

ELEKTROTECHNIEK INNOVATIELAB
HOGESCHOOL VAN AMSTERDAM

Localistatie doormiddel van een Kalman-Filter

Hasan Yildiz, 500738676

Abstract

Dit rapport gaat over het probleem dat opgelost gaat worden tijdens het project innovatielab. Hierbij is er voor gekozen om bij Clean Mobility een opdracht te doen. De opdracht gaat over het onderzoeken en implementeren van een Kalman filter om nauwkeurig de locatie van een voertuig te bepalen. Daarbuiten ligt er een nadruk op het goed documenteren van de werking en de problemen die opkomen.

Inhoudsopgave

| | | |
|----------|---|-----------|
| 1 | Introductie | 1 |
| 1.1 | Probleemstelling | 1 |
| 1.2 | De opdracht | 1 |
| 1.2.1 | Doelstelling | 1 |
| 1.2.2 | Eisen | 2 |
| 1.3 | Vraagstelling | 2 |
| 1.4 | Verslag opbouw | 2 |
| 2 | Beschikbare sensoren | 3 |
| 3 | Kalman-Filter | 4 |
| 3.1 | Algemeen | 4 |
| 3.2 | Verschillende type Kalman-Filter | 5 |
| 3.3 | Implementatie | 6 |
| 3.3.1 | Voorspellingsmodel | 6 |
| 3.3.2 | Fout van voorspellingsmodel | 7 |
| 3.3.3 | Berekenen van de Kalman-Gain | 8 |
| 3.3.4 | Meting | 8 |
| 3.3.5 | Updaten van de staat | 8 |
| 3.3.6 | De fout van het Kalman-Filter bepalen | 9 |
| 3.3.7 | Overdragen | 9 |
| 3.4 | Inputs en outputs | 10 |
| 4 | Validatie | 11 |
| 4.1 | Methode | 11 |
| 4.2 | Resultaten | 11 |
| 4.2.1 | EvA in Parijs | 11 |
| 4.2.2 | H2A in London | 13 |
| 5 | Conclusie | 14 |
| 6 | Aanbevelingen | 15 |
| 6.1 | Inertial measurement unit | 15 |
| 6.2 | Validatie | 15 |
| 6.3 | Schonere data | 15 |

Hoofdstuk 1

Introductie

Dit hoofdstuk behandelt de probleemstelling, doelstelling en vraagstelling. Hierbij zullen ook de eisen kort besproken worden.

1.1 Probleemstelling

Bij Clean Mobility wordt er gewerkt aan verschillende voertuigen. De voertuigen loggen allerlei verschillende sensoren, denk hierbij bijvoorbeeld aan: geleverde vermogen, snelheid, locatie. De data kan later geanalyseerd worden om te conclusies te trekken over bijvoorbeeld de efficiëntie en optimaal rijgedrag. In 2016 werd de data bijvoorbeeld geanalyseerd om na de Shell Eco Marathon race te bepalen wat er nou precies mis ging.

Om optimaal rijgedrag te bepalen is het van belang dat de locatie van het voertuig zo nauwkeurig mogelijk wordt bijgehouden. Momenteel wordt de data van de GPS rauw gebruikt. Buiten de onnauwkeurigheid van de GPS is een ander probleem dat als het voertuig het GPS signaal kwijt raakt, door bijvoorbeeld onder een brug door te rijden, de locatie onzinnige waarden gaat uitspuwen.

1.2 De opdracht

In deze paragraaf wordt de opdracht zelf besproken, hierbij komen de doelstelling en eisen aan hand.

1.2.1 Doelstelling

Het probleem kort samengevat is dat de locatie van een voertuig nauwkeuriger geschat moet worden. Er wordt gebruikt gemaakt van een Kalman-Filter om het voorgeschreven probleem op te lossen.

Het minimale einddoel van dit project was om een goed en duidelijk onderzoek afgerond te hebben naar een implementatie van een Kalman-filter. Vervolgens moet een implementatie gemaakt worden die werkt over historische data. Indien er nog genoeg tijd overblijft, had er gewerkt kunnen worden aan een real-time implementatie binnen het systeem van een voertuig.

1.2.2 Eisen

Zoals is behandeld onder het kopje ‘Doelstelling’ is er minimaal verwacht dat er een werkende implementatie, over historische data, afgerond is. In het geval dat er geen real-time implementatie gemaakt kan worden is een implementatie over historische data een enorm waardevolle resultaat. Buiten het oplossen van het probleem is het van belang dat er zo veel mogelijk duidelijke documentatie over het proces wordt gemaakt. Na dit project zal een ander student het werk moeten oppakken en verder kunnen werken zonder tegen precies dezelfde obstakels te lopen. Hierbij wordt aangenomen dat de volgende student de documentatie goed door heeft genomen.

1.3 Vraagstelling

Om een Kalman-filter te implementeren moeten de volgende vragen beantwoord worden:

- Hoe gaat het filter precies te werk?
- Wat zijn de inputs van het systeem en wat betekenen de outputs?
- Hoe wordt de oorspronkelijke error bepaald?
- Hoe wordt de error van de gps-data bepaald?

De resultaten/uitwerkingen van deze vragen zijn te vinden in de desbetreffende paragrafen.

