

## A.2 Matlab code voor hoofdstuk 2

De volgende code is gebruikt in hoofdstuk 2.

```
Data_Analyse.m
% Het doorlopen van dit script neemt alle stappen die gebruikt zijn in hoofdstuk 2.

clear all
close all
clc

%% Histogram
load('renner1590.mat')
power = data(:,4);
hist(power,40);
xlabel('Vermogen_(Watt)')
ylabel('Frequentie')
title('Histogram_voor_het_vermogen_van_renner_1_op_dag_590')

power(power<20)=[]; % Nu zonder de waarden onder 20 Watt
hist(power,40);
xlabel('Vermogen_(Watt)')
ylabel('Frequentie')
title('Histogram_voor_het_vermogen_van_renner_1_op_dag_590_zonder_de_waarden_onder_20_Watt')

%% Bepalen van de Box-Coxtransformatie, door middel van de Measure of Skewness (MS)
w = VindP(power);
figure(1)
histfit(p(power,w),40)
xlabel('Vermogen_(Watt)')
ylabel('Frequentie')
title(['Histogram_na_de_Box-Cox_transformatie_met_p=', num2str(w)])
```

De volgende drie functies zijn gebruikt in de bovenstaande code

```
VindP.m

function [lambda] = VindP(power)
% VINDP bepaalt de optimale p waarde voor een boxcox transformatie

power(power < 20) = [];
lambda = 1;
a = MS(power);
r = zeros(1,2);
k = 0;
if a > 1.05
    r(1) = -1;
    lambda = lambda - 0.5;
elseif a < 0.95
    r(1) = 1;
    lambda = lambda + 0.5;
end

a = MS(p(power,lambda));
```

```

i=1;
while ((a < 0.95)|(a > 1.05))

    i = i + 1;
    if a > 1.05
        r(i) = -1;
    elseif a < 0.95
        r(i) = 1;
    end

    if (k<1)&&(r(i-1) ~= r(i))
        k = 1;
    else
        k;
    end

    if k < 1
        if a > 1.05
            lambda = lambda - 0.5;
            if lambda == 0
                a = MS(log(power));
            else
                a = MS(p(power,lambda));
            end
        elseif a < 0.95
            lambda = lambda + 0.5;
            a = MS(p(power,lambda));
        end

        else
            lambda = lambda + r(i)*0.5/(2^k);
            a = MS(p(power,lambda));
            k = k + 1;
        end
    end
end
end

```

MS.m

```

function [Ms] = MS(power)
%%MS bepaalt de measure of skewness voor een dataset

n = length(power);
power = sort(power);
dM = floor((n+1)/2);
M = (power(dM));
dHI = floor((dM+1)/2);
HI = (power(dHI));
dHu = n-dHI+1;
Hu = (power(dHu));
Ms = (Hu-M)/(M-HI);
end

```



p.m

```
function [x] = p(x,y)
%P transformeert de data

if y==0
    x=log(x);
else
    x=(x.^y-1)./y;
end
end
```

