

UNIVERSITEIT TWENTE

BACHELOROPDRACHT

**Invoegen van een auto in een
platoon op de snelweg**

EVELINE KOSTER
BEGELEIDER: ANTON STOORVOGEL

30 juni 2017

Inhoudsopgave

1	Inleiding	2
1.1	Probleemstelling	2
1.2	Aanpak	2
2	Model van de platoon	4
2.1	Eerste model	4
2.1.1	Stabiliteit	5
2.1.2	Resultaat omzetten	6
2.2	Tweede model	7
3	De invoegende auto	11
3.1	Opening kiezen	11
3.2	Opening vergroten	11
3.3	De functie r	13
3.4	Controller voor de invoegende auto	14
3.5	Verbetering voor de invoegende auto	16
4	De simulaties	19
5	Conclusie	20
6	Discussie	20

1 Inleiding

Op de snelweg bestaan veel fileproblemen. Verschillende oplossingen die hiervoor worden aangeleverd zijn wegverbredingen en auto's harder laten rijden. Beide hebben echter ook nadelen, zo kan een verbreding van een snelweg erg veel overlast geven aan omwonenden en zorgt harder rijden voor een stuk meer brandstofverbruik.

Een andere mogelijkheid om voor meer doorstroom op snelwegen te zorgen is door auto's dichter op elkaar te laten rijden. Dit kan natuurlijk niet zomaar, want dat zou erg gevaarlijke situaties creëren. Maar als de auto's automatisch op elkaar reageren, en er dus geen vertraging door de menselijke reactiesnelheid inzit, zou de minimaal benodigde afstand wel een stuk kleiner worden. Zogehete 'slimme' cruisecontrollers die dit voor elkaar krijgen zijn in sommige auto's zelfs al in gebruik. Met behulp van zo'n cruisecontroller zouden auto's dus dicht op elkaar kunnen rijden. Dit maakt een platoon van auto's. De auto's rijden in platoons van een aantal auto's over de snelweg en tussen de verschillende platoons zit een grotere afstand.

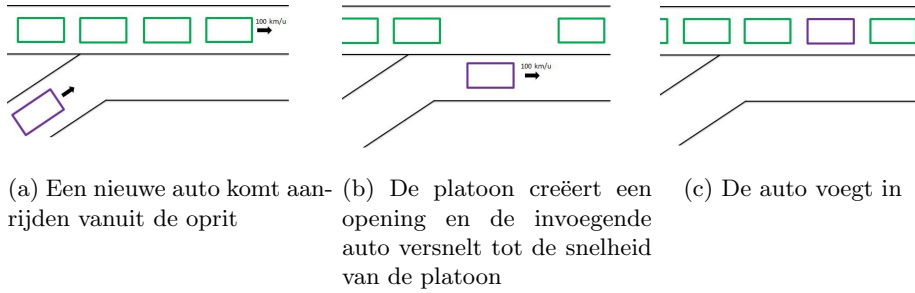
Een vraag die bij het rijden in platoons opkomt is: Hoe kan een nieuwe auto invoegen? Als een auto vanuit een oprit de snelweg opkomt, komt hij op de invoegstrook naast een platoon te rijden. De auto's rijden allemaal zo dicht op elkaar dat er geen mogelijk bestaat om zich ertussen te voegen. Ook kan de auto niet stil gaan staan op de invoegstrook en wachten tot de hele platoon voorbij is.

1.1 Probleemstelling

Als auto's in een platoon rijden kunnen nieuwe auto's niet zomaar invoegen. Hiervoor moet namelijk eerst een gat in de platoon gemaakt worden, waar de nieuwe auto dan kan invoegen. Als het om een invoegende auto gaat die vanuit een invoegstrook wil invoegen, is er een gelimiteerde tijd dat het invoegen mogelijk is. Ook rijdt de auto bij het bereiken van de oprit naar de snelweg nog niet op de juiste snelheid. Mogelijke problemen die zich hierdoor kunnen voordoen zijn dat de auto niet op tijd op snelheid komt of dat er niet op tijd een gat in de platoon kan worden gemaakt op de juiste plek. Een simulatie van het invoegen van een auto in een platoon op de snelweg kan meer zicht geven op de mogelijkheid van het invoegen van een auto in de platoon.