1.4 Verslag opbouw

Dit verslag dient als een ‘inleiding tot.’/‘uitwerking van.’, het is aangeraden om na dit verslag nog verder onderzoek te doen naar de werking van een Kalman-Filter. In hoofdstuk 2 wordt kort behandeld welke sensoren beschikbaar zijn die een rol kunnen spelen in het lokaliseren. Hoofdstuk 3 bespreekt het Kalman-Filter en de implementatie ervan. Hoofdstuk 4 bespreekt de validatie van de implementatie die is besproken in hoofdstuk 3. Vervolgens wordt er een conclusie getrokken in hoofdstuk 5 en sluit het rapport met een hoofdstuk over aanbevelingen.

Hoofdstuk 2

Beschikbare sensoren

In dit hoofdstuk worden alleen de relevante sensoren besproken van de twee voertuigen van Clean Mobility. De twee voertuigen zijn de H2A en EVA. De H2A is een waterstof raceauto en de EVA is een elektrische stadsauto. Er zijn genoeg sensoren in de voertuigen van Clean Mobility, echter zijn een paar eisen gesteld om het Kalman-Filter zo universeel en optimaal mogelijk te ontwerpen. Ten eerste moeten de sensoren op beide voertuigen aanwezig zijn, hierdoor zal het Kalman-Filter op beide voertuigen bruikbaar zijn. Daarbuiten moet de sensor een duidelijk en direct verband hebben met de locatie van het object. Bijvoorbeeld de temperatuur van de motordriver heeft geen direct verband met de locatie van het object en zal dus buiten beschouwing gelaten worden. De sensoren die wel direct effect hebben op de locatie zijn: Global Positioning System (GPS), snelheidssensor, stuurhoeksensor en de Inertial Measurement Unit (IMU). De stuurhoeksensor staat in de lijst omdat het een idee kan geven over de oriëntatie van het object. De IMU en GPS kunnen dit veel nauwkeuriger aangeven, daardoor moet de stuurhoeksensor niet beschouwt worden als ‘main’ sensor maar als een supportive sensor. Hiermee wordt bedoeld dat de oriëntatie niet bepaald moet worden door de stuurhoeksensor maar dat de stuurhoeksensor wel gebruikt kan worden om de oriëntatie van betere sensoren te valideren.

De GPS[5] die in de voertuigen zit maakt gebruik van de NMEA standaard, hieronder valt informatie over locatie, fout van de meting, verbonden satellieten, snelheid en hoek ten opzichte van "true north".

Niet alle sensoren die opgenomen zijn worden gebruikt in deze versie van het Kalman-Filter. Dit is niet door het gebrek aan meerwaarde maar door het gebrek aan tijd.

Hoofdstuk 3

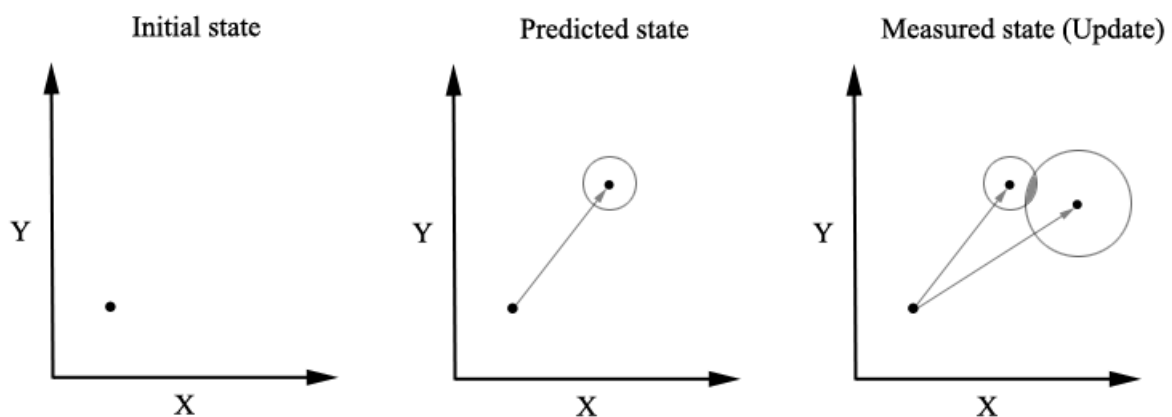
Kalman-Filter

In dit hoofdstuk wordt de werking van het Kalman Filter globaal behandeld en vervolgens wordt de implementatie diepgaand behandeld.

3.1 Algemeen

Kalman-filter is een algoritme die een optimale schatting maakt van een onbekende variabele met metingen die onzekerheid bevatten. Dit is mogelijk doordat het filter gebruik maakt gedeelde waarschijnlijkheidsverdeling van de sensoren/metingen. Het algoritme zal de meest waarschijnlijke oplossing bepalen.[4, p. 584] Dit is in meeste gevallen nauwkeuriger dan het gebruiken van de rauwe metingen. Kalman-filters kunnen in iedere situatie gebruikt worden waar de inputs een onzekerheid hebben met een gaussische verdeling. Een algemene toepassing is begeleiding, navigatie en controle van voertuigen/vaartuigen/vliegtuigen.