### A.3 Matlab code voor hoofdstuk 4

De volgende code is gebruikt voor het bepalen van de  $CP$  en  $W'$  in Hoofdstuk 4.1.

```
Critical_Power.m

% Het doorlopen van dit script produceert de informatie die nodig is voor het bepalen van de
% critical power.

clear all
close all
clc

k = 8; % Kies door k te variëren een renner, k = (1, 2, 3, 4, 5)
renner = {'renner1','renner2','renner3','renner4','renner5'};

% Vector met daarin de dagen maken
dagen = VindDagen(k);
dagen(dagen == 0) = []; % Kies voor renner 4 dagen(dagen < 900) = [];

% Tijdstappen kiezen (in seconden)
c1 = [1,5,10,30,60,120,300,600,900,1200,1500,1800,2400,3000,3600];
c2 = [60,120,300,600,900,1200,1500,1800,2400,3000,3600]; % Zonder de sprintwaarden
c3 = [60,120,300,600,900,1200,1500,1800]; % Zonder de sprintwaarden en tot en met 20 minuten

% d vullen
m = length(c1);
n = length(dagen);
d = zeros(n,m); % Deze matrix bevat straks alle maxima van alle dagen

for i = 1:n
    dag = dagen(i);
    filename = sprintf('%s%g.mat',renner{k},dag);
    data = load(filename);
    variables = fields(data);
    data1 = data.(variables{1});

    for j = 1:m
        d(i,j) = piekpower(data1,c1(j),dag,k); % c1(j) is de tijd waarover het
        % voortschrijdend gemiddelde bepaald moet worden.
    end
end
```

```

end

% dmax bepalen en locatiedmax vullen
dmax = zeros(1,m); % In deze vector komen alle maximale waarden te staan.
locatiedmax = zeros(1,m); % Deze vector bevat het dagnummer van de maxima.

for i = 1:m
    [dmax(i),locatiedmax(i)] = max(d(:,i));
    locatiedmax(i) = dagen(locatiedmax(i));
end

dmax2 = dmax(1,5:12); % Zonder sprintwaarden en zonder alles verder dan 20 min.
locatiedmax2 = locatiedmax(1,5:12);

%% Dit subscript geeft code voor het maken van plots

figure;
plot(c1,dmax,'o');
title('De_tijd-vermogen_plot');
xlabel('Tijd_(sec)');
ylabel('Vermogen_(Watt)');
axis([-5 4000 0 1400]);

figure(2);
plot(c1,c1.*dmax./1000,'o');
title('De_tijd-arbeid_plot');
xlabel('Tijd_(sec)');
ylabel('Arbeid_(kJ)');
axis([-5 4000 0 1400]);

%% Nu verschillende modellen proberen.

%% Nonlineaire 2 parameter model  $P = CP + W'/t$ 

xdata = c3;
ydata = dmax2;
fun = @(x,xdata)x(2)./xdata + x(1);
x0 = [400,30000];
opts = optimset('lsqcurvefit');
optimset(opts,'Display','off');
[x,rss,residual,exitflag,output,lambda,J] = lsqcurvefit(fun,x0,xdata,ydata,[],[],opts);
y = mean(ydata);
tss = som((ydata-y).^2,1,8);
rkw = 1 - rss/tss % Bepaalt r kwadraat
conf = nlparci(x,residual,'jacobian',J) % Bepaalt het 95% betrouwbaarheidsinterval
disp(['Voor_het_nlineaire_model_geldt_CP=', num2str(x(1)), ' en W=', num2str(x(2))]);
figure(3)
plot(xdata,ydata,'ko',xdata,fun(x,xdata),'b-');
title('P=W/t+CP');
xlabel('Tijd_(sec)');
ylabel('Vermogen_(Watt)');

%% Lineaire TW model  $TW = W' + CP*t$ 

```

```

xdata = c3;
ydata = dmax2.*c3;
fun = @(x,xdata)x(2) + xdata.*x(1);
x0 = [400,30000];
opts = optimset('lsqcurvefit');
optimset(opts,'Display','off');
[x,rss,residual,exitflag,output,lambda,jacobian] = lsqcurvefit(fun,x0,xdata,ydata,[],[],opts);
y = mean(ydata);
tss = som((ydata-y).^2,1,8);
rkw = 1 - rss/tss % Bepaalt r kwadraat
conf = nlparci(x,residual,'jacobian',jacobian) % Bepaalt het 95% betrouwbaarheidsinterval
disp(['Voor_het_lineaire_TW_model_geldt_CP=', num2str(x(1)), ' en W=', num2str(x(2))]);
figure(4)
plot(xdata,ydata./1000,'ko',xdata,fun(x,xdata)./1000,'b-');
title('P=W/t+CP');
xlabel('Tijd(sec)');
ylabel('Totale_arbeid_(kiloJoule)');

%% Nonlineaire 3 parameter model P = CP + W/(t+tau)

xdata = c3;
ydata = dmax2;
fun = @(x,xdata)x(2)./(xdata + x(3)) + x(1);
x0 = [400,30000,1];
opts = optimset('lsqcurvefit');
optimset(opts,'Display','off');
[x,rss,residual,exitflag,output,lambda,jacobian] = lsqcurvefit(fun,x0,xdata,ydata,[],[],opts);
y = mean(ydata);
tss = som((ydata-y).^2,1,8);
rkw = 1 - rss/tss % Bepaalt r kwadraat
conf = nlparci(x,residual,'jacobian',J) % Bepaalt het 95% betrouwbaarheidsinterval
disp(['Voor_het_nonlineaire_3_parameter_model_geldt_CP=', num2str(x(1)), ' W=',
num2str(x(2)), ' en tau=', num2str(x(3))]);
figure(5)
plot(xdata,ydata,'ko',xdata,fun(x,xdata),'b-');
title('P=W/(t+tau)+CP');
xlabel('Tijd(sec)');
ylabel('Vermogen_(Watt)');

