

BSc thesis Agrotechnologie (FTE-80812)

---

# Ontwikkeling algoritme voor de verwerking van brandstofmetingen gekoppeld aan RTK-GPS gegevens

D.W.J. Jongmans

Augustus 2011



WAGENINGEN UNIVERSITY  
WAGENINGEN **UR**



PLANT RESEARCH INTERNATIONAL  
WAGENINGEN **UR**

Afbeelding titelpagina: AIC FLOW meter, brandstofverbruiksweergave Fendt Vario Terminal, Labview logo.

# Ontwikkeling algoritme voor de verwerking van brandstofmetingen gekoppeld aan RTK-GPS gegevens

Naam vak : Bachelorafsluiting Agrotechnologie  
Vakcode : FTE-80812  
Studiebelasting : 12 credits  
Datum : Augustus-2011

Student : D.W.J. (Dré) Jongmans  
Registratienummer : 86-12-16-409-040  
Studieprogramma : BSc Agrotechnologie (BAT)

Begeleider(s) : Dr.Ir. J.W. Hofstee, Dr.Ir. A.T. Nieuwenhuizen  
Examinator : Prof.Dr.Ir. E.J. van Henten  
Leerstoelgroep : Leerstoelgroep Agrarische Bedrijfstechnologie  
Wageningen University  
Bornse Weilanden 9  
6708 WG Wageningen



# Voorwoord

Dit verslag is het resultaat van mijn bachelorafsluiting voor de studie Agrotechnologie. Het onderwerp van deze bachelorafsluiting kwam voort uit een lopend onderzoek naar brandstofverbruik uitgevoerd door dr.ir. Ard Nieuwenhuizen, werkzaam voor Plant Research International.

Ik wil dr.ir. Ard Nieuwenhuizen en dr.ir. JanWillem Hofstee bedanken voor de adviezen en begeleiding tijdens mijn bachelorafsluiting.

Tijdens de studie Agrotechnologie worden verschillende disciplines in de landbouwtechniek behandeld. Naast het uitvoeren van een zelfstandig onderzoek wilde ik met deze bachelorafsluiting graag een onderwerp verkennen dat ik nog niet zo goed kende. Zodoende trok dit onderwerp mijn aandacht tussen de andere mogelijkheden. Het ontwerpen van een algoritme en de systematische opzet hiervan leken mij interessante uitdagingen.

Tijdens de studie Agrotechnologie worden verschillende disciplines in de landbouwtechniek behandeld. Met deze bachelorafsluiting wilde ik ook graag nieuwe vakgebieden verkennen. Zodoende sprak dit onderwerp mij aan. Het ontwerpen en systematische opzetten van een algoritme leek mij een interessante uitdaging en was iets dat ik nog niet vaak gedaan had.

Dré Jongmans

Wageningen, Augustus 2011

## Samenvatting

In het kader van een onderzoek naar brandstofverbruik door een trekker tijdens grondbewerkingen zijn brandstofmetingen uitgevoerd. Doel van het onderzoek is het plaatsspecifiek brandstofverbruik van de trekker tijdens de hoofdbewerking en op de kopakker vaststellen. Tot op heden wordt elk bestand handmatig geanalyseerd. Om dit te vereenvoudigen is in deze bachelorafsluiting onderzocht wat de mogelijkheden zijn voor de ontwikkeling van een algoritme dat de data controleert op meetfouten en indeelt op kopakker of hoofdbewerking. Met deze indeling kan het verbruik in l/h en l/ha op de kopakker en tijdens de hoofdbewerking worden berekend. Ook moet het aantal werkgangen op het perceel vastgesteld worden.

Uitgangspunt is een bestaand algoritme, uitgevoerd in het programma Labview. Drie methoden zijn getest om de hoofdbewerking en de kopakker van elkaar te onderscheiden. De eerste methode is gebaseerd op één parameter: de richtingsverandering van de trekker ( $\Delta\theta$ ). Hierbij is een punt naar kopakker geschreven wanneer  $\Delta\theta$  een grenswaarde overschrijdt. De resulterende kopakkerpunten zijn circa 100% correct, er bevinden zich hier geen hoofdbewerkingspunten tussen. Binnen de hoofdbewerking bleven echter veel kopakkerpunten over. Vervolgens is gekeken of selecteren op snelheid beter was dan richtingsverandering. Dit bleek niet het geval.

De tweede methode is de methode van Kmeans. Dit is een statistische methode waarbij punten ingedeeld worden naar twee clusters (hoofdbewerking of kopakker). De indeling is gedaan op basis van drie parameters: gemiddelde richtingsverandering, snelheid en brandstofverbruik. Bij de berekening van de gemiddelden van deze parameters voor de kopakker bleek dat deze methode niet bruikbaar was. De metingen op de kopakker vertoonden namelijk een te klein verschil met de gemiddelden van de hoofdbewerking. Dit in combinatie met een grote standaardafwijking, resulteerde in te veel foutief ingedeelde punten.

De derde methode berust op een combinatie van twee parameters,  $\Delta\theta$  en de snelheid  $V$ . Dit leverde voor elk van de testpercelen een goede selectie van kopakkerpunten en is dus ook de beste methode om de data in te delen naar kopakker en hoofdbewerking.

Om het aantal werkgangen vast te stellen is gekeken naar de richtingsverandering. Deze laat hoge waarden zien tijdens keermomenten wanneer hij geplot wordt in de tijd. Naast de pieken op de keermomenten bevatte de richtingsverandering echter teveel overige pieken om een goede selectie mogelijk te maken. Een betere mogelijkheid lijkt de kopakker of de hoofdbewerking, omvatten met een polygoon en vervolgens te tellen hoe vaak de deze polygoon gepasseerd wordt. Dit is steeds een keermoment.

Het opstellen van een robuuste methode om de kopakker en de hoofdbewerking te scheiden is erg lastig gebleken. Een robuuste methode om de kopakker en de hoofdbewerking te scheiden opstellen is zeer lastig gebleken. Hierin is dan ook veel tijd gaan zitten. Het toevoegen van een extra parameter aan de metingen welke aangeeft of het werktuig in de grond zit (hoofdbewerking) of niet (kopakker) lijkt dan ook een zeer zinvolle toevoeging aan de meetopstelling. Dit

maakt het benodigde algoritme een stuk eenvoudiger en leidt tot een inkorting van de rekentijd welke voor de derde methode behoorlijk lang werd. Het verwijderen van foutieve metingen op basis van tijd en vergelijking gemiddelden van de parameters  $\Delta\theta$ , snelheid(V) en brandstofverbruik (B) werkt naar behoren. Hiermee zijn de gewenste kengetallen dan ook verkregen.

# Inhoudsopgave

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| Voorwoord .....                                                                     | ii  |
| Samenvatting.....                                                                   | iii |
| 1 Inleiding .....                                                                   | 1   |
| 2 Literatuur .....                                                                  | 2   |
| 2.1 Huidige systemen voor de verwerking van brandstofmetingen .....                 | 2   |
| 3 Materialen en Methoden .....                                                      | 5   |
| 3.1 Technische gegevens meetapparatuur .....                                        | 5   |
| 3.1.1 Brandstofmeter .....                                                          | 5   |
| 3.1.2 GPS systeem .....                                                             | 5   |
| 3.2 Lijst met eisen .....                                                           | 5   |
| 3.3 Bestanden .....                                                                 | 6   |
| 3.4 Overzicht percelen .....                                                        | 6   |
| 3.5 Overzicht algoritme .....                                                       | 7   |
| 3.6 Processen in het ontwikkelde algoritme .....                                    | 9   |
| 3.6.1 Analyse bestaande Labview Algoritme.....                                      | 9   |
| 3.6.2 Opstellen polygoon K .....                                                    | 10  |
| 3.6.3 Verwijderen te lage V binnen K .....                                          | 15  |
| 3.6.4 Berekenen Delta K, Delta T en $\Delta^2 T$ .....                              | 15  |
| 3.6.5 Verwijderen $t < 1.5$ sec .....                                               | 15  |
| 3.6.6 Methode A: Berekenen kopakkerpunten op basis $\Delta\theta$ .....             | 16  |
| 3.6.7 Berekenen gemiddelde V en B .....                                             | 17  |
| 3.6.8 Methode B: onderscheiden kopakkerpunten op basis Kmeans .....                 | 17  |
| 3.6.9 Methode C: onderscheiden kopakkerpunten op basis $\Delta\theta$ en V .....    | 18  |
| 3.6.10 Berekenen keermomenten op basis $\Delta\theta$ .....                         | 18  |
| 3.7 Vergelijken resultaten methode A, B en C .....                                  | 19  |
| 4 Resultaten.....                                                                   | 20  |
| 4.1 Scheiden Hoofdbewerking en Kopakker op basis van $\Delta\theta$ (methode A).... | 20  |
| 4.2 Scheiden hoofdbewerking en kopakker m.b.v. Kmeans (methode B) .....             | 21  |
| 4.3 Scheiden hoofdbewerking en kopakker m.b.v. $\Delta\theta$ en V (methode C)..... | 22  |
| 4.4 Vergelijken methode A, B en C kopakkerselectie .....                            | 26  |
| 4.5 Bepalen keermomenten op basis $\Delta\theta$ .....                              | 26  |
| 4.6 Resulterende Kengetallen .....                                                  | 30  |
| 5 Discussie .....                                                                   | 30  |
| 5.1 Wat is de beste onderscheidingsmethode? .....                                   | 30  |
| 5.2 Is de verzamelde data toereikend? .....                                         | 31  |

|     |                                                                |    |
|-----|----------------------------------------------------------------|----|
| 5.3 | Wat is de beste methode om polygoon K op te stellen? .....     | 31 |
| 6   | Conclusie.....                                                 | 32 |
| 7   | Aanbevelingen .....                                            | 33 |
| 8   | Referenties.....                                               | 35 |
| A1  | Overzicht percelen .....                                       | 36 |
| A2  | Opbouw Bestanden .....                                         | 37 |
| A3  | Resultaten kopakkerselectie op basis $\Delta\theta$ .....      | 38 |
| A4  | Resultaten kopakkerselectie op basis $\Delta\theta$ en V ..... | 41 |
| A5  | Resultaten 3 methoden kopakkerselectie .....                   | 44 |
|     | Perceel 1.....                                                 | 44 |
|     | Perceel 2.....                                                 | 44 |
|     | Perceel 3.....                                                 | 44 |
|     | Perceel 4.....                                                 | 44 |
|     | Perceel 5.....                                                 | 44 |
|     | Perceel 6.....                                                 | 44 |



# 1 Inleiding

In de continue zoektocht naar betere en efficiëntere werkmethoden in de landbouw speelt brandstofverbruik een steeds grotere rol. Voor een onderzoek binnen Plant Research International (PRI) zijn er brandstofmetingen uitgevoerd. Dit gebeurde met behulp van een brandstof-flow meter, gemonteerd op een New Holland TM125 tractor van het onderzoeksbedrijf Unifarm, onderdeel van WUR. De tractor is onder andere ingezet voor grondbewerkingen op percelen waar zowel kerende als niet kerende grondbewerking toegepast werd. Hierbij is het verbruik van de motor in liters per uur gemeten. Naast deze flowmeter is de proefopstelling ook uitgerust met een RTK-GPS systeem. Dit systeem bepaalt de positie en de rijsnelheid van de tractor bij elke brandstofmeting.

De verzamelde gegevens worden gebruikt door wetenschappers van PRI om uitspraken te doen over de effecten van wel en niet kerende grondbewerking op het brandstofverbruik. Het verkrijgen van dit soort meetgegevens zal in de toekomst verder bruikbaar blijven voor andere onderzoeken waar op dezelfde manier informatie verzameld wordt uit GPS en brandstofmetingen.

De op de tractor gemonteerde brandstofmeter slaat elke seconde de meting van de flow meter en het GPS systeem in een bestand op. Deze metingen zijn uitgevoerd tijdens het transport van en naar een perceel, de bewerking op het perceel, het keren op de kopakker, maar ook bijvoorbeeld stilstaan tijdens een pauze.

Er is gewerkt aan een algoritme om de meetgegevens om te zetten in de gewenste kengetallen. Met behulp van Labview is een VI gemaakt die hiervoor bewerkingen uitvoert. Het is mogelijk om manueel een indeling te maken tussen kopakker en hoofdbewerking. Verder moet ook handmatig het gedeelte van de metingen dat onder transport valt uit het bestand gehaald worden. Het uitvoeren van deze bewerkingen voor een beperkt aantal percelen is niet problematisch. Met het toenemen van het aantal metingen wordt het handmatig verwerken van de meetgegevens echter een tijdrovende bezigheid.

Voor het onderzoek waarvan deze metingen deel uitmaken zijn de volgende kengetallen gewenst:

- Brandstofverbruik in l/ha en l/h tijdens de hoofdbewerking;
- Brandstofverbruik in l/ha en l/h bij het keren op de kopakker;
- Het aantal keermomenten op de kopakker.

Het is op dit moment nog niet mogelijk om deze kengetallen aan de hand van de bestanden op een efficiënte en betrouwbare manier te verkrijgen. Om onderbouwde conclusies te kunnen trekken is dit echter wel nodig. De data moet gesorteerd worden om de hierboven beschreven kengetallen te verkrijgen.

Het doel van dit onderzoek is:

- De verkregen data uit de brandstof en GPS metingen via een algoritme verwerken tot de gewenste kengetallen.

De resulterende kengetallen moeten “GIS ready” zijn. Dat wil zeggen, eenvoudig in een kaart te verwerken ter visualisatie van de metingen.

Om het hierboven gestelde doel te bereiken zijn de volgende onderzoeksvragen beantwoord:

1. Welke parameters moeten gebruikt worden om de gewenste kengetallen uit de beschikbare bestanden te halen?
2. Hoe moet een algoritme met deze parameters de data verwerken om de gewenste kengetallen te produceren?

Het antwoord op de onderzoeksvragen staat in de conclusie. Om tot dit antwoord te komen is er in de literatuur gekeken naar andere onderzoeken op het gebied van brandstofmetingen. Hierbij is gekeken naar de wijze waarop deze onderzoeken zijn uitgevoerd en de resultaten. In het hoofdstuk materialen en methoden is besproken hoe het algoritme is opgebouwd en welke keuzes zijn gemaakt gedurende de ontwikkeling. In het hoofdstuk resultaten staan de resultaten van de verschillende methoden weergegeven.

## 2 Literatuur

### 2.1 Huidige systemen voor de verwerking van brandstofmetingen

Literatuuronderzoek naar brandstofmetingen resulteert in de vondst van een groot aantal publicaties die gaan over specifieke motortypen, gebruikt in landbouwmachines, met bijbehorende testresultaten. De resultaten omvatten publicaties in vakbladen als Boerderij, Trekker en Landbouwmechanisatie. Hierbij gaat het veelal om vergelijkbare onderzoeken waarbij hoofdzakelijk het specifiek brandstofverbruik bij een gegeven motortoerental en een onbelaste motor als maatstaaf gebruikt is. Deze onderzoeken en resultaten worden ook dikwijls door de producenten van de betreffende machines gebruikt om de zuinigheid aan te prijzen.

Ook zijn er verschillende onderzoeken (Tompkins and Carpenter, 1978), (Wiedemann *et al.*, 1986) gedaan naar trekkrachtbehoefte en brandstofverbruik van verschillende machines op verschillende grondsoorten om zodoende inzicht te krijgen in de juiste tractor voor deze bewerkingen/operaties. Een voorbeeld hiervan is het onderzoek van (McLaughlin *et al.*, 1993). Hierbij is gedurende verschillende grondbewerkingen de trekkrachtbehoefte en het brandstofverbruik gemeten. Hierbij is echter niet exact naar de positie van de tractor op het veld gekeken. De bewerkte oppervlakte was exact bekend en aan de hand hiervan zijn gemiddelden berekend. Hieruit kwam naar voren dat een trekker met groter motorvermogen dan strikt noodzakelijk doorgaans zuiniger met brandstof omgaat dan een kleinere trekker die op het maximum van zijn kunnen gebruikt wordt. Een ander voorbeeld hiervan is het onderzoek van (Kichler *et al.*, 2007)

waarbij gekeken is naar brandstofverbruik gekoppeld aan trekkrachtbehoefte. Bij beide onderzoeken werd een duidelijke relatie tussen trekkrachtbehoefte en brandstofverbruik vastgesteld.

Bij de zojuist genoemde onderzoeken werd niet gewerkt met GPS metingen. (Yule *et al.*, 1999) en (Yahya *et al.*, 2009) hebben wel onderzoeken gedaan op dit gebied.

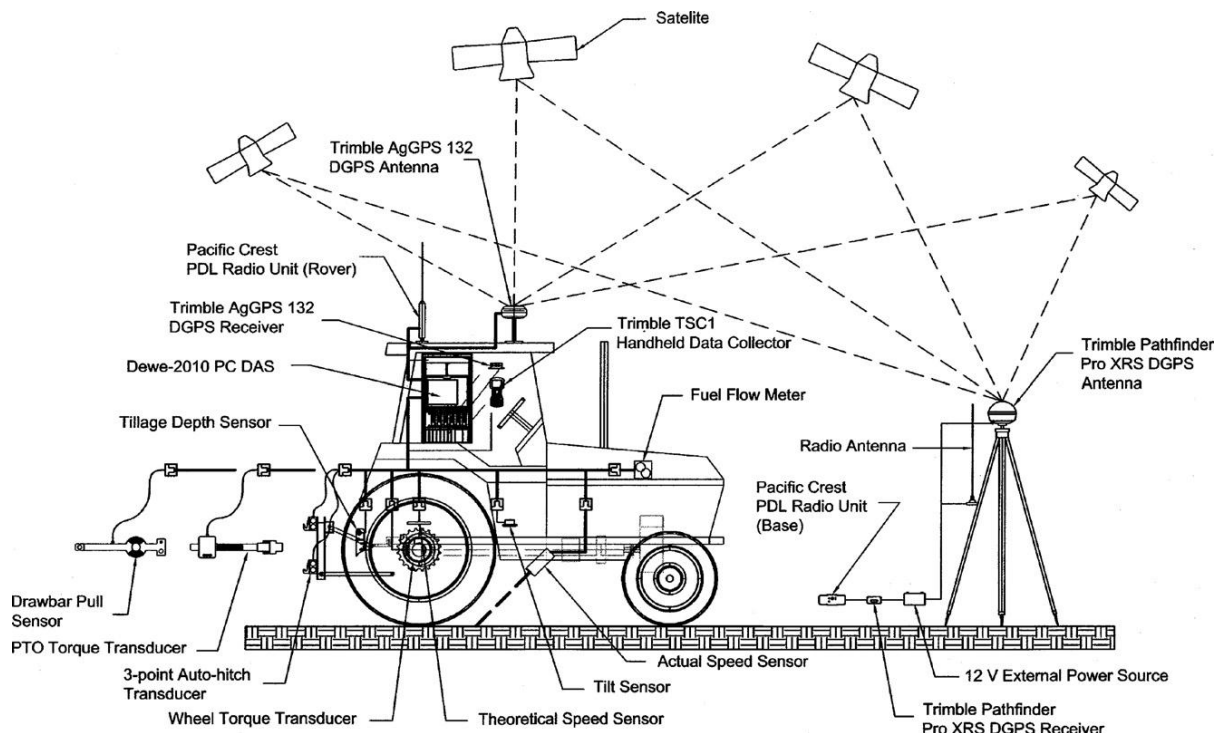
(Yule *et al.*, 1999) richtten zich op het opbouwen en testen van een realtime meetsysteem op een trekker. Hierbij werden tien parameters gemeten bij een grondbewerking uitgevoerd met een cultivator. Dit waren:

|                                      |                                |
|--------------------------------------|--------------------------------|
| motortoerental (omwentelingen/min)   | brandstofverbruik (l/h)        |
| koppel (Nm)                          | vermogen (kW)                  |
| specifiek brandstofverbruik (g/kW/h) | motorgebruik (%)               |
| theoretische snelheid (km/h)         | werkelijke snelheid (km/h)     |
| slipperpercentage (%)                | hellingspercentage perceel (%) |

Daarnaast was de trekker uitgerust met een D-GPS systeem dat aan elke meting een locatie toekende. Met behulp van deze locatiegegevens werden kaarten van elk perceel gemaakt voor onder andere de gemeten parameters hellingspercentage en slippercentage. De resultaten van de metingen werden op een monitor in de trekker cabine weergegeven en konden door de bestuurder gebruikt worden om de instellingen van de trekker aan te passen. Hierbij ging het hoofdzakelijk om opschakelen en het toerental reduceren wanneer de trekkrachtbehoefte dit toestond. Bij de metingen op een perceel werd slechts de hoofdbewerking in kaart gebracht. Tijdens het draaien op de kopakker registreerde het meetsysteem geen gegevens.