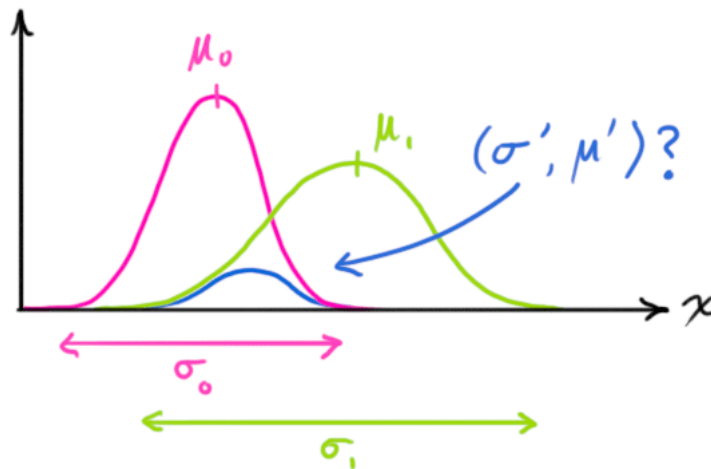
Het filter werkt in twee centrale stappen, een voorspelling en de update. De stappen zijn gevisualiseerd in figuur 3.1. Hierbij is het meest linker de oorspronkelijke waarde van de staat die benaderd moet worden, in dit geval een locatie. Vervolgens wordt de locatie voorspeld met de sensorwaarden, dit heeft een bepaalde fout die gevisualiseerd is als een grens rondom het punt. Ten slotte wordt er een meting gedaan van de locatie, in dit geval is het een relatief onnauwkeurig meting doordat de fout vele malen groter is dan die van de voorspelling. Doordat de meting en de voorspelling beide een gaussische verdeelde fout hebben kan het filter bepalen dat de meest waarschijnlijke locatie in de donkergrijze gebied zit.



Figuur 3.1: De centrale stappen van een Kalman-Filter op een x en y as.

Figuur 3.2 geeft weer waarom een gaussische verdeling het mogelijk maakt om de fouten van beide stappen te minimaliseren. Als beide bijvoorbeeld een uniforme verdeling hebben is het niet

mogelijk de meest waarschijnlijk positie te bepalen omdat vrijwel ieder punt even waarschijnlijk zou zijn. Zo brengen meeste andere verdelingen verschillende problemen met zich mee.



Figuur 3.2: Voorbeeld van gaussische verdelingen¹.

3.2 Verschillende type Kalman-Filter

Er zijn verschillende type Kalman-Filters, in tabel 3.1 staan er drie varianten. Het voornaamste verschil tussen de drie is het type model. Voor een lineair probleem zoals het benaderen van kamertemperatuur is een simpele Kalman-Filter prima. Echter werkt hetzelfde type filter niet bij een lokaal lineair probleem zoals simpele benadering van locatie. Hierbij wordt er vrijwel altijd gebruik gemaakt van de versnellingen van het object, dit heeft een kwadratisch verband met de tijd wat het probleem niet lineair maakt. Het probleem is echter te beschouwen als lokaal lineair en kan dus worden opgelost met een extended Kalman-Filter.[1][3]

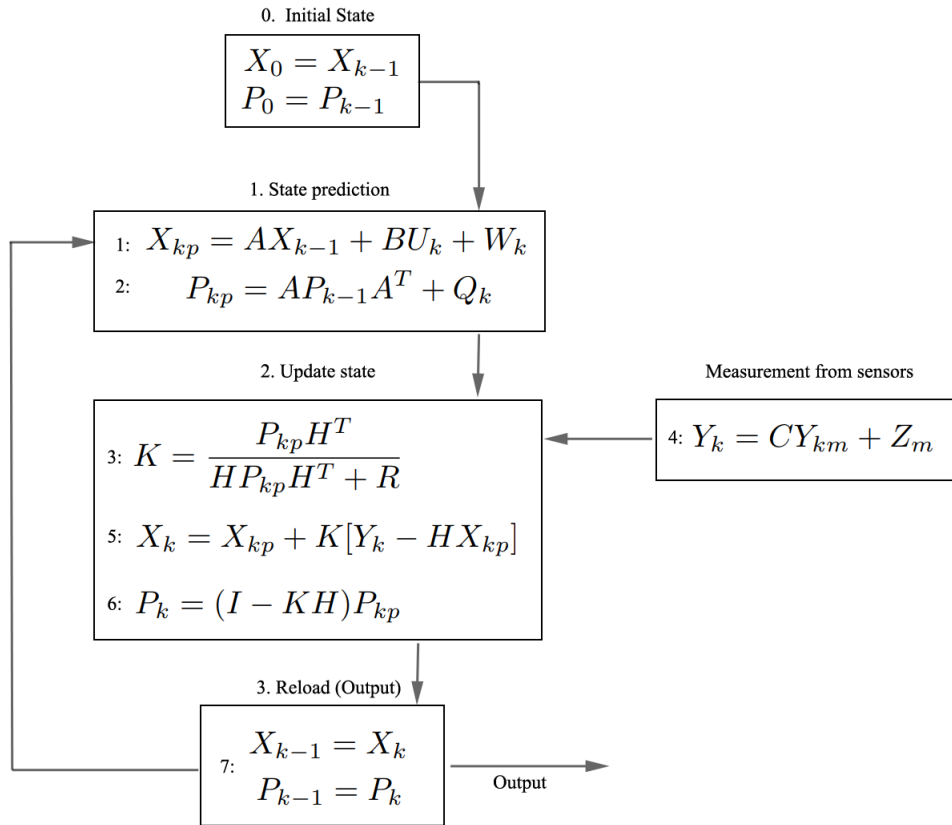
| State Estimator | Model | Rekenkracht |
|-------------------------|----------------|------------------|
| Kalman-Filter | Lineair | Weinig |
| Extended Kalman-Filter | Lokaal lineair | Weinig/Standaard |
| Unscented Kalman-Filter | Niet Lineair | Standaard |

Tabel 3.1: Verschillende type state-estimators

¹Source by <https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>

3.3 Implementatie

De implementatie van het Kalman-Filter staat wiskundig uitgewerkt in figuur 3.3, de figuur staat voor de rest van dit hoofdstuk centraal. Ieder stap in de figuur wordt in de desbetreffende subparagraaf behandeld (subparagraaf 3.3.5 is bijvoorbeeld stap 5).