```

De volgende functies zijn gebruikt in de bovenstaande code

```

piekpower.m

function [data2] = piekpower (data,t,dag,renner)
% PIEKPOWER berekent het maximale gemiddelde wattage.
% t is de tijd in seconden waarvan we het gemiddelde wattage willen weten
% Deze functie is afgeleid van de functie piekpower1.m, geschreven door Teun van Erp
power = data(:,4);
a = 1;
if (renner == 1) && (dag < 777) % Voor renner 1 lager dan dag 777 is de samplefrequentie 0.5
% in plaats van 1.
    a = 0.5;
elseif (renner == 5) && (dag < 375) % Voor renner 5 lager dan dag 375 is het ook 0.5.

```

```

    a = 0.5;
end

if a == 0.5 % Als de samplefrequentie 0.5 is, moet t maal 2
    t = t*2;
end

for i = t;

    if i < length (power);
        y = voortgem(power, i);
        data2 = max (y);
    else
        data2 = 0;
    end
end
end

```

```

voortgem.m

function [y] = voortgem(x,m)
%VOORTGEM berekent het voortschrijdende gemiddelde van lengte m voor de tijdreeks x

if m == 1 % Als we dit per 1 punt willen, hoeft de functie niets te doen
    y = x;
    return
end

n = length(x);
y = zeros(n,1);

if m > n
    y = 0;
    return
end

for i = 1:m-1
    y(i) = NaN;
end

for i = m:n
    a = i - m + 1;
    b = i;
    y(i) = 1/m.*som(x,a,b);
end

end

```

```
som.m
```

```
function [s] = som(x,a,b)
%SOM bepaalt de som van de elementen x(a) tot en met x(b)

s = 0;
for i = a : b
    s = s + x(i);
end

end
```