Als belangrijkste conclusie kwam uit dit onderzoek naar voren dat een dergelijk systeem brandstofbesparingen kan opleveren. Dit doordat de bestuurder beter in staat is de instellingen van de trekker te optimaliseren. Daarnaast bleek het mogelijk, door de kaarten met wielslip en hellingspercentage te combineren, probleemgebieden binnen de percelen te identificeren. Wielslip bleek namelijk zowel voor te komen op steile delen van het perceel alsook op natte plekken in het perceel. Deze natte plekken werden duidelijk in beeld gebracht en hier kan in de praktijk eventueel op ingespeeld worden door bijvoorbeeld te diepwoelen om storende lagen op te breken.

(Yahya *et al.*, 2009) gebruikten een Massey Ferguson 3060 trekker. Deze was uitgerust met een D-GPS systeem. Daarnaast was de trekker uitgerust met sensoren welke trekkracht op het aankoppelingspunt van het werktuig aan de trekker, het koppel en de theoretische snelheid van beide achterwielen, het koppel en de vermogensoutput van de aftakas (PTO), krachten op de driepuntshef en de werkdiepte van deze hef maten.



**Figuur 1** Overzicht van de sensoren gebruikt in het onderzoek van (Yahya *et al.*, 2009)

Met deze sensoren was het voor dit systeem mogelijk de volgende parameters te meten:

- Koppel op het linkertrekkerwiel (kNm)
- Koppel op het rechtertrekkerwiel (kNm)
- Koppeltotaal achterwielen (kNm)
- Theoretische snelheid (km/h)
- Werkelijke snelheid (km/h)
- Berekende slippercentage achterwielen (%)
- Brandstofverbruik (l/h)
- Trekkracht (kN)
- PTO koppel (Nm)
- Horizontale kracht in de linker en rechter hefarm (kN)
- Verticale kracht in de linker en rechter hefarm (kN)
- Werkdiepte van het gebruikte werktuig (cm)
- Pitch en Roll hoek in graden.

Het systeem en de gebruikte sensoren bleken na uitvoerig testen in de praktijk erg robuust en precies te zijn. Dit is dus een betrouwbaar systeem om tijdens lastige veldwerkzaamheden de hierboven genoemde parameters in kaart te brengen.

Bij (Yule *et al.*, 1999) en (Yahya *et al.*, 2009) ging het om het in beeld brengen van de trekkerprestaties. De gekoppelde GPS positie werd hoofdzakelijk gebruikt om een overzicht van het bewerkte perceel te maken. Aangezien het bij beide onderzoeken om Amerikaanse percelen ging werd de kopakker niet in de metingen meegenomen. Deze is verhoudingsgewijs een stuk kleiner op deze grote percelen dan de kleine Nederlandse percelen. In deze onderzoeken viel het op dat er naast brandstofverbruik en de GPS gegevens naar een aantal andere factoren

gekeken werd. Dit waren onder andere: de hoeveelheid koppel op de trekkerwielen, krachten op de driepuntshef en werkdiepte van de hef. Een onderzoek waarbij slechts met brandstofverbruik en GPS gegevens gewerkt werd en waarbij de kopakker meegenomen werd als deel van het perceel, kwam niet uit het literatuuronderzoek naar voren.

## **3 Materialen en Methoden**

### **3.1 Technische gegevens meetapparatuur**

Voor het ontwikkelen van een algoritme is het van belang te weten met welke precisie er gerekend dient te worden. Om een efficiënt algoritme te ontwikkelen dient dit zo goed mogelijk aan te sluiten op de precisie van de meetapparatuur gebruikt in de proefopstelling.

#### **3.1.1 Brandstofmeter**

De metingen op de New Holland TM125 werden uitgevoerd met een AIC-884 instructor flow meter (AIC, Zwitserland). De meter zit op de brandstofleiding van de tractor gemonteerd en meet de hoeveelheid diesel die de tank verlaat. De retourleiding van de brandstofpomp loopt niet terug de tank in maar door de meter welke de retourbrandstof direct terug meegeeft aan de toevoer. Zodoende wordt het daadwerkelijke verbruik gemeten. De meter heeft een meetbereik van 1 tot 80 l/h en een maximale afwijking van 0.01 tot 0.8 l/h. Voor de brandstofmetingen geldt een precisie van l/h.

#### **3.1.2 GPS systeem**

De proefopstelling reed met een RTK-GPS systeem. De coördinaten werden opgeslagen volgens het type WGS 84. Een dergelijk systeem is nauwkeurig tot op twee centimeter. Hierbij dient er wel rekening mee gehouden te worden dat er bij een dergelijk GPS systeem ook storingen kunnen optreden zoals het wegvallen van het signaal. In de bestanden is een aparte kolom opgenomen welke aangeeft of er geen losse stekkers of iets dergelijks de meting verstoorden. Er moet echter rekening mee gehouden worden dat niet 100% van de als “correct” gelabelde metingen correct is.

### **3.2 Lijst met eisen**

Voor het ontwikkelen van een passend algoritme op een efficiënte manier is het belangrijk een lijst met eisen op te stellen die weergeeft wat er van het algoritme verwacht wordt.

Het einddoel van de metingen is het produceren van de volgende kengetallen:

- Verbruik in l/ha en l/h tijdens de hoofdbewerking;
- Verbruik in l/ha en l/h bij het keren op de kopakker;
- Het aantal keermomenten op de kopakker.

Om van het oorspronkelijke bestand, opgebouwd zoals beschreven in paragraaf 3.3, de kengetallen te kunnen berekenen zijn de volgende eisen van belang:

1. Er is een duidelijke perceelsgrens;

2. Binnen deze perceelsgrens is een onderscheid gemaakt tussen slechts drie soorten metingen: kopakker, hoofdbewerking en overig;
3. Onder overig bevinden zich alle metingen waar de trekker niet bezig was met de hoofdbewerking of het keren op de kopakker;
4. Onder hoofdbewerking bevinden zich slechts metingen waar de trekker bezig is met het uitvoeren van de hoofdbewerking;
5. Onder kopakker bevinden zich slechts punten waar de trekker bezig is met het keren op de kopakker;
6. Er kan een telling gemaakt worden van elk keermoment;
7. Dit keermoment vormt een cluster van metingen behorende tot dit specifieke keermoment.

### 3.3 Bestanden

De brandstof flow meter en het GPS systeem produceren elk gegevens opgeslagen in een bestand. Elke seconde voert zowel de flow meter als het GPS systeem een meting uit resulterend in een bestand als Tabel 8 in bijlage A2. In de bijlage is de opbouw van deze bestanden uitgelegd.

### 3.4 Overzicht percelen

In het jaar 2010 zijn meetgegevens verzameld. De metingen omvatten de bewerkingen ploegen, woelen, zaaïen met een zaaicombinatie en transport. Deze bewerkingen werden uitgevoerd op verschillende percelen of de weg.

De beschikbare data bevatte in zijn originele vorm vaak alle metingen vanaf het starten tot het stilzetten van de tractor tijdens een bewerking. Hierbij zit ook het transport van en naar het perceel. Omdat de methode voor het opstellen van polygoon G om een perceel nog niet werkt wanneer er metingen buiten het perceel zitten (zie paragraaf 4.1) zijn de transportmetingen manueel verwijderd.

Uit de beschikbare percelen is een keuze gemaakt op basis van vorm en hoeveelheid meetpunten. Om het algoritme goed te kunnen testen zijn percelen met verschillende vormen nodig. De metingen uit 2010 bevatten één perceel met daarin een bocht. Dit is het perceel met de bestandsnaam: 2010-04-12-10-44-43-GPS-Fuel-Log-1. De methode, om een polygoon om een perceel op te stellen, werkte voor dit perceel niet naar behoren. Daardoor bleven er bruikbare percelen over met drie vormen:

- Regelmatige polygoon (rechthoek).
- Rechthoek + (dit is een rechthoekig perceel met een rechthoekig uitsteeksel)
- Driehoek

Uit de beschikbare bestanden zijn de percelen in Tabel 1 **Error! Reference source not found.** gekozen. Deze bevatten elk minimaal vierduizend datapunten. In dit verslag zijn de nummers voor de percelen gebruikt zoals afgebeeld in Tabel 1.

**Tabel 1 Perceelnummers**

| Perceel Nummer | Bestand                                |
|----------------|----------------------------------------|
| 1              | 2010-11-09-08-22-06-GPS-Fuel-Log-3.dat |
| 2              | 2010-10-27-11-56-17-GPS-Fuel-Log-2.dat |
| 3              | 2010-11-02-12-55-56-GPS-Fuel-Log-1.dat |
| 4              | 2010-11-16-12-41-51-GPS-Fuel-Log-1.dat |
| 5              | 2010-11-22-14-05-13-GPS-Fuel-Log-1.dat |
| 6              | 2010-04-21-08-24-39-GPS-Fuel-Log-2.dat |

Bijlage A1 bevat een overzicht van de percelen horende bij de bestanden uit Tabel 1 in het XY vlak.

### 3.5 Overzicht algoritme

Figuur 2 geeft een overzicht, van de verschillende onderdelen, van het ontwikkelde algoritme.

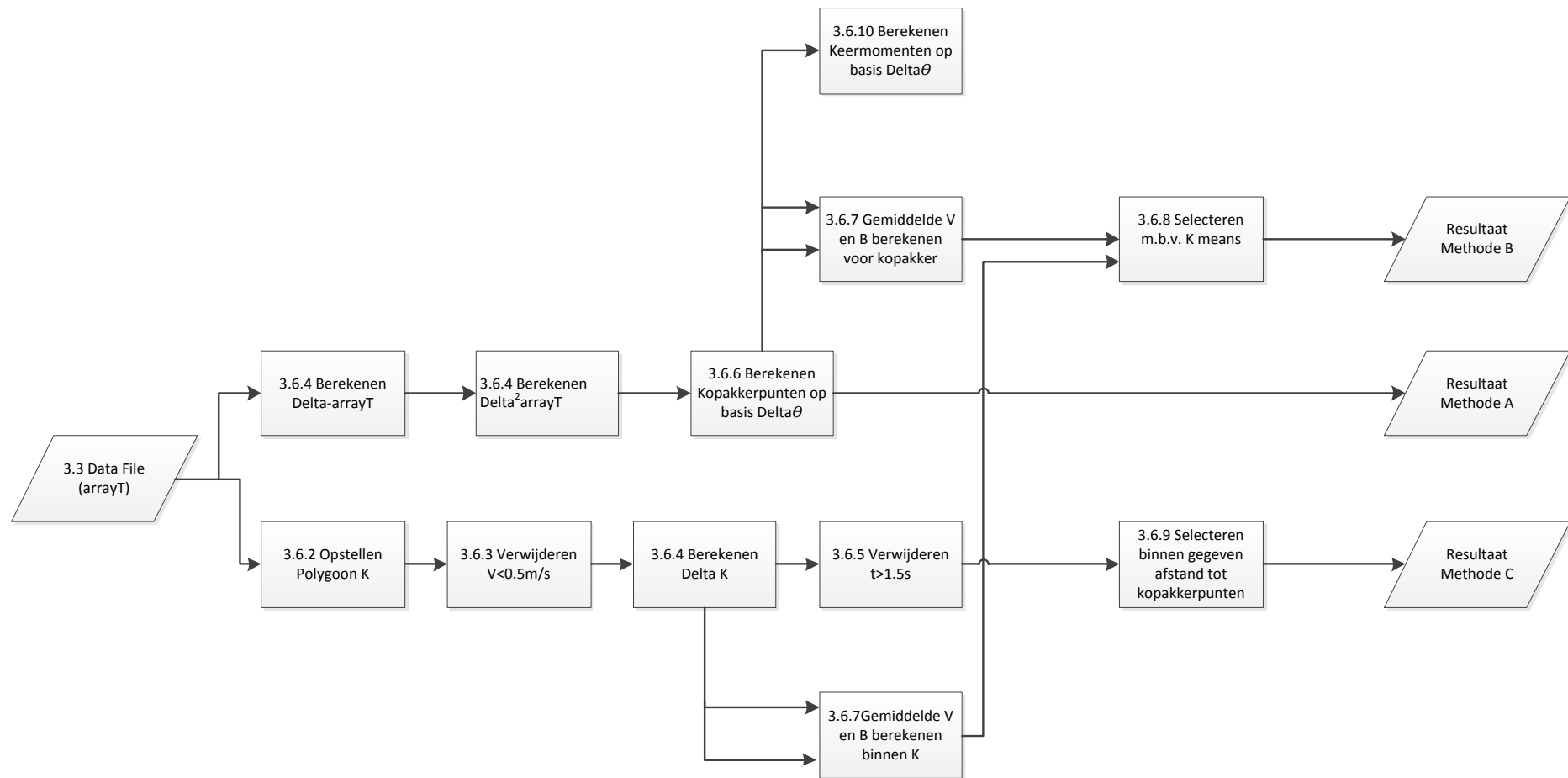
Uit de beschikbare data kunnen drie parameters worden afgeleid die bruikbaar zijn voor het scheiden van de hoofdbewerking en kopakker. Deze drie parameters zijn:

- Rijrichting ( $\theta$ )
- Rijsnelheid ( $V$ )
- Brandstofverbruik ( $B$ )

Voor het scheiden van de kopakker en de hoofdbewerking zijn drie methoden geprobeerd.

- A. Scheiden op basis van rijrichtingsverandering ( $\Delta\theta$ ).
- B. Scheiden met behulp van Kmeans.
- C. Scheiden op basis van rijrichtingsverandering en rijsnelheid.

De gemiddelden van de parameters zijn berekend voor zowel de hoofdbewerking als de kopakkerpunten. Er is gecontroleerd of de punten, gebruikt voor de berekening van de gemiddelden, correct waren ingedeeld. In de volgende alinea is per stap van het algoritme beschreven wat de functie en de werking is.



**Figuur 2** overzicht algoritme. ArrayT bevat alle metingen van één bestand. V is de snelheid van de trekker in m/s, B is het brandstofverbruik in l/h.  $\theta$  is de rijrichting. Polygoon K is een polygoon welke slechts hoofdbewerkingspunten binnen het perceel omvat. K is een array met alle metingen binnen polygoon K. t is de tijd in seconden, binnen Delta K is dit de tijd tussen twee metingen. Kmeans is een rekenmethode om in dit algoritme kopakkerpunten en hoofdbewerkingspunten te selecteren Dit is methode B. Methode A berkend kopakkerpunten en hoofdbewerkingspunten op bais van Delta  $\theta$ . Methode C maakt gebruik van Delta $\theta$  en de rij snelheid om hoofdbewerkingspunten en kopakkerpunten van elkaar te onderscheiden.



### 3.6 Processen in het ontwikkelde algoritme

Aan de hand van de in Figuur 2 gegeven nummering zijn de berekeningen beschreven zoals het algoritme deze uitvoerde.

#### 3.6.1 Analyse bestaande Labview Algoritme

Twee SubVI's in het bestaande algoritme bleken bruikbaar in meerdere delen van het nieuw ontwikkelde gedeelte.

- *SUB-BoundaryPolygon*

Deze is gebruikt voor de berekening van K, hiertoe is de bestaande SubVI BoundaryPolygon aangepast zoals beschreven in paragraaf 3.5.2 en hernoemd tot BoundaryPolygonK.

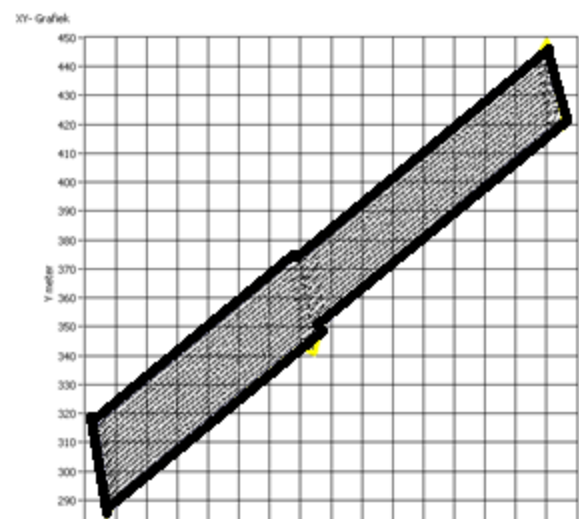
- *SUB-Polygonselections*

Deze SubVI deelt de data van een totale array met daarin een polygoon op in twee arrays waarvan er één binnen en één buiten de polygoon ligt.

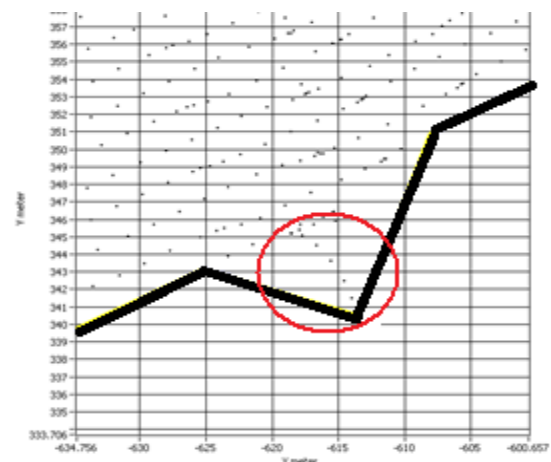
Deze twee SubVI's zijn op een aantal plaatsen gewijzigd. Er is gekeken hoe de methode om een polygoon om een dataset op te stellen werkt zodat deze aangepast kon worden voor het opstellen van polygoon K.

Er was een programma geschreven dat een bestand inlaadde in LabView. Dit bestand bevatte op dat moment nog alle data. Deze zijn in een XY grafiek geplott. Uit het datafile zijn in dit deel van het algoritme alle metingen verwijderd waar Lat of Lon 0 zijn, er een fout in de DistMeter zit of de errorcode een error aangeeft.

Om de XY data voor het hele perceel is door de SubVI BoundaryPolygon een polygoon getekend welke de perceelgrens weergeeft. Dit is te zien in Figuur 3. Binnen deze SubVI is vanuit de oorsprong een vector getekend naar het dichtstbijzijnde punt van het hele perceel in het XY vlak. Vervolgens is er een cirkel getrokken met dit punt als middelpunt. Wanneer deze cirkel een punt tegenkomt is dit het volgende punt van polygoon G. Om dit punt is vervolgens opnieuw een cirkel getrokken. Dit proces herhaalt zich tot een eerder gevonden punt bereikt is, waarna de berekening stopt. Voor een correcte werking van deze SubVI is het erg belangrijk dat het datafile slechts punten bevat binnen het perceel. Zoals te zien is in Figuur 4 worden bijvoorbeeld punten waar de trekker het perceel binnen kwam rijden anders meegenomen



Figuur 3 Polygoon G (perceelsgrens) om perceel



Figuur 4 Aankomst op het perceel wordt door polygoon G meegenomen als punten behorende tot de bewerkingen op het perceel

bij het opstellen van de polygoon. Deze punten horen nog bij transport en niet bij de bewerking van het perceel.

De tweede SubVI die hergebruikt is, is de SubVI PolygonDataSelector. Deze berekent op basis van een grenspolygoon zoals G of K welke punten van het perceel hierbinnen of buiten liggen. Deze SubVI bevat een stacked sequence structure waar in stappen verschillende berekeningen uitgevoerd zijn. In de eerste stap werd de data van de polygoon geïnterpoleerd. Dit zorgde ervoor dat de benodigde rekentijd van dit deel van het algoritme sterk toenam. Om het benodigd aantal berekeningen niet te groot te laten worden is ervoor gekozen de interpolatie uit te schakelen. Alle gebruikte polygonen bestaan bovendien uit voldoende punten om een goede selectie te kunnen maken van punten binnen en buiten de polygoon.

### **3.6.2 Opstellen polygoon K**

De XY data uit de brandstofmetingen vormen een vectorveld. Hierom is een polygoon (G) opgesteld om de grens van het perceel weer te geven. Deze polygoon kan op meerdere manieren opgesteld worden. In het bestaande algoritme was hiervoor een Sub-VI aanwezig. In het algoritme is het gemiddeld brandstofverbruik en de gemiddelde snelheid voor de hoofdbewerking berekend. Dit is gedaan binnen een polygoon K, een verkleinde versie is van polygoon G, die in het midden van het perceel een aantal datapunten omsluit welke zeker tot hoofdbewerking behoren.