Figuur 3.3: De flow van het Kalman Filter.

Er zijn drie (vier met de initialisatie) secties in iedere iteratie van het filter. De iteraties beginnen met het voorspellen van de staat van het model, in dit geval gaat het alleen om de locatie van een object. Vervolgens zal het de locatie updaten/corrigeren met de metingen die de sensoren aangeven. In de derde en laatste stap wordt de uitkomst van deze iteratie vrijgegeven. Alle berekeningen in figuur 3.3 zijn matrix berekeningen, het is daarbij handig om de benaming goed te door te nemen. De X matrix is de matrix waar de uitgangsvariabelen in staan en wordt ook wel de state-matrix genoemd, dit is de matrix die de waardes bevat die men wilt benaderen. De P matrix is de matrix de de fouten bevat van de resultaten in de X matrix. De index van de matrices spelen ook een belangrijke rol in het begrijpen van het hele systeem. Matrixen met een "p" in de index zijn resultaten van voorspellingen (prediction), een X matrix met een index waar p in voorkomt is bijvoorbeeld de voorspelde X matrix. Ten slotte zijn matrixen met een "m" in de index metingen (measurements).

3.3.1 Voorspellingsmodel

De eerste stap van elke iteratie is het voorspellen van de huidige staat, X_{kp} . Door middel van de uitkomst van de vorige iteratie (X_{k-1}), een controle variabele (U_k) en een error factor (W_k). De formule hiervoor dan ook:

$$X_{kp} = AX_{k-1} + BU_k + W_k \quad (3.1)$$

Indien het de eerste iteratie is zal het de initialisatie variabele gebruiken in plaats van de uitkomst van de vorige iteratie, want die bestaat simpelweg nog niet. De A en B matrix in deze vergelijking zijn vervormings matrixen. Deze matrices hoeven niet gebruikt te worden maar staan een mogelijkheid toe voor een extra vertaalslag tussen de matrix van U_k of X_{k-1} en de uitgangsmatrix. De implementatie hiervan zijn te zien in formule 3.2.

$$\begin{bmatrix} x_{kp} \\ y_{kp} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} + \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \cdot c \end{bmatrix} \left[\frac{\Delta T \cdot v_k}{(\frac{2\pi}{360} \cdot r) \cdot 3.6} \right] + 0 \quad (3.2)$$

$$c = \left(\cos \left(x_{k-1} \cdot \frac{\pi}{180} \right) \right)^{-1} \quad (3.3)$$

Hierbij worden x_{kp} (latitude) en y_{kp} (longitude) berekend door de uitkomst van de vorige iteratie op te tellen bij een vertaling van de snelheid in meter per seconde (m/s) naar de verschuiving ten opzichte van graden op aarde. Ten slotte wordt er nul bij op geteld, dit stelt de error factor voor. Het is nul omdat er wordt aangenomen dat de fout van de sensoren een gaussische verdeling heeft rondom nul, hierbij is de meest waarschijnlijk fout altijd nul.

De vertaalslag tussen de snelheid in m/s naar de verschuiving ten opzichte van grade op aarde is noodzakelijk omdat wiskunde alleen aan de kant staat van consistentie in eenheden. De uitgangsmatrix moet in latitude en longitude, die zijn in graden op aarde, de input van het filter is de wielsnelheid die in m/s is. Om van m/s naar verschuiving in graden te komen moet er eerst bepaald worden hoe ver het object verplaatst is in meters. Dit is te bepalen door de m/s te vermenigvuldigen met het aantal seconden met die snelheid, in dit geval is dat ΔT . De rest van de berekening wordt gedaan in kilometers dus zal het resultaat gedeeld moeten worden door 3.6 om uit te komen in kilometers. Ten slotte wordt er gedeeld door het aantal kilometer per graden (k/d) om uit te komen op de hoeveelheid verplaatsing in graden. k/d is bepaald door de radius straal van de aarde (r) te delen voor radialen.[2]

3.3.2 Fout van voorspellingsmodel

Nadat de huidige staat voorspeld is moet de mogelijke fout van dit antwoord berekend worden. De formule en implementatie hiervoor zijn:

$$P_{kp} = AP_{k-1}A^T + Q_k \quad (3.4)$$

$$\begin{bmatrix} E_{xkp} \\ E_{ykp} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} E_{xk-1} \\ E_{yk-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + 0.1 = \begin{bmatrix} E_{xk-1} \\ E_{yk-1} \end{bmatrix} \quad (3.5)$$

De vergelijking laat zien dat we de fout nemen van de vorige eindresultaat, indien het de eerste iteratie is zal het de initialisatie variabele gebruiken, optelt met een factor Q_k . De factor Q_k dient als een ondergrens voor de fout van de voorspelling (en het filter). De waarde hiervoor moet in de orde van de fout van de meting gekozen moeten worden en is noodzakelijk omdat het Kalman-Filter dient als voorspellingsmodel in het geval dat Q_k wordt weggelaten of gelijk staat aan nul. Stap 6 zal dit verder uitleggen.