De volgende code is gebruikt voor het bepalen van de W'bal en GCP's in Hoofdstuk 4.2.

```
W_balans.m
```

```
% Door het doorlopen van dit script, kunen alle berekeningen omtrent W'bal gedaan worden.
```

```
%% Dit subscript bepaalt de W'bal voor renner k op dag d
```

```
k = 4 % Vul hier het nummer van de renner in
d = 670 % Vul hier het nummer van de dag in
renner = {'renner1','renner2','renner3','renner4','renner5'};
load(sprintf('%s%g.mat',renner{k},d));
CP = [415,371,387,327,375]
W = [29211,27314,35410,28018,36280];
a = Wbal(data,CP(k),W(k),d,k);
```

```
%% Dit subscript vergelijkt de Wbal van de verschillende renners voor een bepaalde dag d
```

```
renner = {'renner1','renner2','renner3','renner4','renner5'};
W = [29211,27314,35410,28018,36280];
CP = [415,371,387,327,375];
d = 1240; % Vul hier de desbetreffende dag in
n = length(renner);
plotStyle = ['r','c','b','m','g'];
```

```
for i = 4
```

```
    filename = sprintf('%s%g.mat',renner{i},dag);
    data = load(filename);
    variables = fields(data);
    data1 = data.(variables{1});
    Wb = Wbal(data1,CP(i),W(i),d,i);
    plot(data1(:,1)*60*24,Wb,plotStyle(i))
    legendInfo{i} = ['renner' ,num2str(i)];
    hold on
```

```
end
```

```
xlabel('Tijd_in_minuten');
ylabel('Wbal_in_Joule');
title(['Het_verloop_van_Wbal_over_dag' ,num2str(d), 'voor_alle_renners']);
axis([0 300 0 60000])
legend(legendInfo)
```

```

%% Dit subscript bepaalt de GCP voor een bepaalde periode voor renner k voor alle wedstrijden

k = 3 % Vul hier het nummer van de renner in
a = numtodate(1,1,2015) % Vul hier de begindatum in
b = numtodate(31,12,2015) % Vul hier de einddatum in
dagen = VindWedstrijd(k);
dagen(dagen<a) = [];
dagen(dagen>b) = [];

renner = {'renner1','renner2','renner3','renner4','renner5'};

W = [29211,27314,35410,28018,36280]; % Dit zijn de in hoofdstuk 4.1 bepaalde waarden voor W'

n = length(dagen);
v = zeros(n,2);
for i = 1:n
    v(i,1) = i;
    if dagen(i) > 0
        filename = sprintf('%s%g.mat',renner{k},dagen(i));
        data = load(filename);
        variables = fields(data);
        data1 = data.(variables{1});
        v(i,2) = VindCP(data1,W(k),dagen(i),k);
    else
        v(i,2) = NaN;
    end
end
end

plot(v(a:b,1),v(a:b,2),'.')
title(['GCPs van renner ', num2str(k), ' voor wedstrijd dagen in de periode ', num2str(a) ,
' tot ', num2str(b)])
xlabel('dag')
ylabel('GCP')
axis([a b 100 450])

%% Het onderstaande subscript slaat de waarden van de GCP's op, zodat we die straks in R
kunnen inladen
save('naam.mat','v')

%% Dit subscript bepaalt de GCP's voor een koers
renners = {'renner1','renner2','renner3','renner4','renner5'};
W = [29211,27314,35410,28018,36280];
m = length(renners);
GCPs = zeros(1,m);
begindatum = datetonum(23,3,2015); % Vul de juiste datum hier in.
einddatum = datetonum(29,3,2015); % Vul de juiste datum hier in.
dagen = [begindatum:einddatum];
n = length(dagen);

for i = 1:m
    renner = renners(i);

```

```

for j = 1:n
    dag = dagen(1,j);
    filenaam = sprintf('%s%g.mat', renners{i}, dag);
    try
        load (filenaam);
        GCPs(j,i) = VindCP(data,W(1,i),dag,i);
    catch
        GCPs(j,i) = NaN;
    end
end
end
GCPs;
figure(1)
plot(dagen,GCPs(:,1),'b',dagen,GCPs(:,2),'r',dagen,GCPs(:,3),'g',dagen,GCPs(:,4),'m',dagen,
GCPs(:,5),'c')
legend('renner_1','renner_2','renner_3','renner_4','renner_5');
title('De_GCP_waarden_voor_een_zevendaaigse_koers_in_2015');
xlabel('dag');
ylabel('GCP');
axis([dagen(1,1)-0.1 dagen(1,n)+0.1 150 550])

```

In het bovenstaande scripts wordt gebruik gemaakt van de onderstaande functies

VindCP.m

```

function [fixedCP] = VindCP(data,W,dag,renner)
% VINDCP bepaalt de GCP waar de renner die dag op fietste, uitgaande
% van dat Wbal precies 0 wordt, ofwel komt in het interval [-a,a]

CP = 300; % Dit is de waarde waarmee we starten.
a = 1;
b = Wbal(data,CP,W,dag,renner);
c = Wbaldal(b,W);
i = 1; % i loop alle stappen door
r = zeros(1,2); % r telt de afgelegde weg over de boom mee
k = 0; % k geeft aan wanneer we naar het rijtje overstappen
d = zeros(1,2);
% De eerste doe ik handmatig
if c > a
    r(1) = -1;
    CP = CP - 50;
    b = Wbal(data,CP,W,dag,renner);
    c = Wbaldal(b,W);
elseif c < -a
    r(1) = 1;
    CP = CP + 50;
    b = Wbal(data,CP,W,dag,renner);
    c = Wbaldal(b,W);
end

while ((c<-a)|(c>a))
    i = i + 1;
    if c > a
        r(i) = -1;
    end