1.2 Aanpak

Er zal een simulatie gemaakt worden door een platoon te simuleren en het invoegen van een nieuwe auto hieraan toe te voegen. Voor deze simulatie zal gebruik worden gemaakt van lineaire modellen. Ook zal systeemtheorie worden gebruikt om naar de stabiliteit van het systeem te kijken. Eerst zal er een simulatie van de platoon worden gemaakt. Daarna wordt gekeken hoe een opening tussen twee auto's in de platoon vergroot kan worden. Hierna wordt de invoegende auto meegesimuleerd, deze moet in de vergrote opening invoegen.



In [2] wordt een protocol voor het invoegen van een enkele auto gegeven. Hier wordt echter pas begonnen met een gat creëren als de nieuwe auto al naast de platoon rijdt. Dit zorgt er natuurlijk voor dat het makkelijk is te bepalen waar de auto moet gaan invoegen, maar dit geeft heel weinig tijd om de opening te creëren in het geval dat de invoegende auto op een invoegstrook rijdt. Dit betekent dat de auto's in de platoon heel snel zullen moeten afremmen en versnellen om nog voor het eind van de invoegstrook een opening te maken. Als de invoegende auto op een parallelle rijlaan rijdt, zijn deze problemen niet van toepassing, maar bij een auto op de invoegstrook dus wel. Als de platoon al eerder begint met het creëren van een opening zouden deze problemen kunnen worden opgelost. Hierom zal gekeken worden naar de mogelijkheid om een gat te creëren op het moment dat de auto de oprit naar de snelweg op komt.

2 Model van de platoon

Voor de auto's in de platoon geldt voor auto i de volgende differentiaalvergelijking:

$$\dot{p}_i(t) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{pmatrix} p_i(t) + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix} u_i(t) \quad (1)$$

Waarbij $\tau = 0.1$ en

$$p_i = \begin{pmatrix} q_i \\ v_i \\ a_i \end{pmatrix}$$

Dit model is overgenomen uit [1].

De positie van een auto, q_i , wordt genomen als het middelpunt van de auto. Naast de snelweg loopt voor een aantal meter een invoegstrook. Het begin van de invoegstrook wordt als $q = 0$ genomen.

Verder begint de telling van de auto's bij de auto vooraan, bij auto 1. Voor auto 1 wordt in het model nog een auto toegevoegd, auto 0. Deze virtuele auto rijdt met een constante snelheid voor auto 1 zodat auto 1 deze 'volgt'. u_0 wordt gelijk genomen aan 0.

De lengte van de auto's wordt op 5 meter genomen. In de simulatie wordt gecheckt of er botsingen zijn voorgekomen door te kijken naar het verschil in de positie van de auto's in de platoon met de lengte van de auto's meegenomen.

2.1 Eerste model

In [1] wordt om een controller voor de auto's in de platoon te ontwerpen een gewenste afstand tussen de auto's gedefinieerd. Hierna wordt er een nieuwe toestand gedefinieerd, die naar 0 moet convergeren. Deze bewerking wordt in deze sectie besproken.