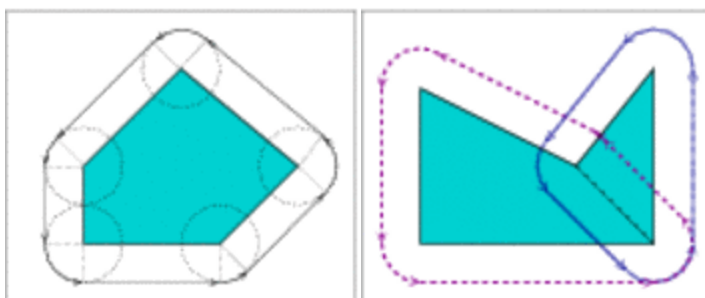
Om te bepalen welke punten behoren tot hoofdbewerking en welke tot kopakker, zijn de parameters rijnsnelheid ( $V$ ) en rijrichting ( $\theta$ ) gebruikt. Stap 3.6.2 bestaat uit het opstellen van een polygoon K met dezelfde vorm maar een kleinere diameter dan polygoon G. Dit is gedaan om zo een polygoon te verkrijgen waarbinnen datapunten liggen waarvan zeker is dat ze tot de hoofdbewerking behoren. Om dit te doen is in literatuur gezocht naar methoden voor het opstellen van een polygoon binnen een vectorveld waarbij de vorm gelijk is aan een originele polygoon.

#### **3.6.2.1 Offsetting**

Offsetting van een polygoon is het berekenen van de Minkowski som van een simpele polygoon met een cirkel (Wein, 2007) Dit proces wordt in drie stappen uitgevoerd:

1. Het berekenen van een nieuwe polygoon met de gewenste afstand tot de originele polygoon met behulp van cirkelbogen en rechte lijnstukken.
2. Het selecteren van de delen van de cirkelbogen en de rechte lijnstukken welke samen de nieuwe polygoon vormen.
3. Het correct combineren van verschillende lijnstukken welke elkaar snijden om een correcte output te hebben bij een niet convexe polygoon.

Hiermee is het mogelijk om bij afwijkende vormen van polygoon G een correcte polygoon K op te stellen zoals te zien is in Figuur 5.



Figuur 5 Offsetting van een convexe en niet convexe polygoon (Wein,2007).

### 3.6.2.2 Scaling

Een andere mogelijkheid voor het opstellen van een kleinere polygoon binnen en met gelijke vorm aan een grotere polygoon is scaling. Bij scaling wordt de bestaande figuur met een factor  $x$  verkleind met behoud van vorm. (Gonzalez and Woods, 2008) beschrijven de wiskundige uitwerking van scaling. Hierbij gaat het om matrix 2.1 en de formules 2.2 en 2.3 voor de coördinaten.

$$\begin{bmatrix} Cx & 0 & 0 \\ 0 & Cy & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad 2.1$$

$$X = Cx * v \quad 2.2$$

$$Y = Cy * w \quad 2.3$$

Waarin:

$Cx$  = schaaltransformatie in de X richting.

$Cy$  = schaaltransformatie in de Y richting.

$(v,w)$  = coördinaten van de punten in de originele figuur.

$(X,Y)$  = coördinaten van de punten in het getransformeerde figuur.

De transformatie van de coördinaten kan uitgedrukt worden in formule 2.3

$$(X,Y) = T\{(v,w)\} \quad 2.3$$

Zo wordt dus bijvoorbeeld met de berekening  $(X,Y)=(v/2,w/2)$  het getransformeerde figuur half zo groot in zowel de X als de Y richting. Hierbij is het belangrijk er rekening mee te houden dat voor deze berekeningen het centrum van het perceel op de coördinaten (0,0) in het XY-vlak geplaatst wordt. Nadat scaling heeft plaatsgevonden, wordt het centrum van het perceel terug op de originele XY coördinaten geplaatst.

### 3.6.2.3 Verkleinen met behulp van vectoren

Naast de in literatuur gevonden methoden was er nog een mogelijkheid om polygoon K op te stellen. In het bestaande algoritme was een SubVI-BoundaryPolygon aanwezig. Deze berekent polygoon G, de grenspolygoon die het perceel omsluit. In paragraaf 3.6.1 staat een analyse van de werking van deze SubVI-BoundaryPolygon.

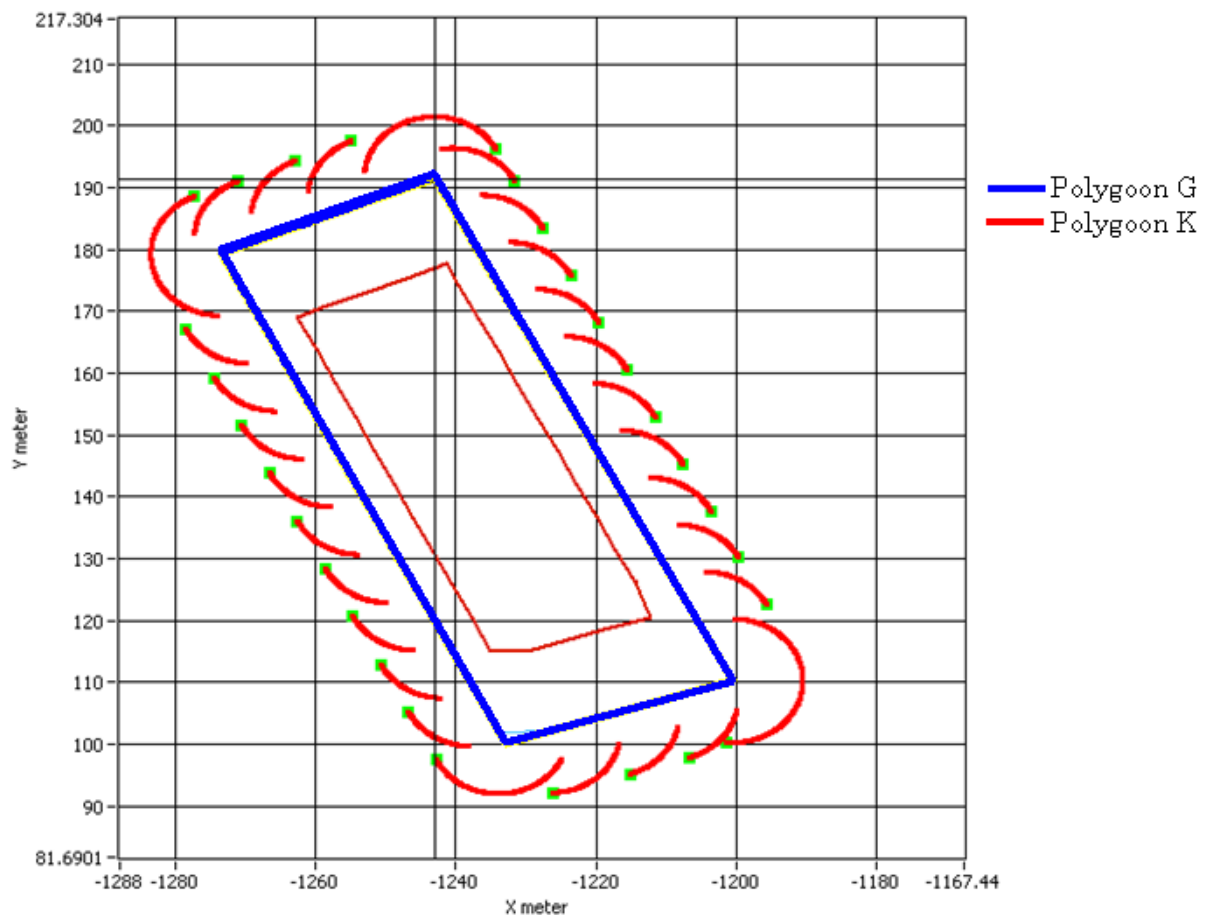
SubVI-BoundaryPolygon is aangepast om polygoon K op te stellen. Hierbij is de cirkel, die rond elk punt van polygoon G getrokken wordt om het volgende punt

van de polygoon te vinden,  $\pi/2$  doorgetrokken. Zo eindigt deze cirkel loodrecht en aan de binnenzijde van polygoon G. Dit punt bevindt zich dus op een vaste afstand van polygoon G, namelijk de straal van de cirkel. Door deze straal in te stellen is de gewenste afstand van polygoon K tot polygoon G aan te passen..

In Figuur 6 is te zien wat het resultaat is van het opstellen van polygoon K binnen polygoon G met deze methode. Ook zijn hierin delen van de cirkels te zien welke gebruikt zijn om polygoon G op te stellen. Deze cirkels een  $\pi/2$  doortekenen levert een goed resultaat op voor de rechte stukken van polygoon G en K.

Echter, op de punten waar polygoon G een hoek maakt draait de cirkel ook  $1,5 \pi$  vanuit het startpunt. Dit is steeds het vorige punt. Wanneer een hoek gemaakt is draait de cirkel meer dan  $\pi$  voordat het volgende punt van polygoon G gevonden is. Het volgende punt van polygoon K komt daardoor in het verlengde van de originele richting te liggen. Hierdoor komt polygoon K in de hoekpunten van polygoon G over kopakkerpunten te liggen. Dit is een ongewenst resultaat, hierdoor is deze methode niet te gebruiken.

XY- Grafiek bevattende polygoon G en K



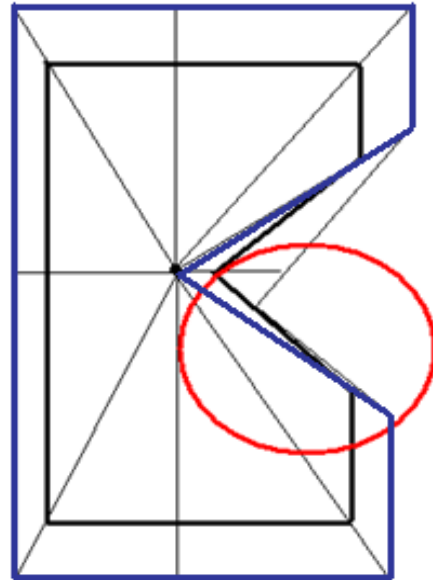
Figuur 6 Polygoon K binnen Polygoon G in het XY vlak inclusief delen van de cirkels gebruikt om polygoon G op te stellen

Voor de verdere berekeningen is een polygoon K nodig, hiervoor is SubVI-BoundaryPolygon op een andere manier aangepast. Alle punten van de perceelsgrenspolygoon G zijn beschikbaar. Daarnaast is ook het middelpunt van

het perceel bekend. Vanuit dit middelpunt is een vector getrokken naar elk punt van polygoon G. Al deze vectoren zijn met een vaste factor verkleind. De eindpunten van de vectoren vormen hierna polygoon K.

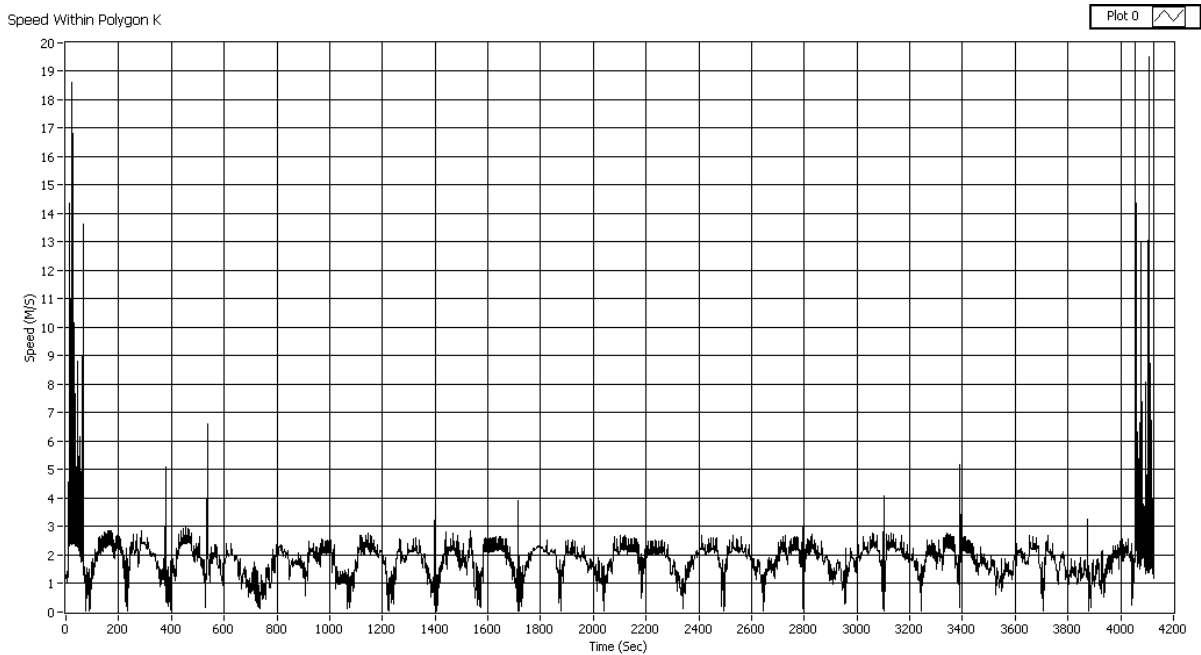
De toegepaste aanpak is getest door een plot te maken van polygoon K in het XY vlak. Zoals in Figuur 6 te zien is resulteert dit in een rechthoekig perceel in een polygoon K die slechts hoofdbewerkingspunten omvat.

Een belangrijke observatie bij andere perceelsvormen was dat deze methode niet afdoende is. Het opstellen van de polygoon K vanuit het middelpunt werkt alleen correct bij vierkante of rechthoekige percelen. Afwijkende perceelsvormen zorgen voor een polygoon K die niet overeenstemt met polygoon G en zelfs buiten het perceel kan komen te liggen. Dit is zichtbaar in Figuur 7 waar polygoon K de dikgedrukte lijn is en polygoon G de blauwe lijn. Het afsnijden van de punten die wegvallen is veroorzaakt doordat alle vectoren opgesteld en ingetekend zijn vanuit het middelpunt. Bij percelen waar hoeken uit ontbreken, zijn deze hoeken door de vectoren afgesneden. Dit resulteerde in fouten in polygoon K. Deze fouten zijn onder andere het tot hoofdbewerking rekenen van kopakkerpunten waar deze binnen K vallen. Dit moet vermeden worden aangezien de punten binnen K zuiver hoofdbewerking moeten zijn voor later gebruik.

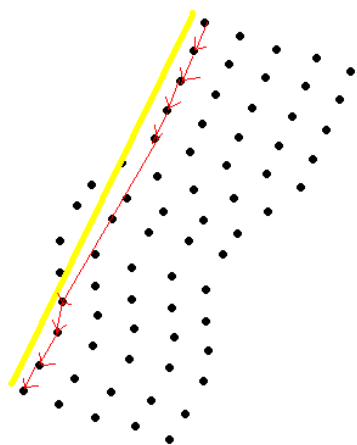


**Figuur 7 Polygoon K(dikgedrukt) snijdt Polygoon G(blauw)**

De percelen waar het algoritme op getest is bestaan uit drie rechthoeken, twee rechthoeken met een uitsteeksel en een driehoek. In bijlage A1 zijn deze perceelsvormen zichtbaar. Voor deze percelen is de XY data geplott, bij een verkleiningsfactor twee voor de vectoren, omvatte polygoon K voor alle percelen uitsluitend hoofdbewerkingspunten. Deze aanpak is dan ook bruikbaar voor deze zes percelen maar zoals gezegd niet robuust voor elke perceelsvorm. Er is gekozen hiermee verder te werken.



**Figuur 8 Snelheid (V) van de trekker tijdens de hoofdbewerking (metingen binnen polygoon K) op het perceel**



**Figuur 9 Het afsnijden van punten binnen dezelfde werkgang door polygoon K. Geel: Polygoon K, Rood: Vectoren**

Figuur 8 is het resultaat van deze datasetselectie binnen polygoon K. De snelheid binnen polygoon K gedurende het verloop van de meting is weergegeven. Hierin is te zien dat de snelheid pieken vertoont aan het begin en het eind van de metingen binnen polygoon K. Dit is veroorzaakt door het feit dat polygoon K punten in dezelfde werkgang afsnijdt zoals te zien in figuur 9. Polygoon K loopt namelijk niet langs gemeten datapunten maar is op een vaste afstand van het middelpunt getekend. Daardoor vallen sommige gedeeltes van werkgangen waar de vorm niet overeenkomt met die van polygoon K gedeeltelijk binnen en gedeeltelijk buiten polygoon K.

Omdat het voor de berekening van de richting en snelheid van belang is dat elke twee opeenvolgende punten in de array, met metingen binnen polygoon K, ook daadwerkelijk de volgende zijn. Is hier nog een filter operatie op basis van tijd (t) op toegepast. Deze filter operatie staat in paragraaf 3.6.5 beschreven.

Er was reeds een SubVI-PolygonDataSelector beschikbaar welke op basis van een polygoon, data van het hele perceel binnen en buiten deze polygoon selecteert. Door polygoon K als grenspolygoon in deze SubVI te stoppen is data verkregen binnen polygoon K welke uitsluitend hoofdbewerkingspunten bevat. Dit is data K.

### 3.6.3 Verwijderen te lage V binnen K

Nu er een dataselectie gemaakt is van een gedeelte van het perceel met slechts hoofdbewerkingspunten moet deze data nog wel gefilterd worden. Stilstaan om een willekeurige reden levert metingen op die niet tot de hoofdbewerking horen.

Deze worden verwijderd door alleen de regels van K naar een nieuwe array K te schrijven die voldoen aan de voorwaarde dat de rijsnelheid  $< 0.5$  m/s.

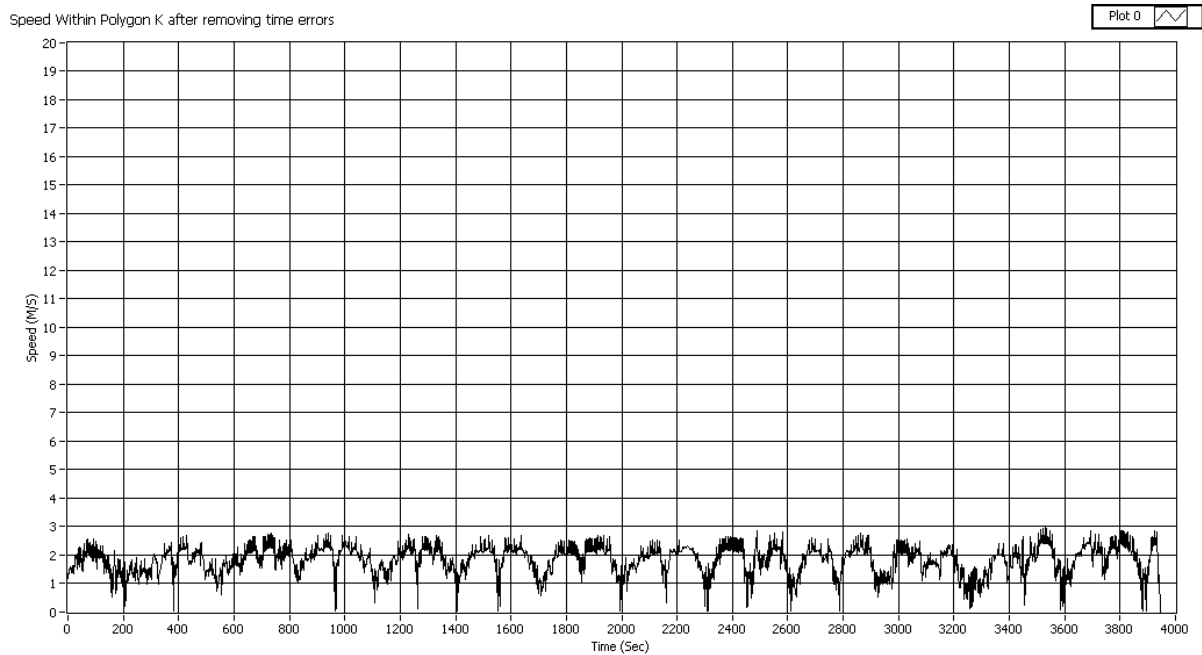
Zo ontstaat een nieuwe array K welke slechts punten bevat waar minimaal 0.5 m/s gereden werd. De waarde 0.5 is gekozen omdat voor de bewerkingen ploegen, zaaien en woelen (waarvan bij de metingen sprake is) de snelheid van de hoofdbewerking nooit onder deze waarde zal liggen. Er liggen wel waarden tussen 0 en 0.5 welke niet meegerekend dienen te worden.