3.3.3 Berekenen van de Kalman-Gain

Deze stap introduceert de matrix K , ook wel de Kalman-Gain. De Kalman-Gain dient als de rechter tussen de meting en de voorspelling, het dient om de blauwe verdeling in figuur 3.2 te bepalen. De formule voor het berekenen van de Kalman-Gain en de implementatie hiervan zijn:

$$K = \frac{P_{kp}H^T}{HP_{kp}H^T + R} \quad (3.6)$$

$$K = \frac{\begin{bmatrix} E_{xkp} \\ E_{ykp} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} E_{xkp} \\ E_{ykp} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} MEA_{HDOP} \\ MEA_{HDOP} \end{bmatrix}} \quad (3.7)$$

De H matrices zijn niet noodzakelijk in deze implementatie en zullen net als de A matrix een eenheidsmatrix voorstellen. De R matrix is de fout matrix van de meting die in de volgende stap mee zal doen. De Kalman-Gain wordt dus bepaald door te kijken naar de verhouding tussen de fout van de voorspelling en de totaal-systeem fout. Het is zal dus altijd tussen de 1 en de 0 zitten, waarbij 0 betekent dat het Kalman-Filter alleen hoeft te luisteren naar de voorspelling. Dit kan op twee manieren, de fout van de meting kan relatief extreem hoog zijn of de fout van de voorspelling kan relatief enorm klein zijn. In het geval dat de Kalman-Gain 1 is, is precies het omgekeerde aan de hand.

3.3.4 Meting

Vervolgens is er een tussenstap nodig om van de gemeten waarden naar de juiste metingen matrix te gaan. De formule en implementatie hiervoor zijn:

$$Y_k = CY_{km} + Z_m \quad (3.8)$$

$$\begin{bmatrix} MEA_x \\ MEA_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} MEA_{xm} \\ MEA_{ym} \end{bmatrix} + 0 \quad (3.9)$$

Er is niks speciaals aan de implementatie. De C matrix is net als de A matrix bedoeld als een mogelijk vertaalslag. De Z matrix is een extra fout matrix die in deze implementatie buiten beschouwing wordt gehouden.

3.3.5 Updaten van de staat

In deze stap komt de rechter, de Kalman-Gain, van pas. In deze stap zal de definitieve resultaat voor de X matrix bepaald worden. De formule en implementatie hiervoor zijn:

$$X_k = X_{kp} + K[Y_k - HX_{kp}] \quad (3.10)$$

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{kp} \\ y_{kp} \end{bmatrix} + K \left[\begin{bmatrix} MEA_{xm} \\ MEA_{ym} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{kp} \\ y_{kp} \end{bmatrix} \right] \quad (3.11)$$

Hierbij wordt de staat bepaald door de voorspelde staat op te tellen met het verschil tussen de gemeten staat en de voorspelde staat. De rechter besluit hoe erg de meting mee zal doen met het eindbesluit. Zoals in stap 3 uitgelegd is zal de meting geen invloed hebben op het filter als de Kalman-Gain de 0 benadert en zal de meting de uitkomst bepalen als de Kalman-Gain 1 benadert.

3.3.6 De fout van het Kalman-Filter bepalen

Als één van de laatste stappen wordt de fout van het Kalman-Filter bepaald, dit is ook de fout die de volgende iteratie in stap 2 zal gebruiken. De formule en implementatie hiervoor zijn:

$$P_k = (I - KH)P_{kp} \quad (3.12)$$

$$\begin{bmatrix} E_{xk} \\ E_{yk} \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - K \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} E_{xkp} \\ E_{ykp} \end{bmatrix} \quad (3.13)$$

Wat hier opmerkelijk aan is, is dat de fout van de output altijd kleiner zijn dan de fouten van de voorspelling en de meting doordat de fout van de voorspelde staat vermenigvuldigd wordt door I (I staat voor ‘identity matrix’, eenheidsmatrix) minus de Kalman-Gain zal. Dit is zeker niet verrassend als men terugdenkt aan figuur 3.2. Dit zorgt er voor dat de Q_k factor in stap 2 een essentiële rol speelt, als deze factor genegeerd wordt zal de fout vrijwel direct 0 benaderen en zal de Kalman-Gain altijd 0 worden. Hierdoor gaat het filter alleen luisteren naar de voorspellingen en is het filter in essentie een voorspellingsmodel geworden.