De juiste afstand tussen twee auto's wordt gezet op $r + hv_i$. Hierbij is r gelijk aan 0.2 en h aan 0.6, wat betekent dat, als de platoonsnelheid 100 km per uur is, er 16,87 meter tussen de middelpunten van twee auto's zit. De werkelijke afstand tussen twee autos is $q_{i-1} - q_i$. Het verschil hiertussen wordt gegeven door e_i :

$$e_i(t) = q_{i-1}(t) - q_i(t) - r - hv_i(t) \quad (2)$$

e_i moet in dit geval dus 0 zijn of worden. Met behulp van vergelijking (1) kunnen de afgeleides berekend worden:

$$\begin{aligned}
\dot{e}_i &= v_{i-1} - v_i - ha_i \\
\ddot{e}_i &= a_{i-1} - a_i + \frac{h}{\tau}a_i - \frac{h}{\tau}u_i \\
\ddot{\bar{e}}_i &= -\frac{1}{\tau}\ddot{e}_i(t) + \frac{1}{\tau}[u_{i-1}(t) - u_i(t) - h\dot{u}_i(t)]
\end{aligned} \tag{3}$$

Definieer $\bar{u}_i(t)$:

$$\bar{u}_i(t) = u_{i-1}(t) - u_i(t) - h\dot{u}_i(t)$$

Oftewel:

$$\dot{u}_i(t) = \frac{1}{h}u_{i-1}(t) - \frac{1}{h}u_i(t) - \frac{1}{h}\bar{u}_i(t) \tag{4}$$

Zodat:

$$\ddot{\bar{e}}_i = -\frac{1}{\tau}\ddot{e}_i(t) + \frac{1}{\tau}\bar{u}_i(t)$$

Hiermee kan een nieuw model beschreven worden:

$$\dot{x}_i(t) = Ax_i(t) + B\bar{u}_i(t) = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{pmatrix} x_i(t) + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix} \bar{u}_i(t) \tag{5}$$

Met

$$x_i = \begin{pmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \end{pmatrix}$$

2.1.1 Stabiliteit

Er moet een $\bar{u}_i(t)$ gevonden worden zodanig dat het systeem van vergelijking (5) stabiel is. Een stabiel systeem betekent namelijk dat de waarde van e_i naar 0 zal convergeren, wat betekent dat de auto's op een gegeven moment op de juiste afstand van elkaar zullen gaan rijden.

Maak van \bar{u}_i een lineaire controller:

$$\bar{u}_i(t) = Fx_i(t) = (f_1 \quad f_2 \quad 0) x_i(t)$$

f_3 wordt gelijk genomen aan 0. Dit is omdat de controller anders afhankelijk is van $a_{i-1} - a_i$, dus het verschil in versnelling tussen de twee auto's. Hoewel het verschil in positie en in snelheid goed te meten zijn voor een auto, bijvoorbeeld voor de snelheid met het dopplereffect, kan dit bij de versnelling een stuk minder

accuraat. Een controller die niet afhankelijk is van het verschil in versnelling, maar alleen het verschil in snelheid en positie zal dus betrouwbaarder zijn.

Voor een stabiel systeem geldt dat de functie $\chi(\lambda) = \det(\lambda I - (A + BF))$ negatieve nulpunten moet hebben. De juiste F hiervoor kan worden bepaald met de Routh Hurwitz test.

Voor $\chi(\lambda)$ geldt in dit geval:

$$\chi(\lambda) = \lambda^3 + \frac{1}{\tau}\lambda^2 - \frac{f_2}{\tau}\lambda - \frac{f_1}{\tau}$$

De Routh Hurwitz test wordt uitgevoerd:

$$\begin{array}{ccc} 1 & -\frac{f_2}{\tau} & \\ \frac{1}{\tau} & -\frac{f_1}{\tau} & \\ -f_1 - \frac{f_2}{\tau} & 0 & \\ -\frac{f_1}{\tau} & 0 & \end{array}$$

De elementen in de eerste kolom moeten allemaal hetzelfde teken hebben. Aangezien τ positief is, betekent dat in dit geval dat f_1 kleiner dan 0 moet zijn en dat f_2 kleiner moet zijn dan τf_1 .