### 3.6.4 Berekenen Delta K, Delta T en Delta<sup>2</sup> T

Het verschil tussen elke twee opeenvolgende metingen binnen het gehele perceel is berekend. Hiermee ontstaat array DeltaT(totaal). Uit het verschil tussen de locatie in een meting en de daaropvolgende meting kan de rijrichting,  $\theta$ , bepaald worden. Het verschil tussen de rijrichting in een punt en het daaropvolgende punt is de verandering van de rijrichting. Deze rijrichtingsverandering, Delta  $\theta$ , komt dus uit een array welke het verschil bevat tussen elke meting en de daaropvolgende meting in de tijd van alle data binnen array Delta T. Alle richtingsveranderingen tussen twee opeenvolgende punten samen vormen array Delta  $\theta$ .

### 3.6.5 Verwijderen $t < 1.5$ sec

Er diende nog een filter operatie plaats te vinden om fouten als gevolg van afsnijden door polygoon K te verwijderen. Voor alle metingen die voldoen aan het criterium  $\Delta t < 1.5$  is een nieuwe array aangemaakt. Deze t is de eerste kolom binnen array K en bevat de tijd van elke meting, waartussen steeds één seconde hoort te zitten. In Figuur 10 staat het resultaat van deze filter operatie. Door alle  $\Delta t > 1.5$  te verwijderen bleef een bruikbare array DeltaK over zonder fouten. De pieken in snelheid veroorzaakt door de te grote afstand tussen twee punten in de tijd zijn nu verdwenen.

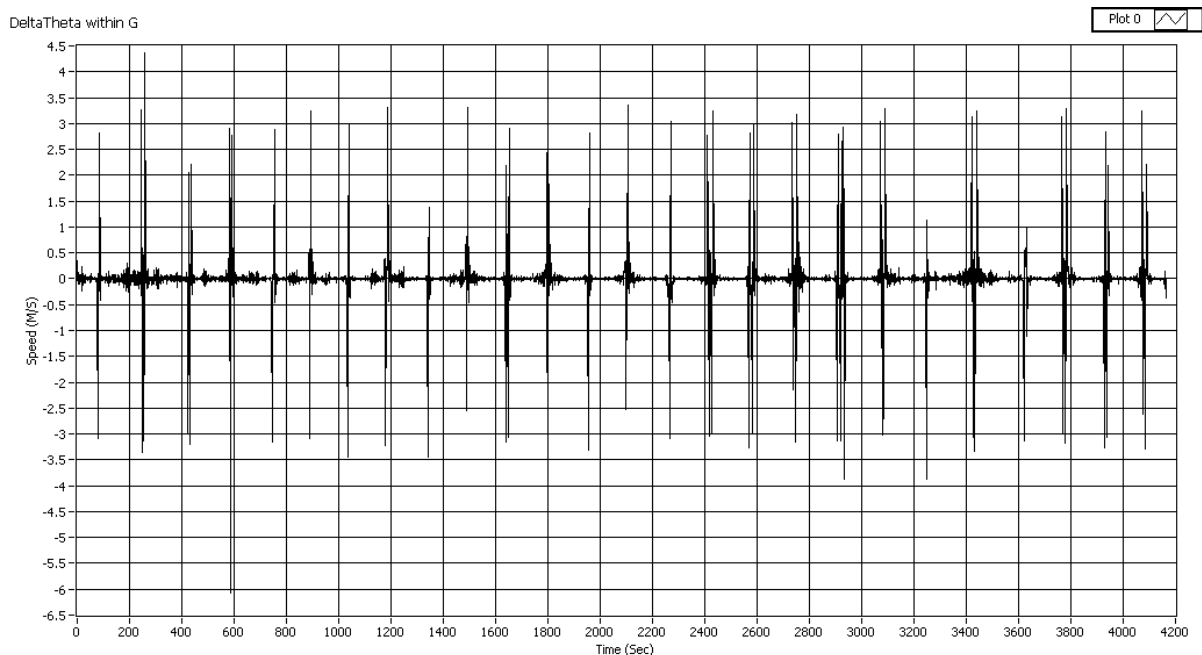


**Figuur 10 Snelheid V binnen K na filter operatie**

### **3.6.6 Methode A: Berekenen kopakkerpunten op basis $\Delta\theta$**

De verandering van rijrichting ten opzichte van elke vorige meting is  $\Delta\theta$  genoemd. Deze is afgeleid uit een X en Y coördinaat voor elk punt van het perceel. Deze zijn door het algoritme omgezet in een r en een  $\theta$ . r is de snelheid tussen twee punten,  $\theta$  de rijrichting en  $\Delta\theta$  dus de verandering van de rijrichting. De trekker waarop de meetapparatuur gemonteerd zit, voert de bewerkingen ploegen, zaaien en woelen uit. Bij deze bewerkingen wordt een vrij constante rijrichting aangehouden tijdens de hoofdbewerking. Het is niet mogelijk om binnen enkele meters een scherpe bocht te maken met een ploeg. Hierdoor kan er vanuit gegaan worden dat tijdens de hoofdbewerking de rijrichting gelijk blijft of slechts een kleine afwijking vertoont. Op basis van dit gegeven zijn kopakkerpunten gescheiden van hoofdbewerkingspunten. Door de  $\Delta\theta$  voor elke twee opeenvolgende punten van het perceel te bekijken, is een constante waarde zichtbaar met uitschieters tijdens keermomenten waarbij scherpe draaien gemaakt worden. Dit is zichtbaar in Figuur 11.





**Figuur 11  $\Delta\theta$  gedurende de bewerking van een perceel**

Het gemiddelde en de standaardafwijking van  $\Delta\theta$  zijn berekend. De data is naar kopakker geschreven op basis van het criterium;  $\Delta\theta \geq 1 \cdot \sigma$ .

Dit is voor alle zes percelen uitgevoerd. Dit is methode A voor het bepalen van de kopakkerpunten. Het resultaat van methode is met de andere methoden vergeleken in hoofdstuk 4.

### **3.6.7 Berekenen gemiddelde V en B**

Om met behulp van Kmeans (methode B) de punten binnen een perceel in te kunnen delen op kopakker of hoofdbewerking zijn de gemiddelden nodig van alle gebruikte parameters voor zowel kopakker als hoofdbewerking. Door het opstellen van polygoon K en het opschonen hiervan, is er een array met slechts hoofdbewerkingspunten beschikbaar. Door het scheiden op basis van  $\Delta\theta$  (methode A) is er een array beschikbaar gekomen met slechts kopakkerpunten. Deze zijn gebruikt om de gemiddelden en standaarddeviaties te berekenen voor de parameters:

- Rijrichtingsverandering  $\Delta\theta$
- Brandstofverbruik B
- Snelheid V

### **3.6.8 Methode B: onderscheiden kopakkerpunten op basis Kmeans**

In de artikelen (Kanungo *et al.*, 2000) en (Kanungo *et al.*, 2002) is het Kmeans clustering algoritme beschreven. Matlab is gebruikt om te zien hoe effectief deze methode kopakkerpunten kan scheiden van hoofdbewerkingspunten. Kmeans is een iteratief clustering algoritme dat als volgt werkt. Een dataverzameling wordt ingedeeld naar een N aantal clusters op basis van parameters. Voor elke parameter is een gemiddelde nodig, voor elk van de N clusters. Vervolgens wordt de data doorlopen en ingedeeld op basis van de grootste overeenkomst met elk van de parameters van cluster N. Dit proces wordt herhaald tot de grootste overeenkomst met elk van de gemiddelden bereikt is.

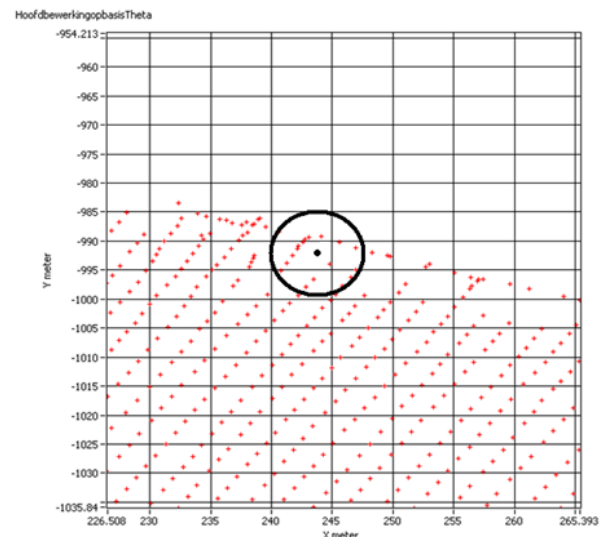
Om Kmeans in Matlab te gebruiken zijn de volgende gegevens nodig:

- Gemiddelden voor de drie parameters  $\Delta\theta$ , V en B, voor zowel kopakker (cluster1) als hoofdbewerking (cluster 2).
- Een txt bestand met daarin voor alle punten van het perceel de drie parameters  $\Delta\theta$ , V en B.

De resultaten de Kmeans berekening in Matlab staan in paragraaf 4.5.

### 3.6.9 Methode C: onderscheiden kopakkerpunten op basis $\Delta\theta$ en V

De derde methode om de kopakkerpunten en de hoofdbewerkingpunten te onderscheiden is die op basis van  $\Delta\theta$  en V. Hierbij is om de punten die op basis van  $\Delta\theta$  (methode A) gevonden zijn een cirkel met een straal van drie meter getekend. Deze cirkel is cirkel C genoemd. Vervolgens is de SubVI PolygonDataSelector gebruikt om de data binnen deze cirkel te selecteren. In Figuur 12 is een voorbeeld van zo'n cirkel en de punten die daarbinnen vallen zichtbaar. De zwarte stip is een kopakkerpunt, gevonden met behulp van  $\Delta\theta$ . Van deze datapunten is de snelheid vergeleken met de gemiddelde snelheid voor de hoofdbewerking zoals deze in paragraaf 3.6.7 berekend is. Wanneer de snelheid minder dan 1x de standaardafwijking van de gemiddelde snelheid afwijkt, is deze tot hoofdbewerking gerekend. Is dit niet het geval dan is deze tot kopakker gerekend. De resultaten van deze derde methode staan in paragraaf 4.3.



**Figuur 12** Cirkel waarbinnen gekeken wordt naar de snelheid V (methode C) met als middelpunt (zwart) een kopakkerpunt gevonden met methode A

### 3.6.10 Berekenen keermomenten op basis $\Delta\theta$

Om het aantal keermomenten op de kopakker vast te stellen zijn twee methoden beproefd.

Als eerste is de mogelijkheid voor peak detection in Labview geprobeerd. Dit is toegepast op  $\Delta\theta$  absoluut.  $\Delta\theta$  gaat door een Sample Compression, hierdoor is een trendlijn door alle datapunten getrokken. Door de hoeveelheid punten waaruit deze trendlijn bestaat te verkleinen, blijven slechts de grootste uitwijkingen over. Zodoende kan de peak detector deze beter waarnemen.

Omdat elk keermoment meer dan één piek laat zien is gezocht naar de beste instelling voor de sample compression. Dus in hoeverre het aantal datapunten gereduceerd moet worden. Dit gebeurt door voor perceel 1 te bekijken of het aantal van 27 keermomenten klopt met de output van de peakdetector. In paragraaf 4.5 wordt het resultaat van deze peak detection besproken.

Omdat de peakdetection verkeerde resultaten opleverd wanneer de verkeerde pieken gedetecteerd worden, is er een FFT test VI aangemaakt waarin een power spectrum opgesteld is van  $\Delta\theta$ . Dit is gedaan met de bedoeling te onderzoeken in hoeverre de pieken duidelijk waarneembare frequenties vertonen. Zo is beter zichtbaar waar de pieken in de grafiek met  $\Delta\theta$  liggen en of deze met behulp van een Low Pass Filter detecteerbaar zijn.

### **3.7 Vergelijken resultaten methode A, B en C**

Tot slot zijn de resultaten van de drie methoden om de kopakkerpunten en hoofdbewerkingspunten te onderscheiden vergeleken. Dit is gedaan in MatLab.

Hiervoor is voor methode A in LabView aan de punten hoofdbewerking binnen polygoon K het label 2 gehangen. De kopakkerpunten bepaald op basis van  $\Delta\theta$  krijgen het label 1 en de punten die niet tot een van deze twee clusters behoren krijgen het label 0 (onbekend). Dit staat in een kolom van een outputfile “perceeltotaal”.

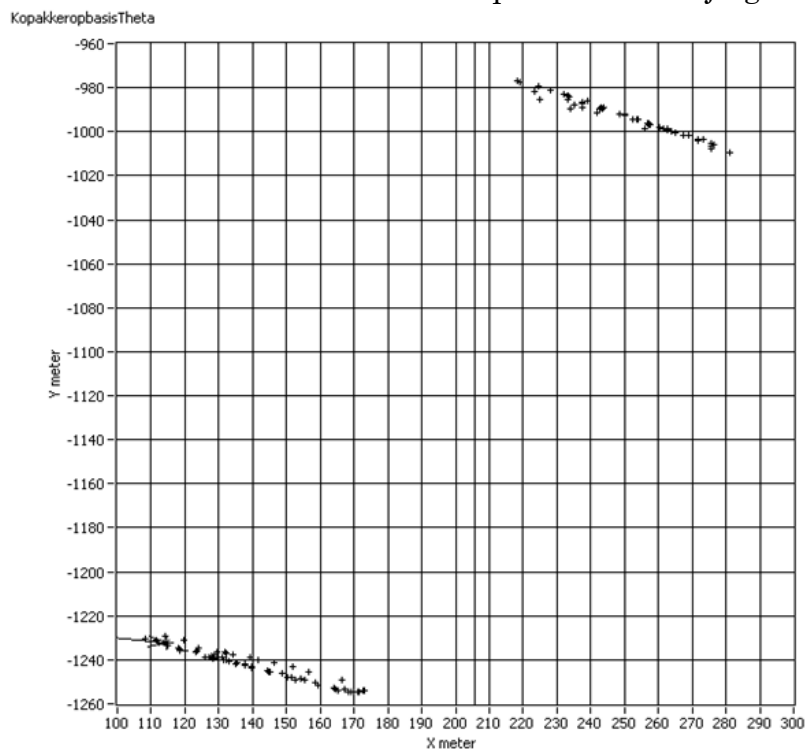
Dit bestand bevat een tweede kolom waarin voor de punten waar methode C kopakker als resultaat gaf, een 1 en een 2 waar deze tot hoofdbewerking behoorde.

De derde kolom is gevormd door de Kmeans methode in MatLab waar ook voor kopakker een 1 en voor hoofdbewerking een 2 als label gebruikt zijn. Deze drie kolommen zijn vergeleken om te zien wat de overeenkomsten en verschillen tussen de drie methoden zijn. Daarnaast zijn voor alle drie de methoden de kopakkerpunten in het XY vlak geplot om zo visueel te controleren waar deze ingedeeld worden. De resultaten hiervan staan in paragraaf 4.4

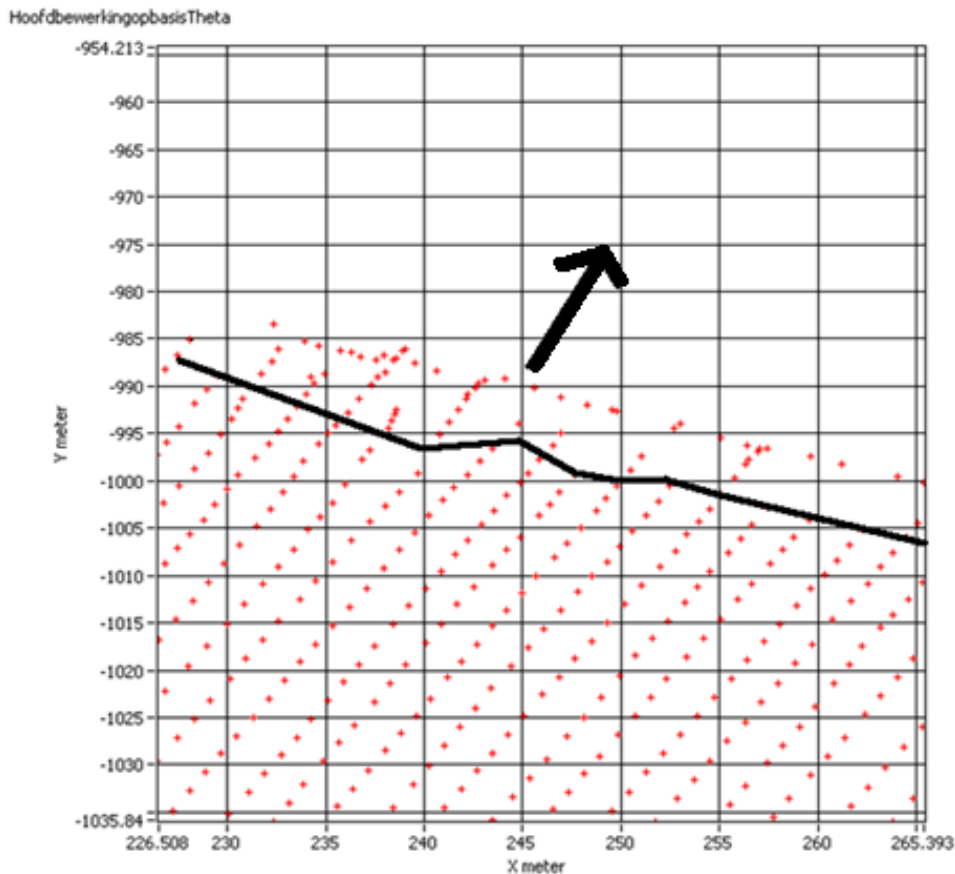
## 4 Resultaten

### 4.1 Scheiden Hoofdbewerking en Kopakker op basis van $\Delta\theta$ (methode A)

Om te controleren of de methode het beoogde resultaat oplevert zijn de kopakkerpunten geplot in het XY vlak. Voor perceel 1 zijn er 121 gevonden. Deze punten liggen duidelijk op de kopakker. Wanneer echter de resterende punten geplot worden in het XY vlak is duidelijk zichtbaar dat er nog een significant gedeelte van de kopakkerpunten niet onderscheiden is van de hoofdbewerkingspunten. In Figuur 13 en Figuur 14 is te zien wat deze methode voor perceel 1 aan kopakker en hoofdbewerkingspunten heeft opgeleverd. De punten die zich achter de zwarte lijn bevinden in Figuur 14 zijn kopakkerpunten. Het is in de Figuur 13 wel duidelijk zichtbaar dat deze methode geen kopakkerpunten binnen de hoofdbewerking oplevert. Deze methode is dus een robuuste methode om een gedeelte van de kopakkerpunten te vinden, maar niet afdoende om al deze punten te vinden. Dit is veroorzaakt door metingen op de kopakker waar de trekker zich in dezelfde richting voortbewoog. Deze zijn door op  $\Delta\theta$  te selecteren niet gevonden. Visueel door de onderstaande grafiek voor alle percelen te plotten is gekeken in hoeverre deze methode betrouwbaar was voor de zes testpercelen. In bijlage A3 staan deze figuren.



**Figuur 13** Punten geselecteerd als behorende tot kopakker met methode A dus op basis  $\Delta\theta$  voor perceel 1



**Figuur 14** Punten achter de zwarte lijn bij de pijl behoren tot kopakker maar zijn ingedeeld bij de hoofdbewerking op basis van  $\Delta\theta$ .

In Tabel 2 is het aantal foutieve metingen voor de selectie van kopakkerpunten met behulp van  $\Delta\theta$  te zien. Door de plot in het XY vlak te controleren is te zien hoeveel metingen bij elk perceel binnen de hoofdbewerking liggen welke tot hoofdbewerking behoren. Daarnaast is gekeken hoeveel kopakkerpunten niet opgenomen zijn als kopakker.