3.3.7 Overdragen

De laatste stap is het wegschrijven van de resultaten en het herladen van de staat matrix en de fout matrix voor de volgende iteratie. Dit gebeurt als volgt:

$$X_{k-1} = X_k \quad (3.14)$$

$$P_{k-1} = P_k \quad (3.15)$$

$$\begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad (3.16)$$

$$\begin{bmatrix} E_{xk-1} \\ E_{yk-1} \end{bmatrix} = \begin{bmatrix} E_{xk} \\ E_{yk} \end{bmatrix} \quad (3.17)$$

3.4 Inputs en outputs

In de vorige paragrafen is de implementatie besproken en zijn de inputs en outputs al bekend gemaakt. Deze paragraaf dient als samenvatting van alle ingangsvariabelen en uitgangsvariabelen. Het is niet verstandig om een iteratie te beginnen als niet alle ingangsvariabelen bekend zijn. Tabel 3.2 geeft alle variabele weer met een korte beschrijving.

| Variabel | Input/Ouput | Beschrijving |
|--|-------------|--|
| Latitude en longitude (x_k/y_k) | Ouput | De locatie van het object op aarde. |
| Fout van x_k/y_k (E_{xk}/E_{yk}) | Ouput | De fout van de uitgangsvariabele x_k en y_k in HDOP. |
| Tijd (ΔT) | Input | Seconden sinds vorige iteratie van het filter. |
| Snelheid (v_k) | Input | Snelheid van het object in m/s. |
| Hoek (θ) | Input | De hoek in graden ten opzichte van "true north". |
| Fout (MEA_{HDOP}) | Input | Fout van de meting (GPS). |
| Latitude GPS (MEA_{xm}) | Input | Latitude aangegeven door de GPS. |
| Longitude GPS (MEA_{ym}) | Input | Longitude aangegeven door de GPS. |

Tabel 3.2: Inputs en outputs van de huidige implementatie.

Hoofdstuk 4

Validatie

In dit hoofdstuk wordt de validatie van de implementatie van het vorige hoofdstuk besproken. Hierbij wordt de methode behandeld en vervolgens de resultaten.

4.1 Methode

Het Kalman-Filter wordt gevalideerd door het te testen met historische data. Het model wordt gebruikt om de races van 2018 in Parijs en London te filteren.

In Parijs is er rondje gereden met de real-time kinematic (RTK), de RTK geeft relatief een extreem nauwkeurige locatie weer. De metingen van de RTK worden genomen als waarheid om te berekenen wat de fout is van het Kalman model.

Het parkour in Londen bevat een deel waarbij er onder een brug door wordt gereden. In 2017 was dit een groot probleem voor de implementatie van de GPS, de locatie schoot over naar een nabije treinspoor. Echter lijkt dit niet meer het geval te zijn in 2018 en gaat de nieuwe implementatie beter om met de ‘blind spot’ onder de brug. Over beide datasets wordt hetzelfde Kalman model gebruikt en wordt de gefilterde data vergeleken met de rauwe data.

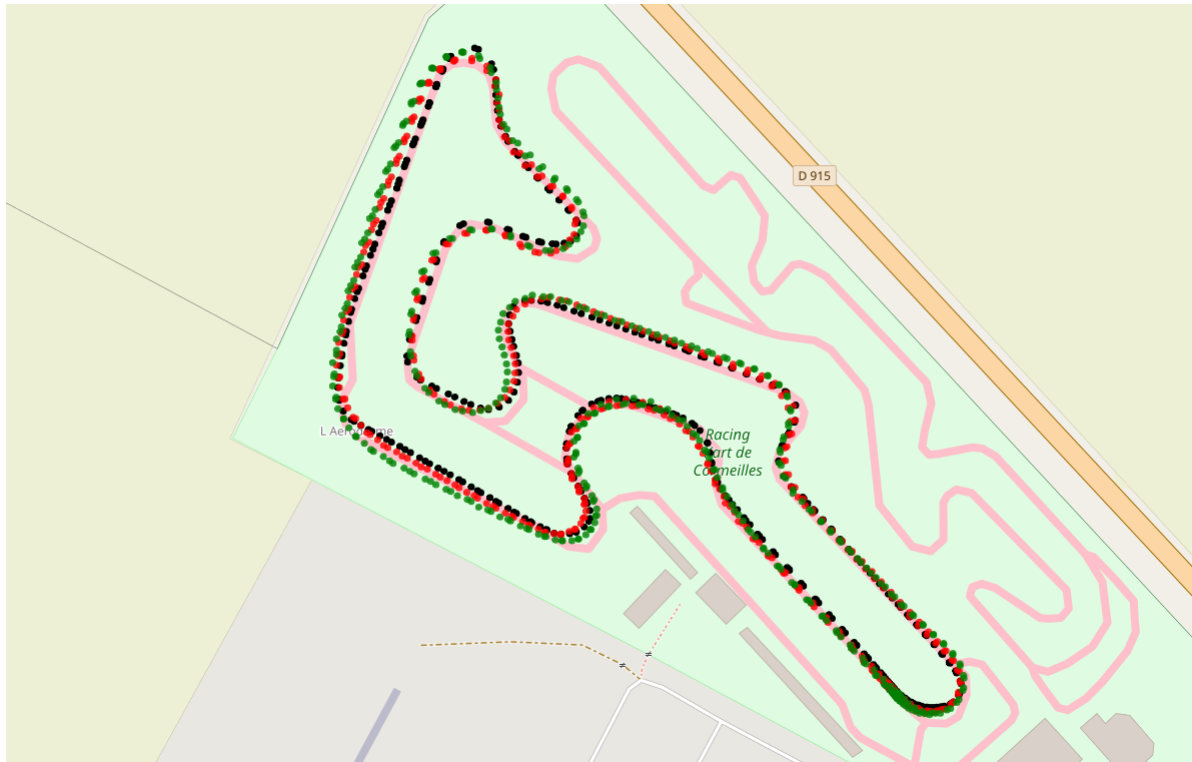
4.2 Resultaten

Zoals is uitgelegd in het vorige kopje zijn er twee historische datasets die gebruikt worden voor het valideren van het model. De resultaten van beide sets worden apart behandeld in de volgende paragrafen. De plots die voorkomen in dit verslag staan allemaal in de git repository. Hiervan zijn meest html plots, dit betekent dat de plots interactief zijn.