In dit gebied kan dus elke waarde voor f_1 en f_2 geselecteerd worden, maar een te hoge waarde (dicht bij 0) zal betekenen dat de auto's in de platoon niet snel genoeg op een goede afstand van elkaar komen te rijden. Een te lage waarde (ver onder 0) betekent weer dat de auto's veel te abrupt moeten remmen of versnellen. Voor de resultaten in dit verslag zijn f_1 en f_2 gelijk genomen aan -1. Dit zorgt voor niet al te abrupt gedrag voor de auto's, maar lijkt ook niet te hoog te zijn, zodat er geen gevaarlijke situaties zouden ontstaan in de realiteit. In de simulatie kunnen f_1 en f_2 worden aangepast.

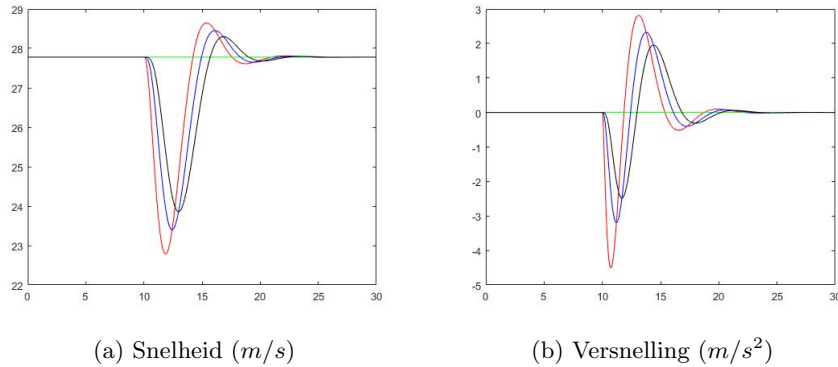
2.1.2 Resultaat omzetten

Om te zorgen dat ook q_i , v_i en a_i bepaald kunnen worden, moet u_i ook bekend zijn. Hiervoor wordt de differentiaalvergelijking van u_i (vergelijking (4)) toegevoegd aan het systeem:

$$y_i = \begin{pmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \\ u_i \end{pmatrix}$$

Het totale systeem ziet er dan als volgt uit:

$$\dot{y}_i = \begin{pmatrix} A + BF & O \\ -\frac{1}{h}F & -\frac{1}{h} \end{pmatrix} y_i + \begin{pmatrix} O \\ \frac{1}{h} \end{pmatrix} u_{i-1}$$



Figuur 2: De opening tussen rood en groen wordt verbreed voor de invoeger, zwart rijdt achter blauw rijdt achter rood

Dit wordt in matlab voor elke i apart opgelost met *lsim*, waarbij begonnen wordt bij $i = 1$. $u_0 = 0$, zodat telkens de u_{i-1} bekend is. De tijdstapgrootte, dt , wordt op 0.01 gezet.

De relaties tussen de elementen van y_i en a_i , v_i en q_i volgen uit vergelijking (3):

$$\begin{aligned}
 a_i &= \frac{\tau}{\tau - h} a_{i-1} - \frac{h}{\tau - h} u_i - \frac{\tau}{\tau - h} \ddot{e}_i \\
 v_i &= v_{i-1} - h a_i - \dot{e}_i \\
 q_i &= q_{i-1} - h v_i - r - e_i
 \end{aligned}$$

Met deze recursieve formules kunnen dus de waardes voor elke auto in de platoon bepaald worden.

In figuur 2 zijn de snelheid en versnelling te zien als er een opening wordt gecreëerd voor de invoegende auto. Het is te zien dat de auto's vrij abrupt hard moeten remmen om de opening te creëren. Dit wordt verder besproken in secties 3.2 en 3.3. Ook het model dat hierna beschreven wordt helpt bij het vergroten van de opening.