**Tabel 2** Aantal foutieve metingen voor de kopakker op het totaal aantal punten

| Perceel | Totaal aantal punten | Fouten kopakkerpunten | %    | Niet gevonden kopakkerpunten | %  |
|---------|----------------------|-----------------------|------|------------------------------|----|
| 1       | 4168                 | 0                     | 0.00 | 452                          | 80 |
| 2       | 13341                | 7                     | 0.05 | 600                          | 51 |
| 3       | 9215                 | 0                     | 0.00 | 373                          | 44 |
| 4       | 8239                 | 3                     | 0.04 | 970                          | 76 |
| 5       | 11024                | 4                     | 0.04 | 920                          | 72 |
| 6       | 16007                | 3                     | 0.02 | 1410                         | 52 |

Uit bovenstaande tabel blijkt duidelijk dat de met methode A gevonden kopakkerpunten nagenoeg voor 100% tot kopakker behoren. Er blijkt ook dat doorgaans meer dan 50% van de kopakkerpunten niet gevonden wordt.

## 4.2 Scheiden hoofdbewerking en kopakker m.b.v. Kmeans (methode B)

In Labview is allereerst het gemiddelde berekend voor  $\Delta\theta$ , V en B voor alle punten binnen polygoon K. Dit is ook gebeurd voor alle kopakkerpunten gevonden op basis van  $\Delta\theta$ . Deze gemiddelden staan voor alle percelen in Tabel 3.

**Tabel 3 Gemiddelde en standaardafwijking parameters  $\Delta\theta$ , B en V Kmeans voor de kopakkerpunten (K) en de hoofdbewerkingspunten (H) van alle zes testpercelen**

| Perceel No. | Perceels Vorm | $\Delta\theta$ -H<br>mean(st.<br>dev) | B-H<br>mean(st.<br>dev) | V-H<br>mean(st.<br>dev) | $\Delta\theta$ -K<br>mean(st.<br>dev) | B-K<br>mean(st.<br>dev) | V-K<br>mean(st.<br>dev) | Datapunt<br>en<br>perceel |
|-------------|---------------|---------------------------------------|-------------------------|-------------------------|---------------------------------------|-------------------------|-------------------------|---------------------------|
| 1           | rechthoek     | 0.00(0.42)                            | 20.04(3.78)             | 2.07(0.18)              | 0.01(2.67)                            | 19.37(6.79)             | 1.79(0.44)              | 4168                      |
| 2           | rechthoek     | 0.00(0.41)                            | 24.27(3.26)             | 1.51(0.08)              | 0.00(2.65)                            | 21.37(7.33)             | 1.39(0.38)              | 13341                     |
| 3           | rechthoek     | 0.00(0.47)                            | 21.16(3.10)             | 2.11(0.05)              | 0.00(2.62)                            | 18.29(7.05)             | 1.78(0.62)              | 9215                      |
| 4           | Rechthoek+*   | 0.00(0.47)                            | 25.00(3.95)             | 1.46(0.11)              | 0.00(2.49)                            | 21.86(7.97)             | 1.31(0.46)              | 8239                      |
| 5           | Rechthoek+*   | 0.00(0.35)                            | 25.96(3.46)             | 1.25(0.21)              | 0.00(2.64)                            | 24.55(6.00)             | 1.24(0.32)              | 11024                     |
| 6           | Driehoek      | 0.00(0.48)                            | 19.04(3.24)             | 2.07(0.04)              | 0.00(2.62)                            | 15.07(7.23)             | 1.64(0.72)              | 16007                     |

In deze tabel is duidelijk te zien dat de standaardafwijking voor de gemiddelden van de kopakker erg groot is. Soms zelfs groter dan het gemiddelde. Dit is logisch verklaarbaar, op de kopakker rijdt de trekker voor en achteruit, staat stil, er wordt op het gaspedaal gereden en niet bij een constant toerental. Dit zorgt voor grote verschillen in snelheid, brandstofverbruik en rijrichting. Er is gekeken hoeveel punten die bij methode A correct tot kopakker gerekend werden door Kmeans omgezet zijn naar hoofdbewerking. In Tabel 4 is dit weergegeven.

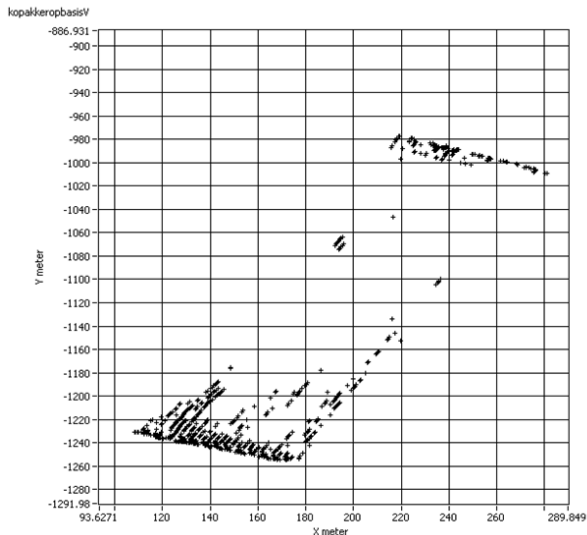
**Tabel 4 Aantal foutieve metingen voor de kopakke rop het totaal aantal punten**

| Perceel | kopakkerpunten methode A | Veranderd in hoofdbewerking door methode B | Percentage |
|---------|--------------------------|--------------------------------------------|------------|
| 1       | 120                      | 118                                        | 98         |
| 2       | 585                      | 480                                        | 82         |
| 3       | 475                      | 383                                        | 80         |
| 4       | 299                      | 247                                        | 82         |
| 5       | 351                      | 286                                        | 81         |
| 6       | 1327                     | 1207                                       | 91         |

Het aantal foutief veranderde kopakkerpunten in vergelijking met methode A, bleek voor methode B voor alle percelen meer dan 80% te zijn.

### 4.3 Scheiden hoofdbewerking en kopakker m.b.v. $\Delta\theta$ en V (methode C)

Naast het onderscheiden op basis van  $\Delta\theta$  is dit ook op basis van de snelheid V in elk punt van het bestand geprobeerd. Zoals in Figuur 15 duidelijk te zien is, levert dit teveel foutief ingedeelde kopakkerpunten op. Dit is veroorzaakt door hoofdbewerkingspunten waar de snelheid de gestelde grenswaarde overschrijdt. Omdat de snelheid toch een bruikbare parameter blijft, is deze toch gebruikt.

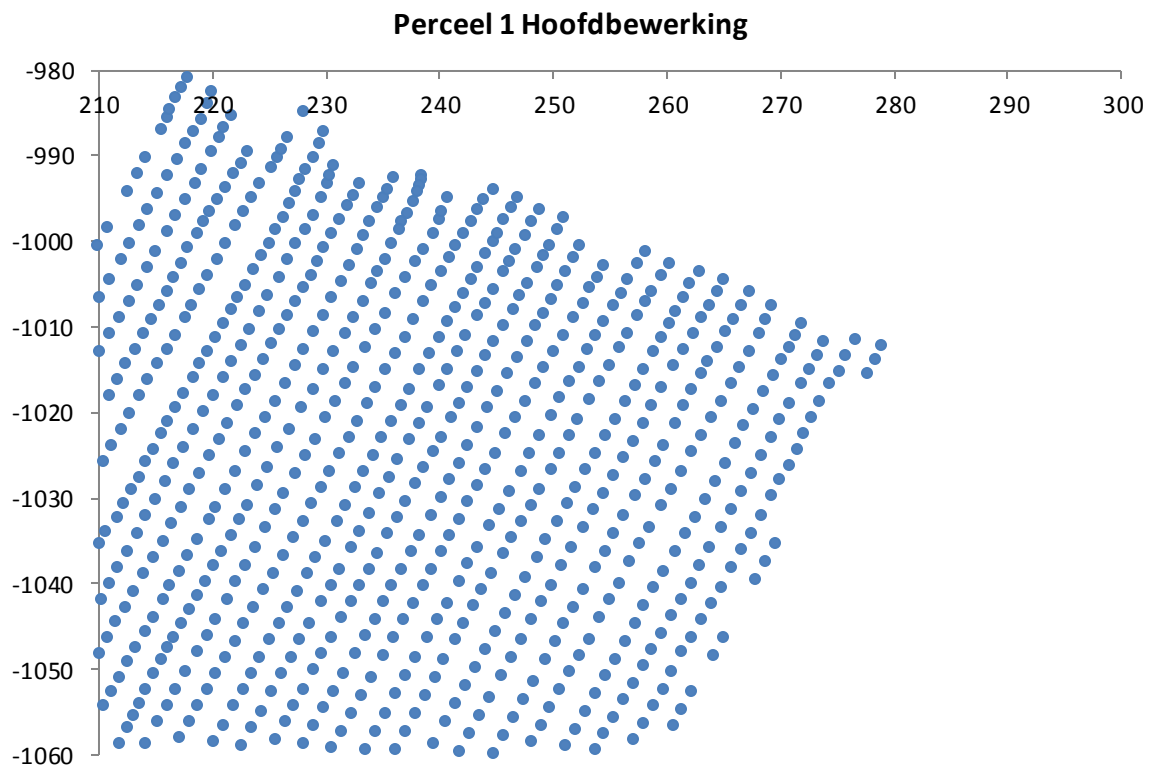


**Figuur 15 Kopakkerpunten Perceel 1 op basis Vhoofdbewerking gemiddeld**

Door het trekken van een cirkel, met een straal van drie meter, om elk kopakkerpunt gevonden met methode A en vervolgens binnen deze cirkel selecteren op snelheid, is in methode C de kopakker gescheiden van de hoofdbewerking. Dit gebeurt geheel in Labview en hier zijn ook plot's gemaakt van het resultaat, in het XY vlak. Dit is gebeurd voor alle zes percelen. Daaruit blijkt dat deze methode aanzienlijk meer punten vindt dan  $\Delta\theta$  alleen. Deze dubbele selectie levert minder foutieve kopakkerpunten in de hoofdbewerking op. Voor perceel 1 is dit weergegeven in Figuur 16 tot en met Figuur 18. De XY plot van alle zes percelen is te zien in bijlage A4.

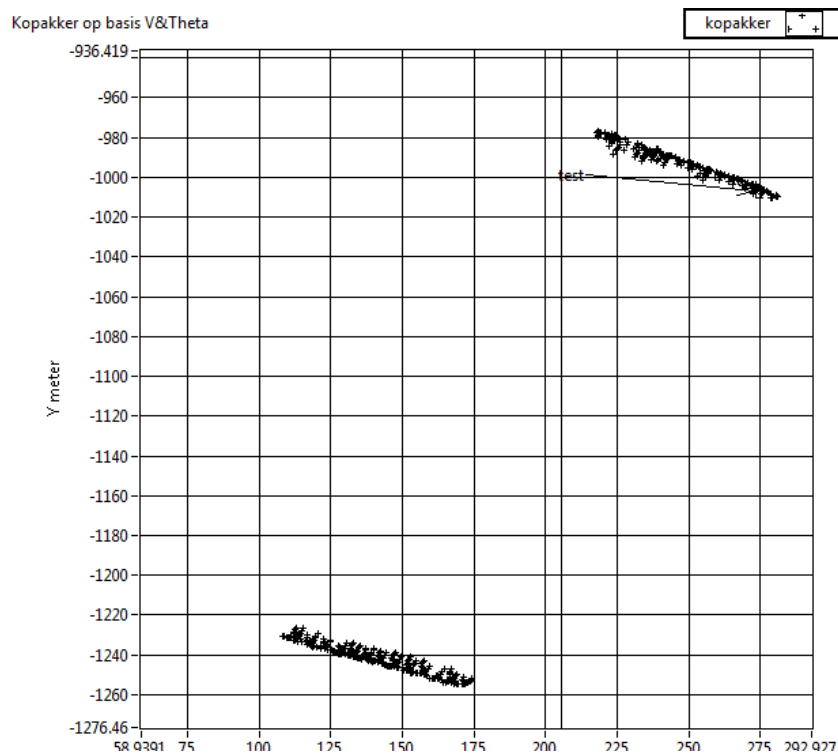
**Tabel 5 Resultaten method C**

| Perceel | Totaal aantal punten | Fouten kopakkerpunten | %    | Niet gevonden kopakkerpunten | %   |
|---------|----------------------|-----------------------|------|------------------------------|-----|
| 1       | 4168                 | 0                     | 0.00 | 30                           | 0.7 |
| 2       | 13341                | 18                    | 0.01 | 39                           | 0.3 |
| 3       | 9215                 | 0                     | 0.00 | 28                           | 0.3 |
| 4       | 8239                 | 15                    | 0.20 | 50                           | 0.6 |
| 5       | 11024                | 12                    | 0.10 | 65                           | 0.6 |
| 6       | 16007                | 8                     | 0.05 | 70                           | 0.4 |

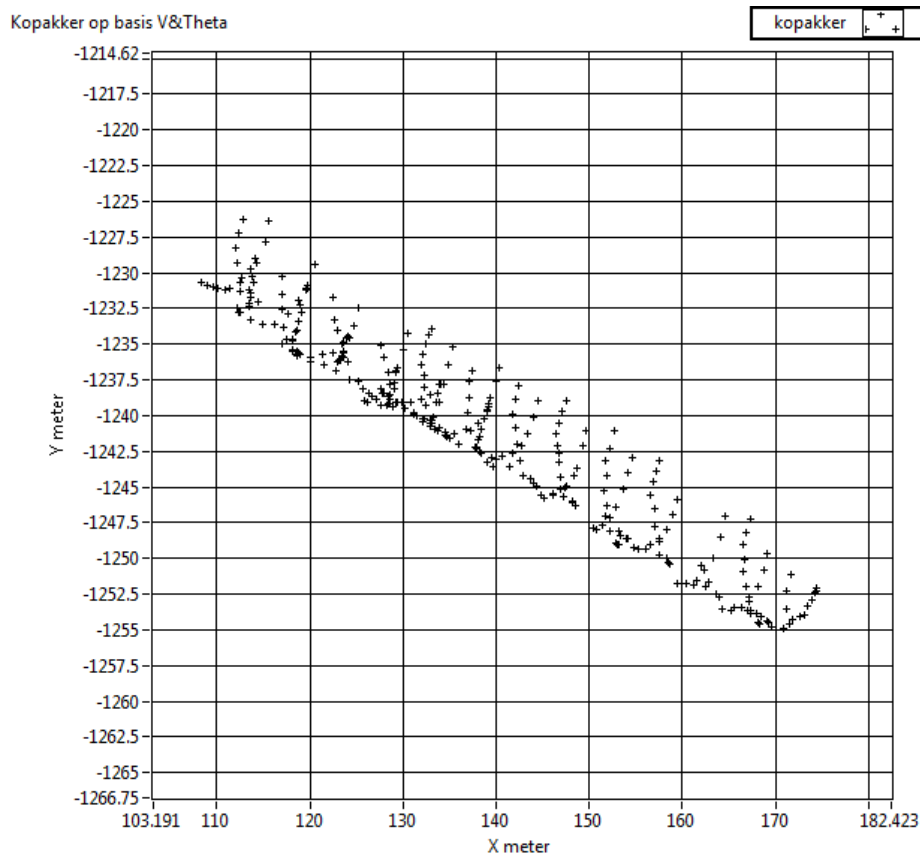


**Figuur 16** Hoofdbewerkingspunten perceel 1 ingedeeld op basis van methode C





**Figuur 17 Kopakkerpunten perceel 1 ingedeeld op basis van Methode C**



**Figuur 18 Uitvergrote weergave gedeelte geselecteerde kopakkerpunten perceel 1 op basis van methode C. Duidelijk zichtbaar dat nu de bochten op de kopakker meegenomen worden in de selectie voor kopakker.**

#### 4.4 Vergelijken methode A, B en C kopakkerselectie

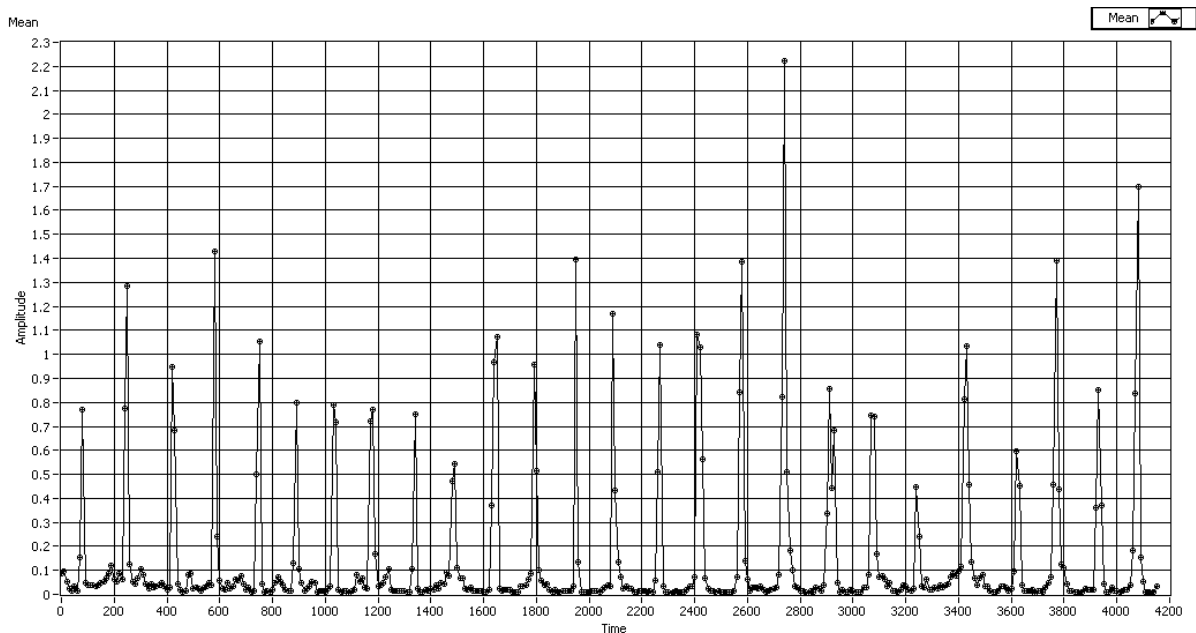
In Tabel 6 zijn de resultaten van de drie methoden voor kopakkerselectie voor perceel 1 naast elkaar gezet. Hierin is te zien dat de methode A, berustend op de richtingsverandering, het minst aantal punten oplevert voor elk perceel. Deze punten blijken wanneer ze in het XY vlak geplot worden echter wel voor zo goed als 100% op de kopakker te liggen. De punten hoofdbewerking zijn die punten die binnen polygoon K liggen. Punten die niet geselecteerd zijn als kopakker of binnen polygoon K liggen zijn onbekend genoemd. Methode B resulteert in een groter aantal kopakkerpunten. Wanneer de kopakkerpunten van methode A met die van methode B vergeleken worden blijken deze voor een groot gedeelte in hoofdbewerking veranderd te zijn zoals in Tabel 6 te zien is. Dit is veroorzaakt door het te kleine verschil tussen de gemiddelden voor de gebruikte parameters en de te grote standaardafwijking van de gemiddelden voor de kopakker. Dus hoewel het aantal gevonden kopakkerpunten het hoogst is zijn deze voor het grootste gedeelte fout ingedeeld. Methode C levert naast de punten van de eerste methode een behoorlijke aanvulling hierop. Bij plotten van deze punten blijkt dat deze correct zijn ingedeeld en deze methode dus het beste werkt. De resultaten voor de overige percelen onderstrepen dit resultaat. Deze zijn terug te vinden in bijlage A6.