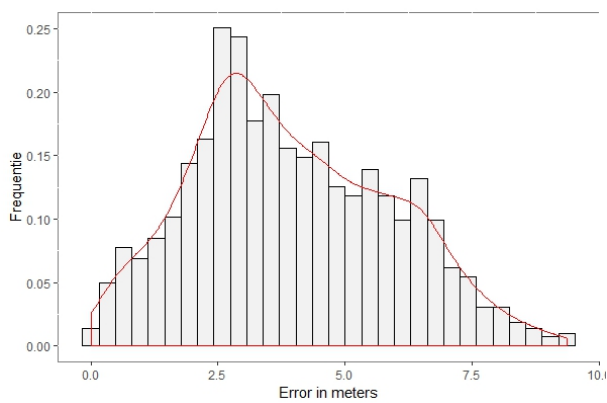
4.2.1 EvA in Parijs

Het parkour van Parijs is te zien in figuur 4.1, hierbij zijn de zwarte stippen de RTK waardes, de groene stippen de meningen van de GPS en de rode stippen de output van het Kalman model. In het eerste opzicht is de meerwaarde van het Kalman model al te zien, het is vooral duidelijk linksboven de plot.

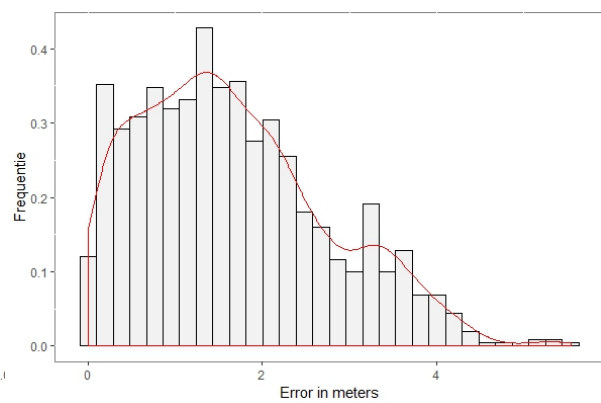
Door de RTK te beschouwen als het midden van de baan is het mogelijk om de fout te berekenen. De afstand tussen de punten en de meeste dichtbij punt op de baan worden individueel bekeken van zowel de GPS als het Kalman model. Het resultaat hiervan is te zien in figuur 4.2. Op de baan in Parijs heeft het Kalman-Filter de gemiddelde fout van 3.94 meter van de GPS verlaagd naar 1.65 meter. Dit is een enorme verbetering, echter moet er rekening gehouden worden met het feit dat hier is aangenomen dat de RTK op het midden van de baan heeft gestaan, de breedte van de baan is ook niet in beschouwing genomen. Alle fouten zijn ten opzichte van het midden van de baan en niet ten opzichte van de baan zelf.



Figuur 4.1: Het parkour in Parijs.



(a) Error van de GPS.



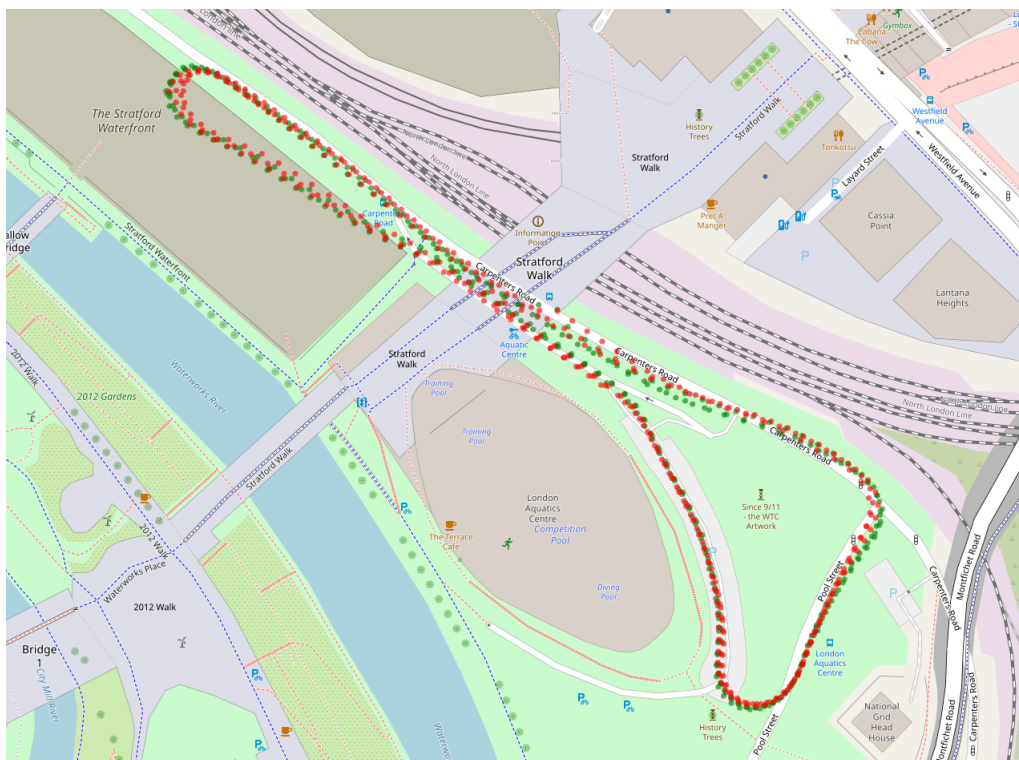
(b) Error van het model.