2.2 Tweede model

Vanwege de manier waarop e_i gedefinieerd is (zie vergelijking (2)), reageert alleen de auto aan de achterkant van de opening als deze opening te klein of te groot wordt. Het zou natuurlijk handiger zijn, vooral als er een nieuwe auto wil invoegen, dat ook de auto voor de opening gaat versnellen om plek te maken. Dit wordt ook in [1] gedaan. Dit kan door e_i anders te definiëren:

$$e_i = l_1(q_{i-1} - q_i - h v_i - r) + l_2(q_i - q_{i+1} - h v_i - r) \quad (6)$$

Hierbij zijn l_1 en l_2 de weegfactoren van beide termen. $l_1 = 0.7$ en $l_2 = 0.3$, zodat het reageren op de auto voor de prioriteit krijgt op reageren op de auto achter.

Met deze definitie van e_i kan er niet meer gebruik gemaakt worden van recursie zoals dat hiervoor gedaan is om q_i , v_i en a_i te bepalen. Wel kan op dezelfde manier, door de afgeleides van e_i te berekenen, de differentiaalvergelijking voor e_i bepaald worden:

$$\begin{aligned}\dot{e}_i &= l_1(v_{i-1} - v_i - ha_i) + l_2(v_i - v_{i+1} - ha_i) \\ \ddot{e}_i &= l_1(a_{i-1} - a_i + \frac{h}{\tau}a_i - \frac{h}{\tau}u_i) + l_2(a_i - a_{i+1} + \frac{h}{\tau}a_i - \frac{h}{\tau}u_i) \\ \ddot{e}_i &= -\frac{1}{\tau}\ddot{e}_i + \frac{l_1}{\tau}(u_{i-1} - u_i - hu_i) + \frac{l_2}{\tau}(u_i - u_{i+1} - hu_i)\end{aligned}\quad (7)$$

Hieruit volgt dat in dit geval de volgende vergelijking voor \bar{u}_i te gebruiken is:

$$\bar{u}_i = l_1(u_{i-1} - u_i - hu_i) + l_2(u_i - u_{i+1} - hu_i)$$

Wat om te schrijven is naar de volgende differentiaalvergelijking voor u_i :

$$\begin{aligned}\dot{u}_i &= \frac{l_1}{h(l_1 + l_2)}u_{i-1} + \frac{l_2 - l_1}{h(l_1 + l_2)}u_i - \frac{l_2}{h(l_1 + l_2)}u_{i+1} - \frac{1}{h(l_1 + l_2)}\bar{u}_i \\ &= \frac{l_1}{h(l_1 + l_2)}u_{i-1} + \frac{l_2 - l_1}{h(l_1 + l_2)}u_i - \frac{l_2}{h(l_1 + l_2)}u_{i+1} - \frac{1}{h(l_1 + l_2)}Fx_i\end{aligned}\quad (8)$$

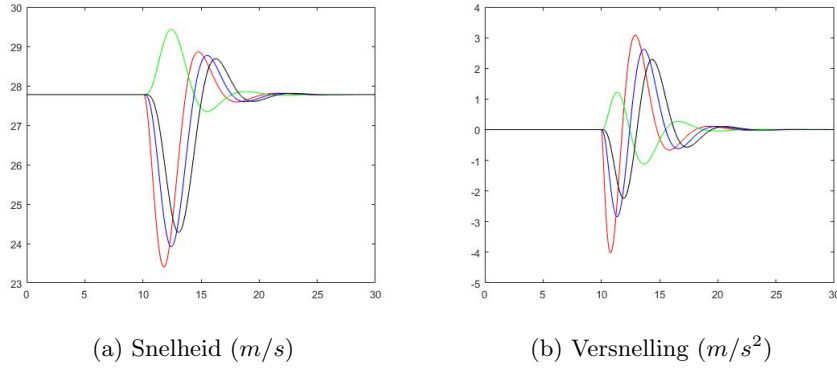
Dit zorgt ervoor dat weer geldt: $\ddot{e}_i = -\frac{1}{\tau}\ddot{e}_i(t) + \frac{1}{\tau}\bar{u}_i(t)$.