**Tabel 6 Resultaten indeling datapunten Perceel 1, kolom 1: 0=onbekend, 1=kopakker, 2=hoofdbewerking**

|                | Keuze Methode A ( $\Delta\theta$ ) | Keuze Methode B(Kmeans) | Keuze Methode C (Cirkel) |
|----------------|------------------------------------|-------------------------|--------------------------|
| Onbekend       | 3042                               | 0                       | 0                        |
| Kopakker       | 120                                | 739                     | 572                      |
| Hoofdbewerking | 1004                               | 3427                    | 3594                     |
| Totaal         | 4166                               | 4166                    | 4166                     |

#### 4.5 Bepalen keermomenten op basis $\Delta\theta$

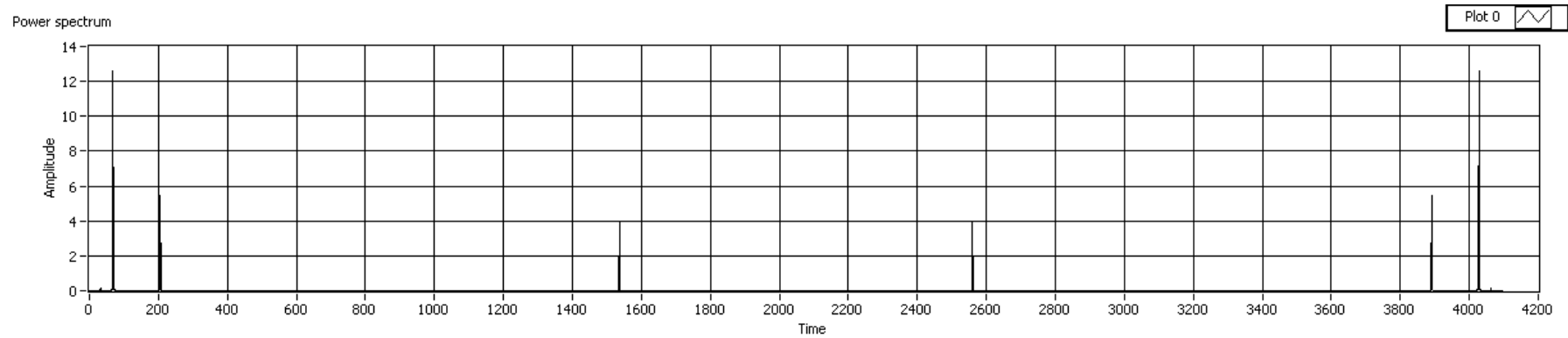
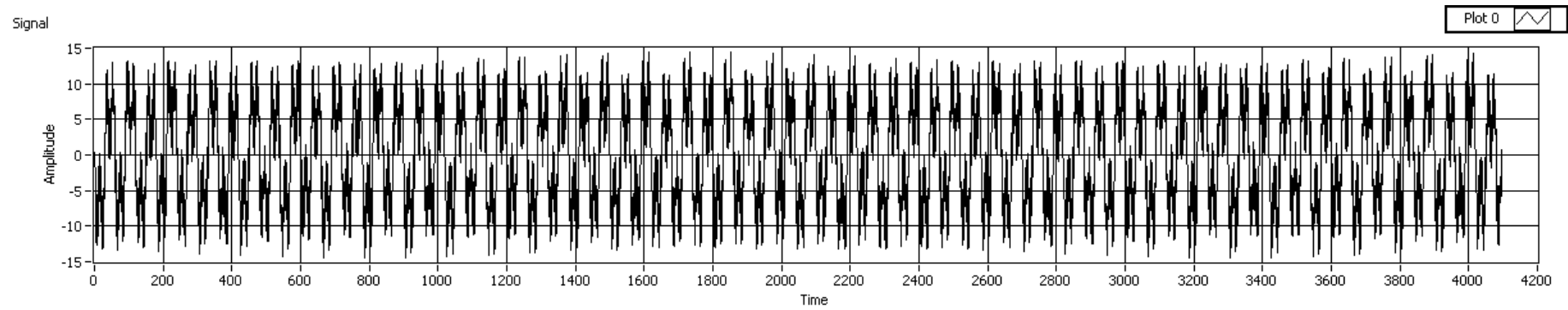
In Figuur 19 geeft elke stip een piek aan gevonden door de peakdetector van Labview. In de figuur bevinden zich 27 duidelijk waarneembare pieken, dit stemt overeen met het aantal keermomenten van perceel 1. Tussen deze pieken blijft  $\Delta\theta$  vrij constant onder een waarde van 0.1. Dit zijn de momenten waarop de hoofdbewerking uitgevoerd is. Wanneer de pieken van dichterbij bekeken worden blijkt dat  $\Delta\theta$  binnen een piek, kleine pieken laat zien welke door de peak detector ook gevonden zijn. Hierdoor stemt het totaal gevonden aantal pieken niet overeen met het aantal keermomenten van perceel 1.



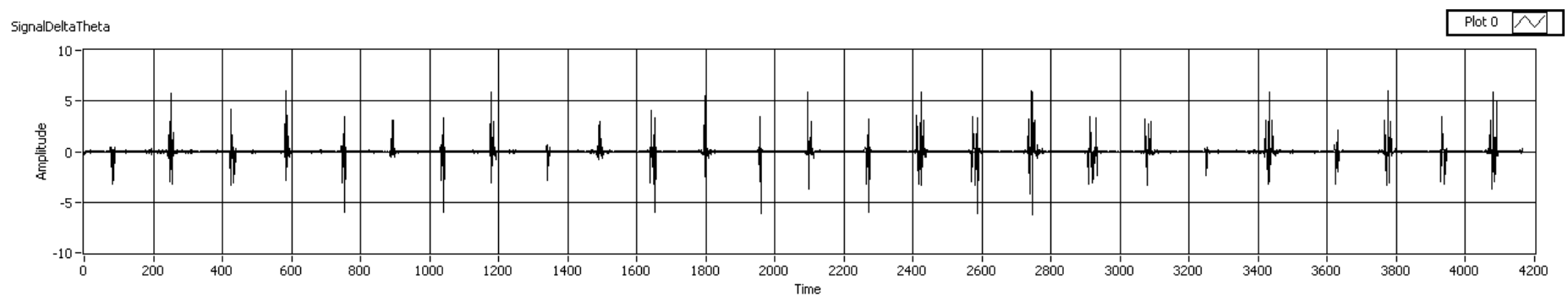
**Figuur 19 Ouput Peak Detection**

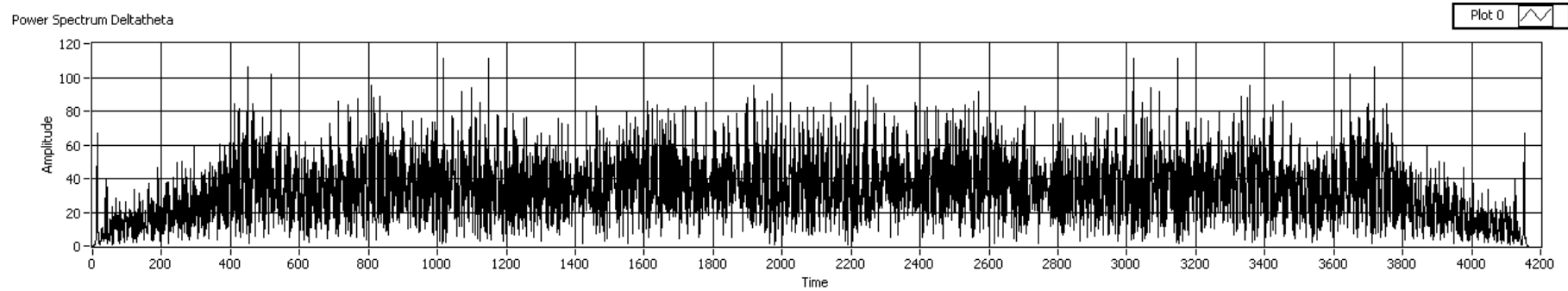
De in Figuur 19 weergegeven resultaten waren voor een handmatig geselecteerde data compressie waarde van 10x. Omdat vaste zelf gekozen waarden gebruiken voor de datacompressie de robuustheid van het algoritme vermindert is gezocht naar alternatieven.

Om het aantal werkgangen-keermomenten vast te stellen is gekeken naar de mogelijkheid een low pass filter te implementeren. Hiertoe moest gekeken worden of  $\Delta\theta$  dit toelaat. Er is een Fast Fourier Transformatie uitgevoerd voor  $\Delta\theta$ , dit om vast te stellen of er voor de pieken op de keermomenten en de richting tijdens de hoofdbewerking een duidelijk onderscheid in frequentie te maken zou zijn tussen deze twee. Een som van drie normale sinussen met verschillende frequentie en amplitude resulteert in Figuur 20. Hierin zijn duidelijk drie pieken te zien welke elk een van de sinussen weergeeft. Deze drie pieken zijn in het powerspectrum gespiegeld. Figuur 21 is het resulterende powerspectrum voor  $\Delta\theta$ . In dit figuur is duidelijk zichtbaar dat er geen duidelijk waarneembare pieken zijn. Dit betekent dat er te veel verschillende sinussen de pieken van  $\Delta\theta$  vormen. Hierdoor is het powerspectrum niet bruikbaar om het aantal pieken en daarmee het aantal keermomenten vast te stellen. De beide beproefde methoden om het aantal werkgangen vast te stellen bleken niet robuust.



**Figuur 20** Signaal en powerspectrum van 3 sinussen met verschillende frequentie en amplitude





**Figuur 21 Signaal en powerspectrum van  $\Delta\theta$**

## 4.6 Resulterende Kengetallen

Methode C bleek de beste resultaten te geven als het gaat om het onderscheiden van kopakker en hoofdbewerkingspunten. De in de onderzoeksvraag gestelde kengetallen voor het brandstofverbruik in l/h op de kopakker en de hoofdbewerking zijn berekend uit de resultaten die deze methode oplevert. In Tabel 7 zijn deze terug te vinden voor de zes gebruikte percelen. Hierin is een duidelijk verschil te zien tussen hoofdbewerking en kopakker. Bij het eerste perceel is de bewerking woelen, de overige vijf was dit ploegen. Zoals uit de getallen blijkt zit er ook een verschil in verbruik per perceel. Dit kan aan een grote hoeveelheid factoren liggen.

**Tabel 7 Gemiddeld brandstofverbruik (standaardafwijking) en het aantal datapunten voor kopakker en hoofdbewerking op de zes test percelen**

| Perceel                     | Perceel 1 | Perceel 2  | Perceel 3 | Perceel 4 | Perceel 5  | Perceel 6 |
|-----------------------------|-----------|------------|-----------|-----------|------------|-----------|
| Kopakkerpunten              | 572       | 1185       | 848       | 1269      | 1271       | 2737      |
| Verbruik kopakker l/h       | 7.5(6.6)  | 7.8(8.5)   | 7.3(7.6)  | 12.6(9.7) | 15.6(10.0) | 7.6(6.2)  |
| Hoofdbewerkingspunten       | 3594      | 12154      | 8365      | 6968      | 9751       | 13268     |
| Verbruik hoofdbewerking l/h | 20.8(5.3) | 22.17(6.4) | 19.0(6.4) | 23.0(6.8) | 25.3(4.7)  | 17.0(5.3) |

## 5 Discussie

### 5.1 Wat is de beste onderscheidingsmethode?

Methode A bleek nagenoeg 100% betrouwbaar wat betreft gevonden kopakkerpunten. Dit wil zeggen er werden geen punten behorende tot hoofdbewerking aan kopakker toegekend. Van alle kopakkerpunten is echter meestal meer dan 50% en soms zelfs 80 procent niet gevonden.

Met behulp van Kmeans is vervolgens gekeken naar de mogelijkheid om alle drie de parameters  $\Delta\theta$ , V en B, te gebruiken voor het onderscheiden van kopakker en hoofdbewerking. De gemiddelden van de drie parameters bleken voor de met methode A gevonden kopakkerpunten echter een te grote standaardafwijking te hebben. Hierdoor is het niet mogelijk met Kmeans een betrouwbaar onderscheid te maken tussen hoofdbewerking en kopakker. Dit doordat er teveel punten foutief ingedeeld werden. Dit kwam ook naar voren wanneer de output van Kmeans met die van methode A vergeleken werd. Het was reeds duidelijk dat hierbij de kopakkerpunten nagenoeg voor 100% correct waren ingedeeld. Kmeans deelt hiervan voor alle percelen meer dan 80% verkeerd in.

Methode C selecteert meer punten van de hoofdbewerking dan methode A, terwijl er weinig nieuwe foutieve indelingen bijkomen. Geen van de punten gevonden met methode A is veranderd, er komen slechts punten bij die als kopakker geselecteerd zijn. Hierdoor komt methode C als meest bruikbare onderscheidingsmethode naar voren. Punt van discussie bij het vergelijken van de drie methoden is dat de resultaten steeds visueel gecontroleerd zijn. De selectie hoofdbewerking en kopakker is geplot in het XY vlak. Hier is vervolgens gekeken hoeveel punten er in hoofdbewerking resteerden die tot de kopakker behoorden. De afstand tot de perceelsgrens, dichtheid van de punten (indicatie

voor de snelheid) en de richting van opeenvolgende punten werden bij deze controle als parameters gebruikt. Er was geen informatie beschikbaar waar een betrouwbare indeling bekend was om te kunnen controleren in hoeverre het resultaat van het algoritme overeen kwam met de werkelijkheid. Een harde statistische onderbouwing van de resultaten ontbreekt.

## **5.2 Is de verzamelde data toereikend?**

In de gevonden literatuur bleek dat onderzoek naar plaatsspecifiek brandstofverbruik met behulp van GPS relatief nieuw is. Wat opviel bij eerdere onderzoeken was dat er naast de metingen van het brandstofverbruik bij grondbewerking ook metingen naar verschillende andere factoren uitgevoerd werden. Trekkkrachtmetingen, het moment op de aandrijving van de tractor etc. zijn parameters die gemeten werden. Hieruit werden betrouwbare conclusies getrokken aangaande het brandstofverbruik. In het onderzoek waarvoor dit algoritme ontwikkeld is, is slechts data verzameld over de positie en snelheid van de trekker en diens brandstofverbruik op dat moment. Dit uitbreiden met overige parameters lijkt een bruikbare en rendabele verbetering.

Daarnaast is het tijdens de ontwikkeling van het algoritme, voor het scheiden van kopakker en hoofdbewerking, duidelijk geworden dat het bijzonder lastig is hiervoor een robuuste methode te ontwikkelen. Dit wordt hoofdzakelijk veroorzaakt door de problemen bij het vinden van een doeltreffende maar betrouwbare methode om de kopakkerpunten van de hoofdbewerking te onderscheiden. Om dit te verbeteren is het waarschijnlijk eenvoudiger extra parameters aan de metingen toe te voegen dan een robuust algoritme te ontwikkelen. De meest directe methode om kopakker en hoofdbewerking te onderscheiden bij de uitgevoerde bewerkingen, is het in kaart brengen van de positie van de hefinrichting. Het indelen naar kopakker of hoofdbewerking op basis van een gegeven of in werking zijnde hefinrichting is een relatief eenvoudige en toch robuuste methode om de kopakkerpunten van de hoofdbewerkingspunten te scheiden. Het meten van deze hefpositie kan het gebruik van methode A, B en C om kopakkerpunten en hoofdbewerkingspunten te scheiden op basis van de behandelde criteria overbodig maken. Dit zal de robuustheid waarschijnlijk ten goede komen. Zoals uit de vergelijking van deze drie methoden blijkt zijn ze geen van allen dekkend voor alle kopakkerpunten. Kanttekening hierbij is wel dat dit slechts werkt voor bewerkingen waarbij de hef gebruikt wordt. Is dit niet het geval dan kan er niet gelogd worden en zal methode C de meest robuuste oplossing zijn voor het scheiden van kopakker en hoofdbewerkingsmetingen.

## **5.3 Wat is de beste methode om polygoon K op te stellen?**

Het opstellen van een Polygoon K welke een goede selectie van hoofdbewerkingspunten maakt dient ook nog te gebeuren. Op dit moment is slechts een oplossing aanwezig voor de zes geteste percelen. Het inkorten van vectoren vanuit het middelpunt naar de grenspolygoon is echter niet betrouwbaar. Hiervoor kan beter de in de literatuur gevonden methode offsetting danwel scaling gebruikt worden. Uit het literatuuronderzoek bleek dat scaling beduidend minder berekeningen behoeft. Om de rekentijd voor het algoritme

kort te houden gaat de voorkeur dan ook uit naar het implementeren van deze methode.

## 6 Conclusie

Om het brandstofverbruik op de kopakker en tijdens de hoofdbewerking op een perceel te berekenen moet bekend zijn waar de trekker zich bevindt ten tijde van een meting. Het is lastig een betrouwbaar algoritme te ontwikkelen om hoofdbewerking en kopakker te onderscheiden op een perceel wanneer uit de beschikbare metingen slechts de parameters tijd van de meting, locatie, snelheid en brandstofverbruik beschikbaar zijn.

Via de locatie en de tijd zijn de parameters snelheid en rijrichting te berekenen. De verandering van de rijrichting in combinatie met de snelheid zijn de meest bruikbare parameters om het onderscheid tussen kopakker en hoofdbewerking te maken. Dit omdat de verandering van de rijrichting kopakkerpunten oplevert welke voor bijna 100% op de kopakker liggen. Hierbij werd voor de testpercelen meer 50% van de kopakkerpunten echter niet gevonden. Door het toevoegen van een selectie op rijnsnelheid binnen een vaste afstand van de gevonden kopakkerpunten komt het percentage gevonden kopakkerpunten op meer dan 90%. Dit is de beste implementatie van de parameters in het algoritme die gevonden is.

De parameter brandstofverbruik vertoont op de kopakker en tijdens de hoofdbewerking te grote afwijkingen van het gemiddelde om bruikbaar te zijn. Dit is met behulp van Kmeans getest waarbij de parameter brandstofverbruik in combinatie met rijrichting en snelheid gebruikt is. Deze methode veranderde meer dan 80% van de correct, met rijrichtingsverandering gevonden kopakkerpunten, foutief in hoofdbewerking.

Om het aantal keermomenten vast te stellen is de parameter rijrichtingsverandering getest. Deze laat duidelijke pieken zien op de keermomenten en lijkt dan ook bruikbaar. De extreme waarden die de rijrichtingsverandering op keermomenten vertoont zijn verschillend voor elk keermoment. Het vaststellen van de keermomenten met deze parameter is niet mogelijk met een peak detector of low pass filter.



## 7 Aanbevelingen

In het bestaande algoritme, ontwikkeld door Nieuwenhuizen en verwerkt in Main Processing VI in Labview, dienen de subVI's BoundaryPolygon en PolygonSelections verbeterd te worden. Op dit moment wordt het transport naar het perceel handmatig verwijderd uit de oorspronkelijke meting. Een methode ontwikkelen om deze transportmetingen achteraf los te koppelen van de bewerkingen op een perceel is lastig omdat de perceelsgrenzen de eerste keer dat een perceel bewerkt wordt nog niet vaststaan. Het eenmalig vaststellen van de perceelsgrenzen lijkt een goede alternatief voor het ontwikkelen van een algoritme dat kan berekenen wanneer de trekker op een perceel bezig is. Dit zou apart uitgevoerd kunnen worden of de chauffeur zou eenmalig aan moeten geven wanneer hij op het perceel aangekomen is en wanneer hij het verlaat. Dan is duidelijk welk deel van het bestand daadwerkelijk metingen op het perceel bevat.

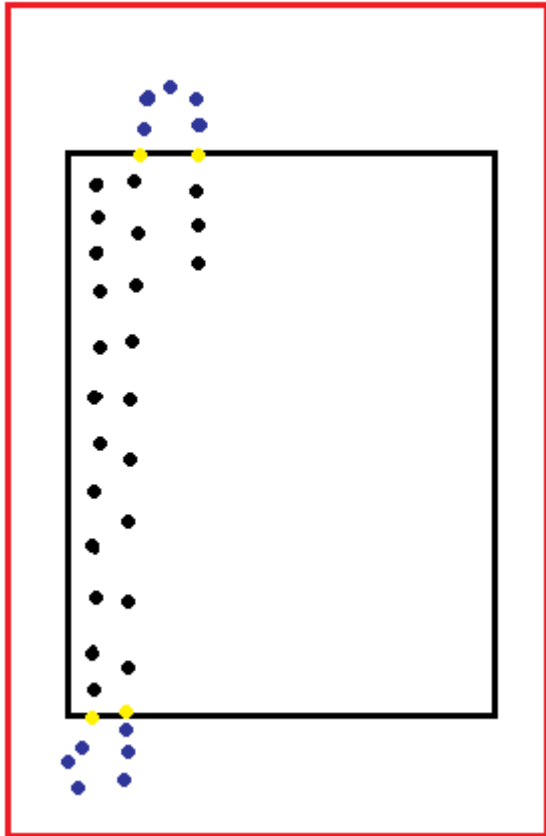
Naast het loggen van de aankomst op het perceel is het nuttig een sensor te installeren welke de positie van de hef inrichting meet. Dit hoeft slechts een eenvoudige sensor te zijn met een 1-0 output. Met behulp van de gegevens over de hef is het mogelijk de kopakker en hoofdbewerkingspunten goed en duidelijk te scheiden. Dan dient het benodigde algoritme nog slechts om foutieve metingen op het perceel te verwijderen.