```

```

elseif c < -a
    r(i) = 1;
end

if (k<1)&&(r(i-1) ~ = r(i))
    k = 1;
end

if k<1
    if c > a
        CP = CP - 50;
        b = Wbal(data,CP,W,dag,renner);
        c = Wbaldal(b,W);
    elseif c < - a
        CP = CP + 50;
        b = Wbal(data,CP,W,dag,renner);
        c = Wbaldal(b,W);
    end
    else
        CP = CP + r(i)*50/(2^k);
        b = Wbal(data,CP,W,dag,renner);
        c = Wbaldal(b,W);
        k = k+1;
    end

    d(i) = CP;
end
r;
fixedCP = CP;
end

```

Wbal.m

```

function [Wb] = Wbal(data,CP,W,dag,renner)
% WBAL bepaalt de funtie W'bal voor een bepaalde dag

power = data(:,4);
t = tau_w(power,CP);
Sr = 1;
if (renner == 1) && (dag < 777) % Voor renner 1 lager dan dag 777 is de samplefrequentie 0.5
    % in plaats van 1.
    Sr=0.5;
elseif (renner == 5) && (dag < 375) % Voor renner 5 lager dan dag 375 is het ook 0.5.
    Sr=0.5;
end

n = length(power);
P = zeros(1,n);
Wexp = zeros(1,n);
Sn = 0;
ln = 0;
Wb = zeros(1,n);

```



```

for i = 1:n
    if power(i) < CP
        Wexp = 0;
    else Wexp = power(i)-CP;
end

Sn = Sn + Wexp*exp(i*Sr/t);
ln = exp(-i*Sr/t)*Sn;
Wb(i) = W-ln;
end

plot(data(:,1)*60*24,Wb,'r')
xlabel('Tijd(min)')
ylabel('Wbal(Joule)')
title('Het verloop van Wbalans')
axis([0 275 0 36000])

figure(2)
[ax,h1,h2] = plotyy(data(:,1)*60*24,data(:,4),data(:,1)*60*24,Wb);
xlabel('Tijd(min)')
linkaxes(ax, 'x')
axis(ax(1), [0 250 0 1200])
axis(ax(2), [0 250 0 36000])
title('De tijdsvermogenplot en de grafiek van Wbalans')
ylabel(ax(1), 'Vermogen(Watt)')
ylabel(ax(2), 'Wbal(Joule)')

end

```

```

tau_w.m

function [ tau_w ] = tau_w(power,CP)
% TAU_W bepaalt tau voor een bepaalde dag. De input is een vector
% met de Power en een double, CP, de output een double, tau_w.

n = length(power);
m = 0; % Dit wordt de optelsom van alle Wattages onder CP
l = 0; % En deze telt hoeveel het er zijn

for i = 1:n
    if power(i) < CP;
        m = m + power(i);
        n = n + 1;
    end
end
Wrecovery = m/n;
tau_w = 546*exp(-0.01*(CP-Wrecovery))+316;

end

```

```

Wbaldal.m

function [Wdal,k,m] = Wbaldal(a,W)

```

```

% WBALDAL berekent waar Wbal de laagste waarde aanneemt en wanneer
% deze kleiner dan 0 is.

% De input is de vector Wbal en de waarde W'
% De output is het getal Wdal, de laagste waarde in Wbal, k, de plaats waar
% dit gebeurt en de vector m, die alle plaatsen bevat waar Wbal onder 0
% zit.

n = length(a);
Wdal = W;
k = 0;
m = zeros(1,n);
for i = 1:n
    if a(i) < Wdal
        Wdal = a(i);
        k = i;
        if Wdal < 0
            m(i) = i;
        end
    end
end
end
k;
m(m==0) = [];
Wdal;
end