Een punt waar nog naar gekeken moet worden zijn de voorste en achterste auto's van de platoon. De auto helemaal vooraan in de simulatie, auto 0, is de virtuele auto, welke geen controller heeft (oftewel een controller gedefinieerd als 0). Dit betekent dat auto 1 gewoon kan reageren op de auto voor en achter zich. Voor de achterste auto geldt dat deze alleen reageert op de auto voor zich en niet op de (niet bestaande) auto achter zich, wat betekent dat l_1 gelijk is aan 1 en l_2 aan 0. Dit geeft de oude vergelijkingen voor de afgeleides van e_N en u_N zoals in vergelijkingen (3) en (4).

Op deze manier blijft het systeem voor x_i hetzelfde als in vergelijking (5), maar nu met e_i en \bar{u}_i anders gedefinieerd. De Routh Hurwitz test geeft dus ook dezelfde resultaten en F houdt dezelfde voorwaarden voor stabiliteit.

Om nu het systeem op te lossen kan echter niet recursie worden gebruikt, waarbij elke auto wordt afgegaan, aangezien de auto's van zowel de auto voor als achter hen afhankelijk zijn. Hierom moeten alle e_i 's en u_i 's tegelijk bepaald worden. Er wordt een grote matrix voor A en een voor B opgesteld, wat een differentiaalvergelijking geeft voor e_i , \dot{e}_i , \ddot{e}_i en u_i voor alle $i = 1 : N$. Dit wordt opgelost in matlab waarna a , v , q moeten worden bepaald voor elke auto. Dit kan door een lineair systeem op te lossen van alle vergelijkingen die de verbanden aangegeven met e en u .

Differentiaalvergelijking voor x en u :



Figuur 3: De opening tussen rood en groen wordt verbreed voor de invoeger, zwart rijdt achter blauw rijdt achter rood

$$y = \begin{pmatrix} x_1 \\ \vdots \\ x_N \\ u_1 \\ \vdots \\ u_N \end{pmatrix}$$

Met x_i hetzelfde gedefinieerd als in vergelijking (5). Aan de hand van vergelijking (5) en vergelijking (8) kan dan het systeem worden opgesteld:

$$\dot{y} = \begin{pmatrix} A + BF & O & \dots & O & O & O & O & \dots & O \\ O & A + BF & \dots & O & O & O & O & \dots & O \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & \dots & A + BF & O & O & O & \dots & O \\ -\frac{1}{h(l_1+l_2)}F & O & \dots & O & \frac{l_2-l_1}{h(l_1+l_2)} & -\frac{l_2}{h(l_1+l_2)} & O & \dots & O \\ O & -\frac{1}{h(l_1+l_2)}F & \dots & O & \frac{l_1}{h(l_1+l_2)} & \frac{l_2-l_1}{h(l_1+l_2)} & -\frac{l_2}{h(l_1+l_2)} & \dots & O \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ O & O & \dots & -\frac{1}{h}F & O & O & \dots & \frac{1}{h} & -\frac{1}{h} \end{pmatrix} y + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{l_1}{h(l_1+l_2)} \\ 0 \\ \vdots \\ 0 \end{pmatrix} u_0 \quad (9)$$

Dit gehele systeem is stabiel vanwege het feit dat bekend is dat het systeem van vergelijking (5) stabiel is door de keuze van F . Omdat in vergelijking (5)

het verschil in positie, snelheid en versnelling asymptotisch stabiel is, is ook het systeem uit vergelijking (1) stabiel. Dit betekent dat u_i , de controller in vergelijking (1), begrensd moet zijn en dus is dit gehele systeem stabiel.