Het ontwikkelde algoritme in LabView bestaat op dit moment uit aan elkaar gekoppelde onderdelen om alle gebruikte methoden te berekenen. Het is praktisch deze in aparte SubVI's onder te brengen om zodoende te kunnen kiezen welke bouwstenen gebruikt worden en welke niet. Alle berekeningen gebruiken dezelfde meetgegevens. Hierdoor is het relatief eenvoudig de SubVI-Dre op te splitsen in losse onderdelen.

In de For loop waarin de cirkels getekend worden en de punten op basis van V geselecteerd worden is de SubVI PolygonDataSelector verwerkt. Deze moet voor elk kopakkerpunt gevonden op basis van  $\Delta\theta$  om de data binnen de cirkel om dit punt selecteren. Dit blijkt zelfs wanneer de interpolatie in de SubVI uitgeschakeld is nog steeds een berekening te zijn die veel rekenvermogen vraagt. Hier kan waarschijnlijk efficiënter geprogrammeerd worden. Een ander punt van aandacht zijn de meetfouten uit methode A. Dit zijn er ten opzichte van het totaal minder dan 1% maar dit neemt toe wanneer in een cirkel rond deze punten op V geselecteerd wordt in methode C.

De nog te testen alternatieve methode om het aantal keermomenten vast te stellen berust op het volgende principe. Er wordt een polygoon om alle hoofdbewerkingspunten, gevonden met methode C, opgesteld. Vervolgens worden in chronologische volgorde de XY coördinaten van het gehele perceel doorlopen. Wanneer deze voor het eerst overeenkomen maakt de trekker voor het eerst de overgang van kopakker naar hoofdbewerking. De volgende overeenkomende getallen geven het einde van de eerste werkgang aan. Hierna volgt het terug naar hoofdbewerking gaan, einde werkgang enz. In Figuur 22 is dit weergegeven. Door het totaal aantal keren dat de polygoon welke

hoofdbewerking omsluit gepasseerd wordt vast te stellen, kan het aantal keermomenten op de kopakker in kaart gebracht worden. Hierbij is wel een vereiste dat de kopakker niet aan het einde ook bewerkt wordt waarbij er een groot aantal overeenkomsten ontstaat wanneer over de grens tussen kopakker en hoofdbewerking gereden wordt.

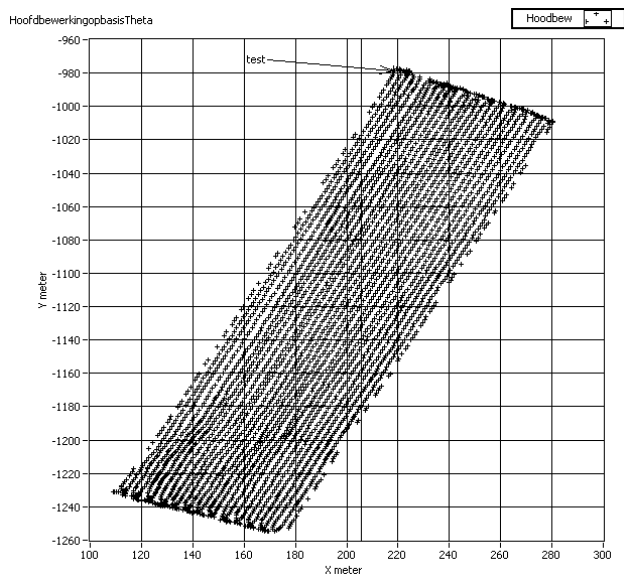


**Figuur 22** Passeren van polygoon welke hoofdbewerking omvat (zwart) aangegeven met de gele metingen. Hierna bevindt de trekker zich op de kopakker aangegeven met de blauwe metingen. Voordat de trekker verder gaat met de hoofdbewerking passeert de trekker nogmaals de polygoon die de hoofdbewerking omsluit.

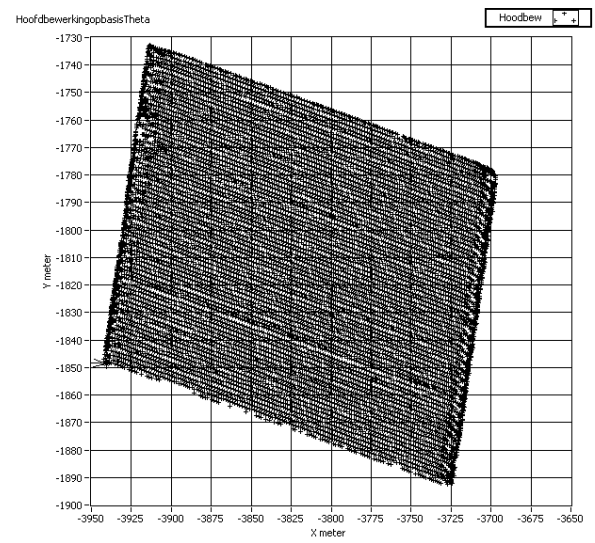
## 8 Referenties

- Kanungo, T., D. M. Mount, et al. (2000). Analysis of a simple k-means clustering algorithm. Proceedings of the Annual Symposium on Computational Geometry.
- Kanungo, T., D. M. Mount, et al. (2002). "An efficient k-means clustering algorithms: Analysis and implementation." IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7): 881-892.
- Kichler, C. M., J. P. Fulton, et al. (2007). Spatially monitoring tractor performance to evaluate energy requirements of variable depth tillage and implement selection. 2007 ASABE Annual International Meeting, Technical Papers.
- McLaughlin, N. B., L. C. Heslop, et al. (1993). "A General Purpose Tractor Instrumentation and Data Logging System." **36**(2): 265-273.
- Tompkins, F. D. and R. G. Carpenter (1978). "FUEL CONSUMPTION MONITORING FOR FIELD POWER UNITS." Paper - American Society of Agricultural Engineers.
- Wein, R. (2007). "Exact and approximate construction of offset polygons." CAD Computer Aided Design **39**(6): 518-527.
- Wiedemann, H. T., L. E. Clark, et al. (1986). IMPLEMENT POWER REQUIREMENTS FOR REDUCED TILLAGE. Paper - American Society of Agricultural Engineers.
- Yahya, A., M. Zohadie, et al. (2009). "Mapping system for tractor-implement performance." Computers and Electronics in Agriculture **69**(1): 2-11.
- Yule, I. J., G. Kohnen, et al. (1999). "A tractor performance monitor with DGPS capability." Computers and Electronics in Agriculture **23**(2): 155-174.

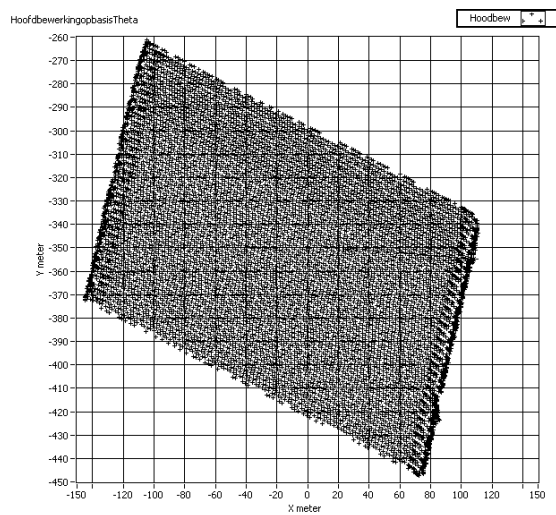
# A1 Overzicht percelen



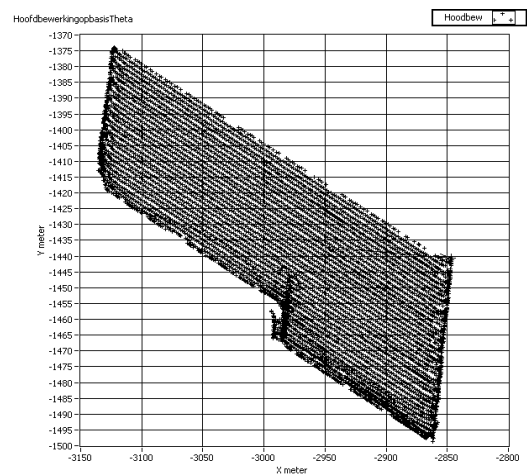
**Figuur 23 Perceel 1**



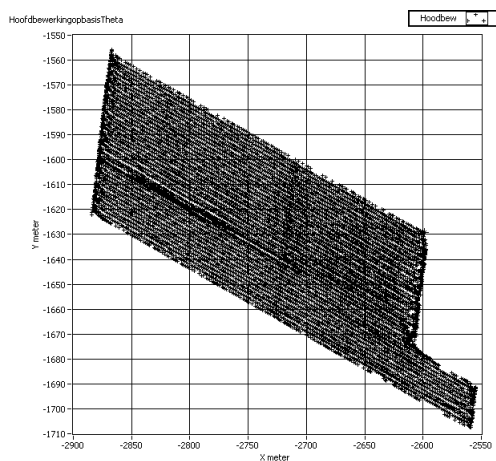
**Figuur 24 Perceel 2**



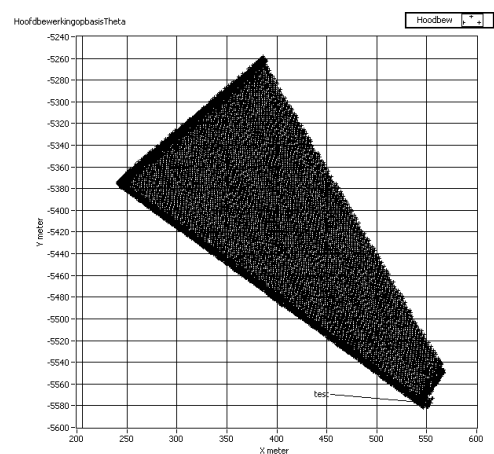
**Figuur 25 Perceel 3**



**Figuur 26 Perceel 4**



**Figuur 27 Perceel 5**



**Figuur 28 Perceel 6**

## A2 Opbouw Bestanden

Het bestand zoals het aangemaakt wordt door de GPS datalogger en de brandstofmeter.

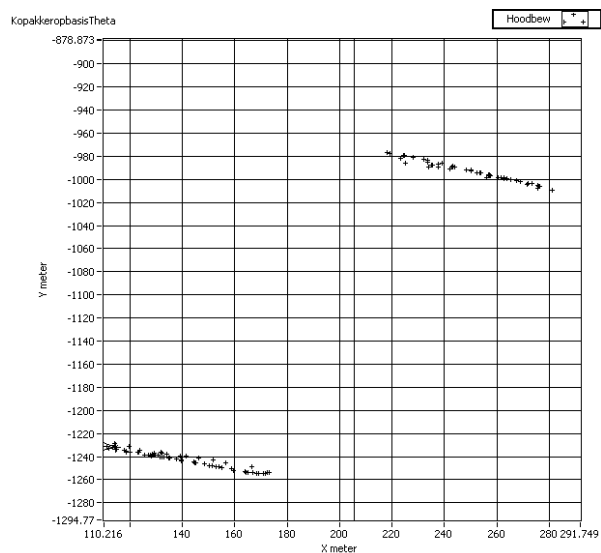
De kolommen van een bestand bevatten de volgende informatie van links naar rechts:

- 1: Tijd (UTC) deze kolom bevat de tijd van de opeenvolgende metingen.
- 2: Latitude. WGS 84
- 3: Longitude. WGS 84
- 4: AntAlt: hoogte GPS ontvanger op tractor boven NAP
- 5: GeoidSep: hoogteverschil Geoid (NAP) boven WGS84 elips
- 6: Speed: door het GPS systeem gemeten snelheid in km/h
- 7: CumPulses: aantal metingen uitgevoerd door de flow meter
- 8: Liter/Hr: brandstofverbruik op gegeven tijdstip in liters per uur
- 9: X-coördinaat in meters
- 10: Y-coördinaat in meters
- 11: Afgelegde weg in meters
- 12: Measurement error 0 betekent geen fout, 1 is meetfout

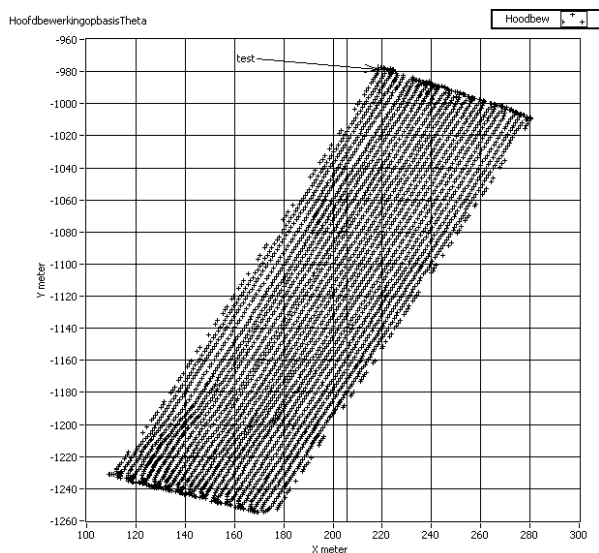
Tabel 8 opbouw bestand

| UTC   | LAT    | LON   | AntAlt | Geoid Sep | Speed (km/h) | Cum Pulses | Liter/ Hr | X meters | Y meters     | Distance | Error Code |
|-------|--------|-------|--------|-----------|--------------|------------|-----------|----------|--------------|----------|------------|
| 10012 | 51.982 | 5.644 | 6.7    | 47.17     | 7            | 23852      | 0         | 177.158  | -<br>1255.36 | 1.537003 | 0          |
| 10013 | 51.983 | 5.644 | 6.73   | 47.17     | 4.5          | 23854      | 4.5       | 177.177  | -<br>1254.87 | 1.362616 | 0          |
| 10013 | 51.94  | 5.644 | 6.66   | 47.17     | 3.5          | 23854      | 9         | 177.215  | -<br>1252.82 | 1.179886 | 0          |
| 10013 | 51.981 | 5.644 | 6.69   | 47.17     | 4.2          | 23855      | 0         | 177.716  | -<br>1251.57 | 1.344805 | 0          |
| 10013 | 51.983 | 5.644 | 6.71   | 47.17     | 3.3          | 23856      | 4.5       | 178.198  | -<br>1250.38 | 1.284771 | 0          |

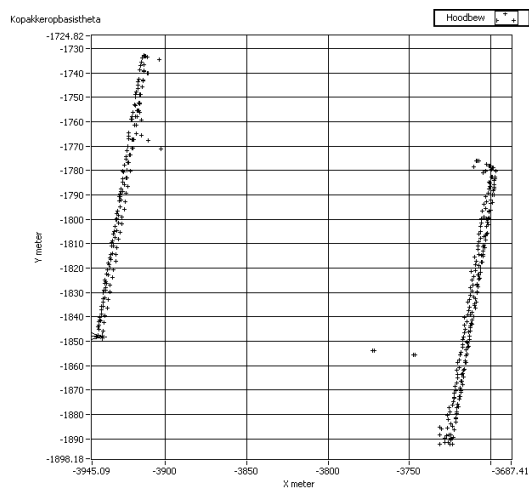
### **A3 Resultaten kopakkerselectie op basis $\Delta\theta$**



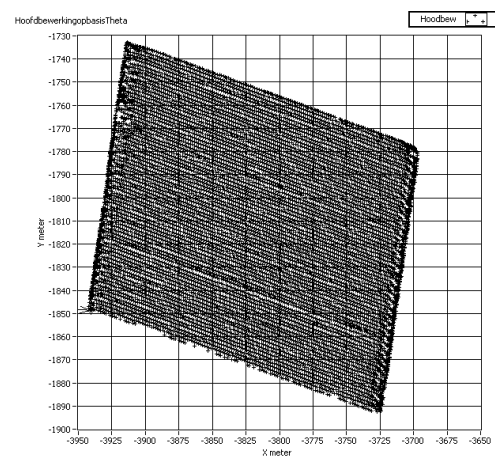
**Figuur 29 Kopakkerpunten Perceel 1 op basis  $\Delta 0$**



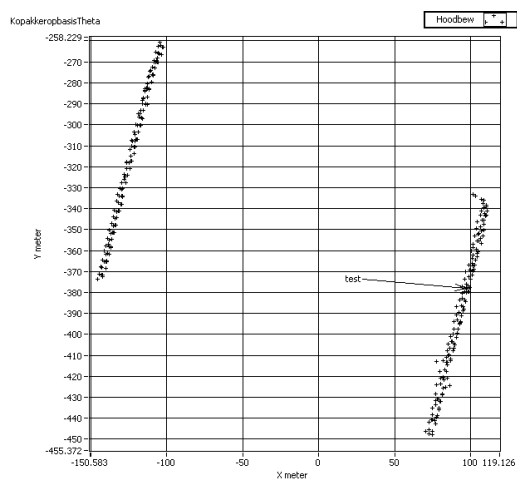
**Selectie Hoofdbewerkingpunten**



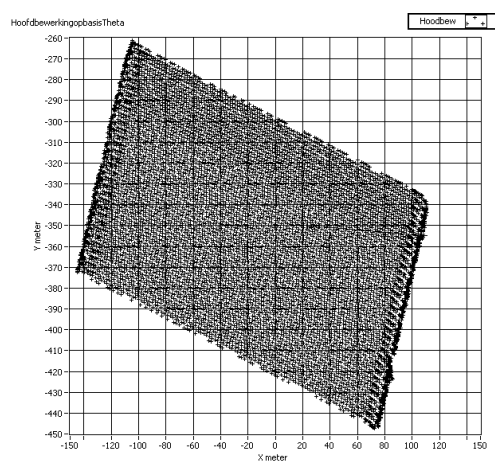
**Figuur 30 Kopakkerpunten Perceel 2 op basis  $\Delta 0$**



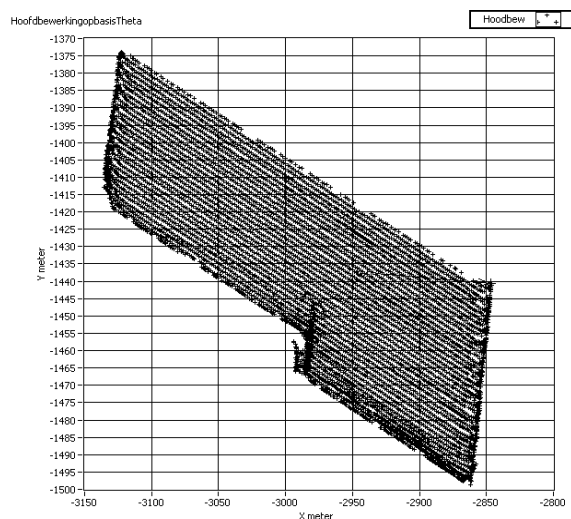
**Selectie Hoofdbewerkingpunten**



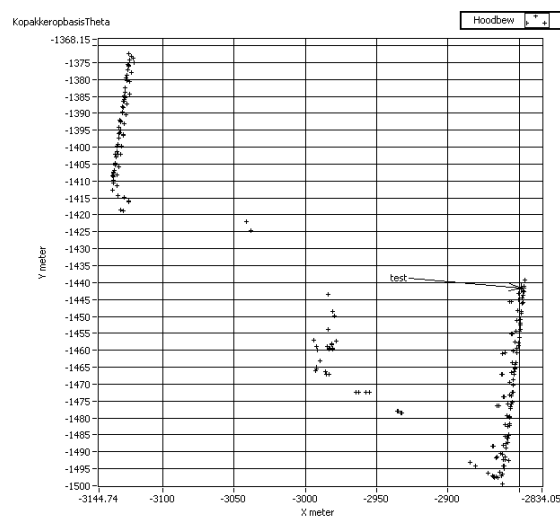
**Figuur 31 Kopakkerpunten Perceel 3 op basis  $\Delta 0$**