```

#### A.4 R code voor hoofdstuk 5

De volgende code is gebruikt voor het simuleren van de data en het vergelijken van de methoden in hoofdstuk 5.3.

```

Vergelijk.r

# Benodigde packages laden
library(changepoint)
library(ftnonpar)

# Vector y creëren
y <- c(rnorm(100,500,50), rnorm(100,350,50), rnorm(100,150,50),
rnorm(100,400,50), rnorm(10,250,50)) # In totaal 500 observaties.
plot(y,main='De_gesimuleerde_data')

# Changepoint Analyse
y.pelt <- cpt.mean(y,penalty='Manual', pen.value=27000,
method='PELT', class=TRUE, param.estimates=TRUE)
plot(y.pelt,typ='l', xlab='Index', main='Changepoint_analyse',cpt.width=4)
cpts(y.pelt) # Geeft de locaties van de changepoints
coef(y.pelt) # Geeft de waarde van de changepoints

# code voor de plot om de penalty uit te zetten tegen de changepoints
k <- rep(0,100001)
a <- seq(from=0, to=10000000, by=100)
for (i in a){

```

```

v.pelt <- cpt.mean(v,penalty='Manual',pen.value=i,,method="PELT",class=TRUE,
param.estimates=TRUE)
k[i/100+1] <- length(cpts(v.pelt))
}

plot(log(a),k,ylim=c(0,600),xlim=c(0,17),xlab='log(penalty)',ylab='Aantal_changepoints',
main='De_log_van_penalty_uitgezet_tegen_het_aantal_changepoints')
points(0,k[1])
plot(log(a),k,ylim=c(0,20),xlim=c(8,17),xlab='log(penalty)',ylab='Aantal_changepoints',
main='De_log_van_penalty_uitgezet_tegen_het_aantal_changepoints')

# Taut String methode
tmp <- pmreg(y,band = 9)
plot(y,col='grey',main='Taut_String')
lines(tmp$y,col='red')

a <- tmp$knotsind
n = length(a)
ygem = rep(a,0)
for (i in 1:n-1){
  ygem[i] = y[a[i]]
}
tmp$knotsind # Geeft de locaties van de knopen
ygem # Geeft de waarden van de gemiddelden

# code voor de plot om de band uit te zetten tegen het aantal knopen
l <- rep(0,81)
b <- seq(from=0.1,to = 40.1, by=0.5)
for (i in b){
  tmp <- pmreg(v,band=i)
  l[2*i+0.9] <- length(tmp$knotsind)-2
}
plot(b,l,xlim=c(0,43),ylim=c(0,150),xlab='Band',ylab='Aantal_knopen',
main='De_band_uitgezet_tegen_het_aantal_knopen')
plot(b,l,xlim=c(0,43),ylim=c(0,15),xlab='Band',ylab='Aantal_knopen',
main='De_band_uitgezet_tegen_het_aantal_knopen')

```

## A.5 R code voor hoofdstuk 6

De volgende code is gebruikt om veranderingen te detecteren in de *GCP* waarden die gecreëerd zijn in hoofdstuk 4

```

Toepassing.r

library(changepoint)
library(ftnnonpar)
library(R.matlab)

GCP <- readMat('naam.mat') # deze file is geconstrueerd in W_balans.m
GCP <- GCP[[1]]

CP <- GCP[1159:1523,] # Varieer hier het getal van de renner en het interval waarover je
# de changepoints wilt bepalen.

```

```

cp <- na.omit(CP)
cripo <- cp[,2]

# Changepoint
ts.plot(cripo,xlab="Index")
cp.pelt<-cpt.mean(cripo,penalty='Manual',pen.value='6000',method="PELT") # Varieer de
# penalty
plot(cp.pelt,typ='l',xlab='Index',main='Changepoint_analyse',cpt.width=4)
l <- cpts(cp.pelt)
n <- length(l)
dag <- rep(0,n)
for (i in 1:n){
  dag[i] <- cp[[i],1]
}
dag # geeft de nummers van de dagen waarop het changepoint plaatsvond

# Taut String
tmp <- pmreg(cripo,band = 0.8) # Varieer de band
plot(cripo,ylab=('GCP'),main=('Taut_String'))
lines(tmp$y,col='red')
a <- tmp$knotsind
n <- length(a)-1
dag <- rep(0,n)
for (i in 1:n){
  dag[i] <- cp[a[i],1]
}
dag # geeft de nummers van de dagen waarop de verandering plaatsvond

```