Figuur 4.2: De Error in meters ten opzichte van RTK.

4.2.2 H2A in London

De race die gereden is in London 2017 is te zien in 4.3, hierbij zijn de groene stippen de locatie van de GPS en de rode stippen de output van het Kalman model. Bij deze test was het vooral de bedoeling om te kijken naar het resultaat onder de brug, omdat de brug problemen kan opbrengen voor de GPS. Echter is het bij deze brug niet meer het geval bij deze implementatie van de GPS.

Daarbij dient de London dataset ook als test voor de toepasbaarheid van het Model. Doordat de EVA dataset ook de resultaten van de RTK hebben zijn de resultaten uit Parijs daadwerkelijk verifieerbaar. Echter loopt men al snel dan tegen het probleem aan dat het model zo wordt opgesteld dat het past bij een specifieke dataset omdat het bijvoorbeeld in dit geval de enige verifieerbare set is. De London set is dus ook om de toepasbaarheid van het model te testen. Veel meer dan discussiëren over de figuur 4.3 valt er echter niet te doen, het model lijkt te werken maar of het ook een meerwaarde heeft is niet bekend. Er moeten er meer testen gedaan worden om het model verder te valideren.



Figuur 4.3: Het parkour in London.

Hoofdstuk 5

Conclusie

Er is een Kalman-Filter ontworpen en geïmplementeerd om de locatie van de voertuigen van Cleanmobility te lokaliseren. Dit model is getest over historische data over een parkour in Parijs en een race in London. Er is een extra dataset beschikbaar voor het parkour in Parijs, de dataset bevat een ronde gemeten door een real-time kinematic. Deze dataset is beschouwd als waarheid en dient als precies het midden van de baan. Hierdoor is de fout te berekenen van de metingen en het model. Het resultaat hiervan is dat het model de gemiddelde fout van 3.94 meter, van de GPS, verlaagd naar 1.65 meter. Het model geef ook een soepele baan aan bij de data van de race in London. Echter valt er niet concreet te zeggen dat het Kalman-Filter in ieder volgende race/parkour meerwaarde zal hebben, hiervoor zullen er meer testen gedaan moeten worden.

Hoofdstuk 6

Aanbevelingen

Het rapport is niet compleet zonder een aantal aanbevelingen.

6.1 Inertial measurement unit

Het Kalman-Filter werkt momenteel zonder de sensor input van de inertial measurement unit (IMU). De IMU heeft een accelerometer die een enorme meerwaarde zal hebben in de voorspellingsfase van het filter, paragraaf 3.3.1. De accelerometer heeft een direct verband met de locatie van een voorwerp en is niet geïmplementeerd door het gebrek aan tijd. De accelerometer kan daarbij ook gebruikt worden om de snelheid te voorspellen van het object. (De snelheid zou hierbij in de X matrix komen te zitten) Daarbuiten heeft de IMU ook een nauwkeurige gyroscoop die de hoek van het object kan valideren.

6.2 Validatie

In de conclusie is er al vermeld dat er meer testen gedaan moeten worden om het filter verder te valideren. Er zijn maar twee bruikbare/representeerbare sets om een echt mee te testen. Het liefst worden er meer testen gedaan waarbij ook de RTK aanwezig is.

Het is het ook aangeraden om het werk van dit rapport te valideren. Hierbij is het verstandig om ook wat te testen met de Q factor uit paragraaf 3.3.2.

6.3 Schonere data

Het werken met de data van CleanMobility ging meestal niet soepel. Als voorbeeld zijn er vier races gemeten in London, hierbij zit de eerste dataset grotendeels op long/lat van 0/0, de tweede set was wel rondom de baan maar de metingen zaten zenden op de baan. Alleen de vierde dataset was volledig bruikbaar. Dit zijn punten waar naar gekeken moet worden.

Daarbuiten is de documentatie van de datasets niet altijd even duidelijk, vooral als het gaat om eenheden en grootheden. Een korte tip voor de opvolgers bij CleanMobility; gebruik de RAW data.

Bibliografie

- [1] S. Kar. Study of different types of gaussian filters (kalman ,extended kalman ,unscented,extended complex kalman filters). Master's thesis, National Institute of Technology, Rourkela, No Date.
- [2] Movable Type Ltd. *Calculate distance, bearing and more between Latitude/Longitude points*. <http://www.movable-type.co.uk/scripts/latlong.html>.
- [3] P. PICHLÍK. Comparison of different kalman filters types performance for a locomotive slip control purposes. Master's thesis, Czech Technical University, 2017.
- [4] S. Russell and P. Norvig. *Artificial Intelligence, A modern Approach*. Pearson, 3rd edition, 2016.
- [5] SkyTraq Technology, Inc. *Venus638FLPx GPS Receiver*, 0.7 edition, 2011. <https://cdn.sparkfun.com/datasheets/Sensors/GPS/Venus/638/doc/Venus638FLPx_DS_v07.pdf>.