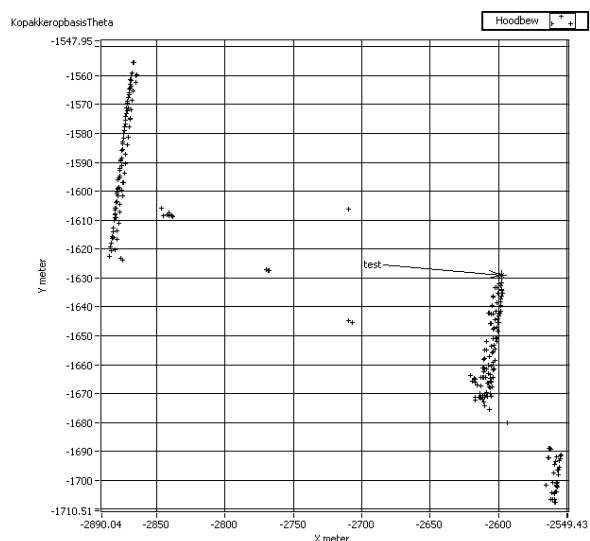
**Selectie Hoofdbewerkingpunten**



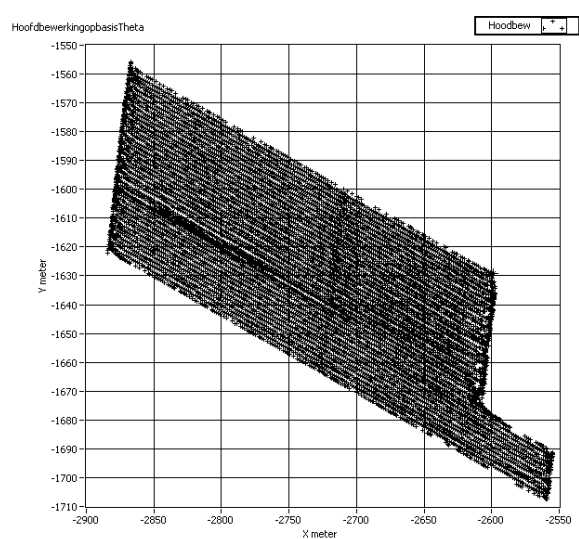
**Figuur 32 Kopakkerpunten Perceel 4 op basis  $\Delta 0$**



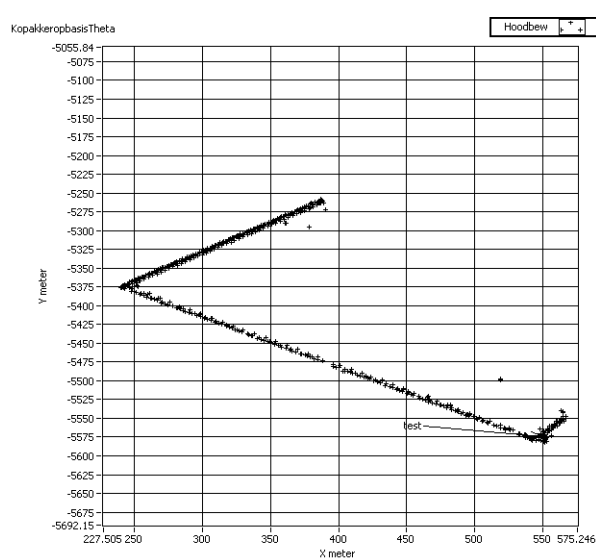
**Selectie Hoofdbewerkingspunten**



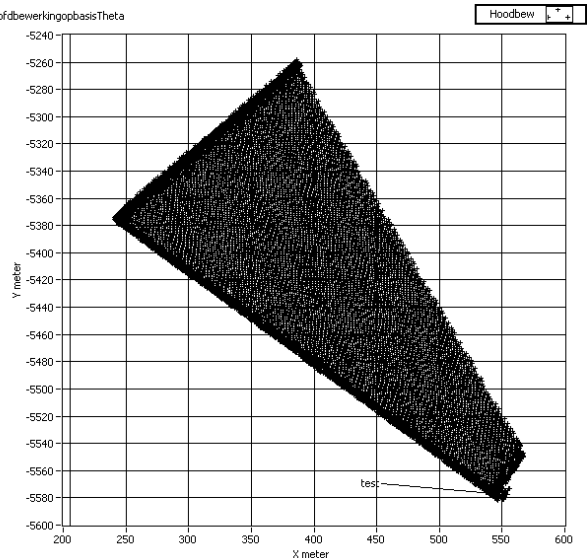
**Figuur 33 Kopakkerpunten Perceel 5 op basis  $\Delta 0$**



**Selectie Hoofdbewerkingspunten**



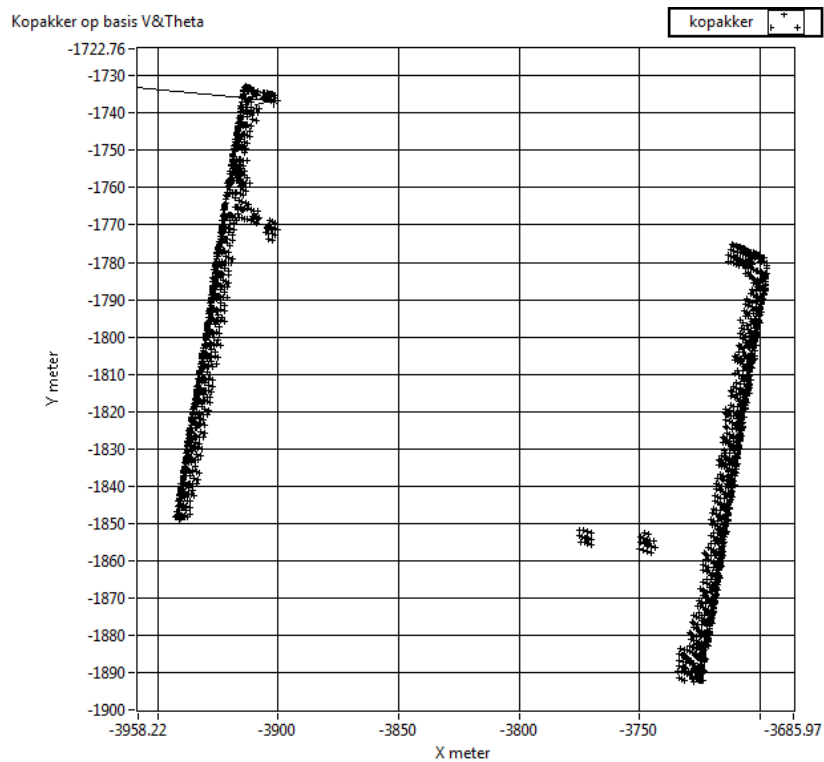
**Figuur 34 Kopakkerpunten Perceel 6 op basis  $\Delta 0$**



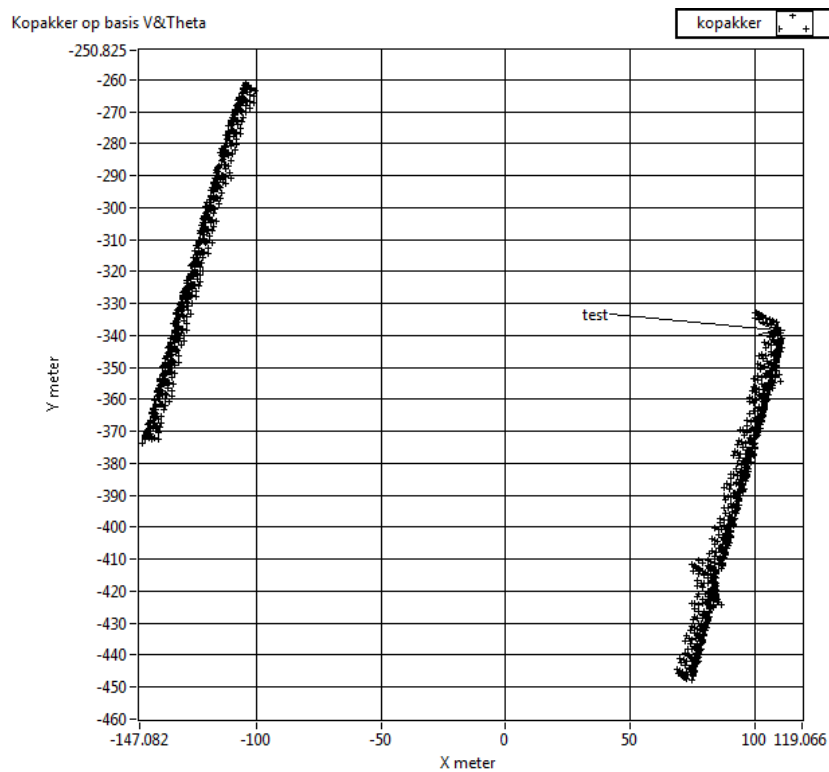
**Selectie Hoofdbewerkingspunten**



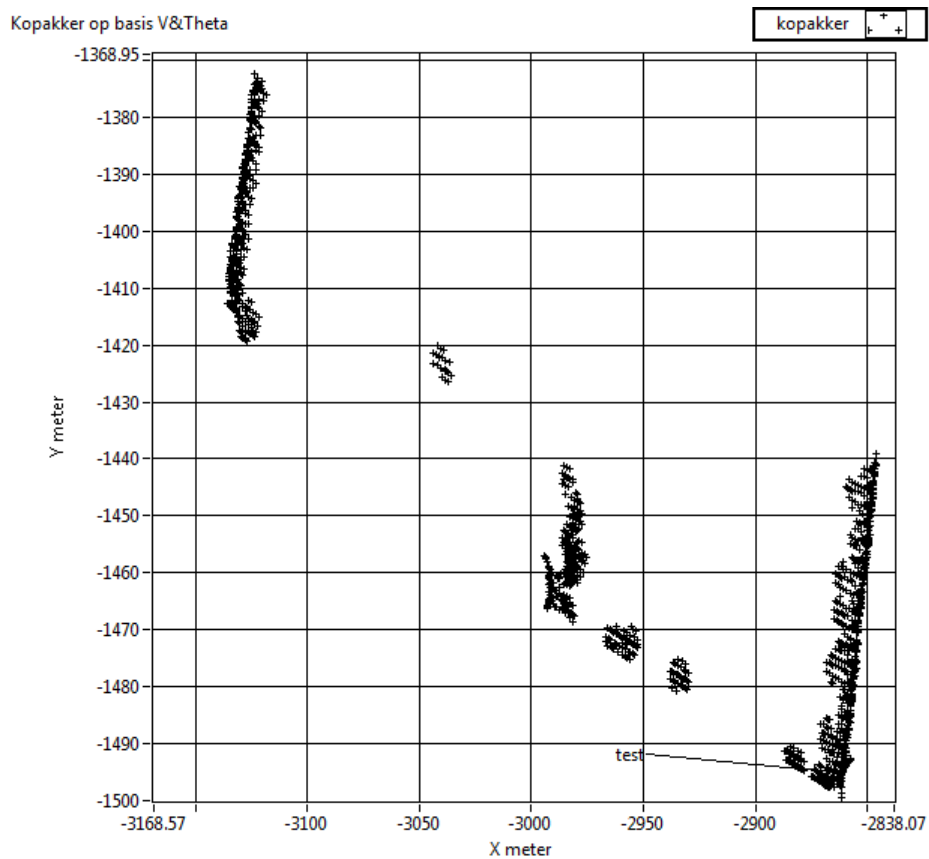
## A4 Resultaten kopakkerselectie op basis $\Delta\theta$ en V



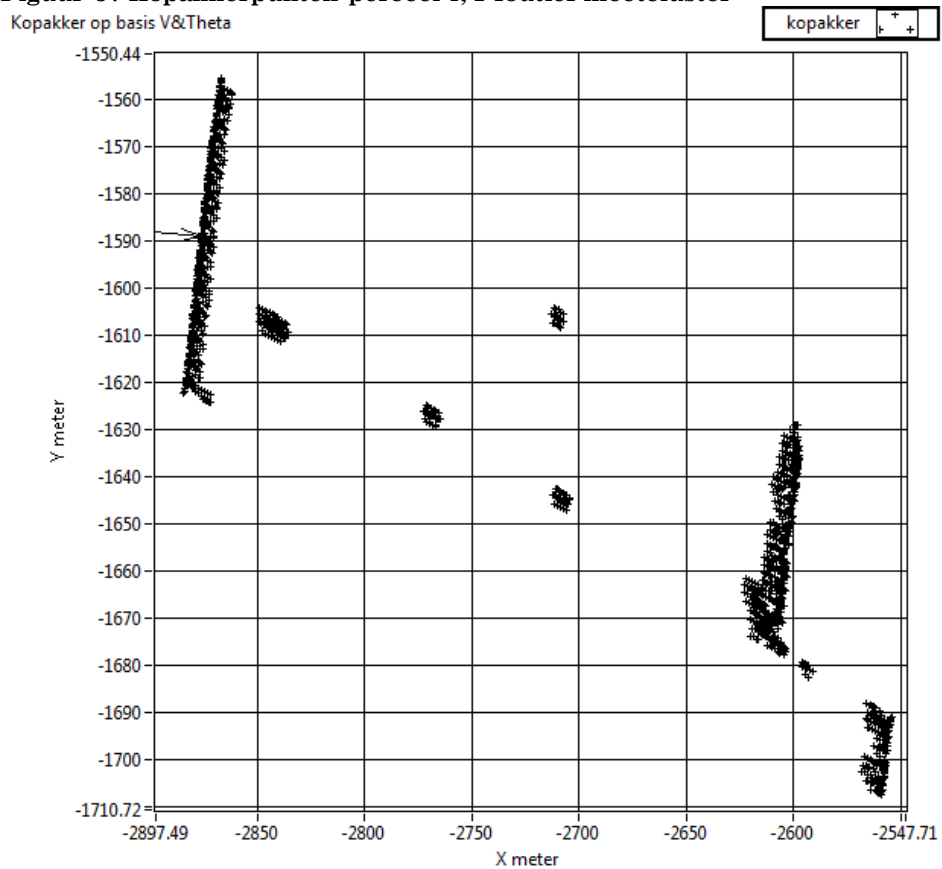
Figuur 35 Kopakkerpunten perceel 2, 3 foutieve meetclusters



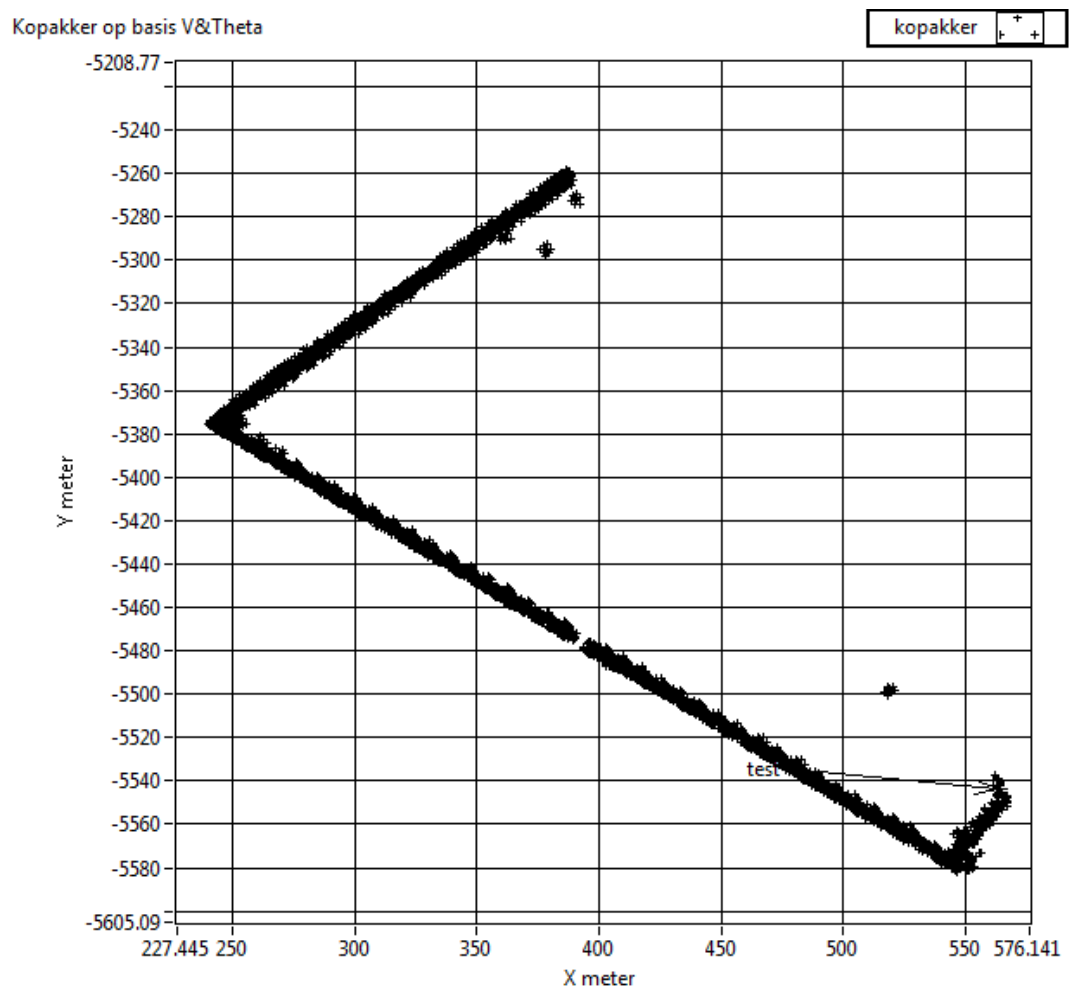
Figuur 36 Kopakkerpunten perceel 3, geen foutieve meetclusters



**Figuur 37 Kopakkerpunten perceel 4, 1 foutief meetcluster**



**Figuur 38 Kopakkerpunten perceel 5, 4 foutieve meetclusters**



**Figuur 394 Kopakkerpunten perceel 6, 3 foutieve meetclusters**

## A5 Resultaten 3 methoden kopakkerselectie

### Perceel 1

Tabel 9 Resultaten Perceel 1, kolom 1: 0=onbekend, 1=kopakker, 2=hoofdbew.

|                | Keuze $\Delta\theta$ | KeuzeKmeans | KeuzeCirkel |
|----------------|----------------------|-------------|-------------|
| Onbekend       | 3042                 | 0           | 0           |
| Kopakker       | 120                  | 739         | 572         |
| Hoofdbewerking | 1004                 | 3427        | 3594        |
| Totaal         | 4166                 | 4166        | 4166        |

### Perceel 2

Tabel 10 Resultaten Perceel 2, kolom 1: 0=onbekend, 1=kopakker, 2=hoofdbew.

|                | Keuze $\Delta\theta$ | KeuzeKmeans | KeuzeCirkel |
|----------------|----------------------|-------------|-------------|
| Onbekend       | 9813                 | 0           | 0           |
| Kopakker       | 585                  | 1752        | 1185        |
| Hoofdbewerking | 2941                 | 11587       | 12154       |
| Totaal         | 13339                | 13339       | 13339       |

### Perceel 3

Tabel 11 Resultaten Perceel 3, kolom 1: 0=onbekend, 1=kopakker, 2=hoofdbew.

|                | Keuze $\Delta\theta$ | KeuzeKmeans | KeuzeCirkel |
|----------------|----------------------|-------------|-------------|
| Onbekend       | 6800                 | 0           | 0           |
| Kopakker       | 475                  | 1819        | 848         |
| Hoofdbewerking | 1938                 | 7394        | 8365        |
| Totaal         | 9213                 | 9213        | 9213        |

### Perceel 4

Tabel 12 Resultaten Perceel 4, kolom 1: 0=onbekend, 1=kopakker, 2=hoofdbew.

|                | Keuze $\Delta\theta$ | KeuzeKmeans | KeuzeCirkel |
|----------------|----------------------|-------------|-------------|
| Onbekend       | 5814                 | 0           | 0           |
| Kopakker       | 299                  | 1138        | 1269        |
| Hoofdbewerking | 2124                 | 7099        | 6968        |
| Totaal         | 8237                 | 8237        | 8237        |

### Perceel 5

Tabel 13 Resultaten Perceel 5, kolom 1: 0=onbekend, 1=kopakker, 2=hoofdbew.

|                | Keuze $\Delta\theta$ | KeuzeKmeans | KeuzeCirkel |
|----------------|----------------------|-------------|-------------|
| Onbekend       | 7608                 | 0           | 0           |
| Kopakker       | 351                  | 1017        | 1271        |
| Hoofdbewerking | 3063                 | 10005       | 9751        |
| Totaal         | 11022                | 11022       | 11022       |

### Perceel 6

Tabel 14 Resultaten Perceel 6, kolom 1: 0=onbekend, 1=kopakker, 2=hoofdbew.

|                | Keuze $\Delta\theta$ | KeuzeKmeans | KeuzeCirkel |
|----------------|----------------------|-------------|-------------|
| Onbekend       | 12002                | 0           | 0           |
| Kopakker       | 1327                 | 4839        | 2737        |
| Hoofdbewerking | 2627                 | 11166       | 13268       |
| Totaal         | 16005                | 16005       | 16005       |