Met vergelijking (7) kan het systeem voor a , v en q opgesteld worden:

$$z = \begin{pmatrix} a_1 \\ v_1 \\ q_1 \\ \vdots \\ a_N \\ v_N \\ q_N \end{pmatrix}$$

$$\begin{pmatrix} b + \frac{1}{\tau}c & 0 & 0 & -l_2 & 0 & 0 & 0 & 0 & 0 & \dots \\ -c & b & 0 & 0 & -l_2 & 0 & 0 & 0 & 0 & \dots \\ 0 & -c & b & 0 & 0 & -l_2 & 0 & 0 & 0 & \dots \\ l_1 & 0 & 0 & b + \frac{1}{\tau}c & 0 & 0 & -l_2 & 0 & 0 & \dots \\ 0 & l_1 & 0 & -c & b & 0 & 0 & -l_2 & 0 & \dots \\ 0 & 0 & l_1 & 0 & -c & b & 0 & 0 & -l_2 & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 & \frac{h}{\tau} - 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 & -h & -1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & -h & -1 \end{pmatrix} z = \begin{pmatrix} \ddot{e}_1 + \frac{1}{\tau}cu_1 - l_1a_0 \\ \dot{e}_1 - l_1v_0 \\ e_1 + cr - l_1q_0 \\ \ddot{e}_2 + \frac{1}{\tau}cu_2 \\ \dot{e}_1 \\ e_1 + cr \\ \vdots \\ \ddot{e}_N + \frac{h}{\tau}cu_N \\ \dot{e}_N \\ e_1 + r \end{pmatrix}$$

Met $b = l_2 - l_1$ en $c = h(l_1 + l_2)$.

Deze twee systemen kunnen met matlab worden opgelost.

In figuur 3 zijn de snelheid en versnelling te zien als er een opening voor de invoegende auto wordt vergroot. Hier is te zien dat de auto voor de opening ook licht versneld om plek te maken.

3 De invoegende auto

Voor het invoegen van de nieuwe auto moeten er een aantal dingen gebeuren. De auto begint bij een oprit naar de snelweg. Er wordt aangenomen dat zodra de auto deze oprit oprijdt de platoon een signaal krijgt dat er een opening gemaakt moet worden en waar deze opening moet komen. De nieuwe auto moet dan de snelheid van de platoon bereiken en op de goede positie naast de platoon komen.

Als de auto de invoegpositie bereikt heeft, wordt gecheckt of hij naast de juiste opening rijdt en dat hij ongeveer de snelheid van de platoon heeft. Als dit zo is, wordt ervan uit gegaan dat de auto succesvol kan invoegen.

3.1 Opening kiezen

Er moet een opening gekozen worden die zal worden vergroot voor de invoegende auto. Om te bepalen welke opening geschikt is hiervoor, wordt een ruwe schatting gemaakt van het tijdstip dat de invoegende auto bij P_I , de invoegpositie, aankomt. Voor dit tijdstip wordt bepaald welke opening van de platoon het dichtst bij P_I is. Deze opening wordt gekozen als de opening waar de auto moet gaan invoegen.

Om te schatten op welk tijdstip de invoegende auto aankomt bij P_I wordt uitgegaan van een constante versnelling van 2 m/s^2 waarmee de auto versnelt over de oprit en invoegstrook, totdat de snelheid van de platoon bereikt is. De voorspelde positie van de auto wordt met zijn beginsnelheid en de genomen versnelling geüpdated totdat P_I bereikt is. Voor dit voorspelde tijdstip wordt de dichtsbijzijnde opening gekozen.

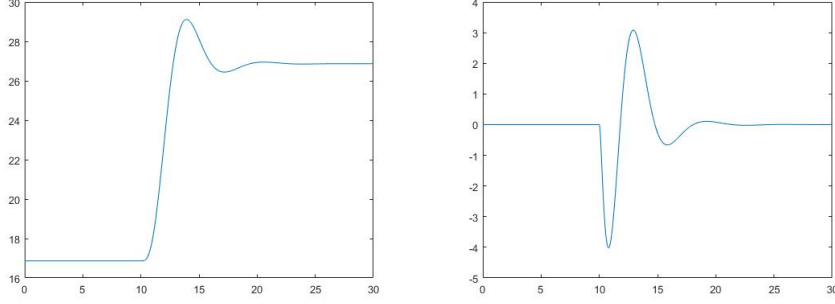
3.2 Opening vergroten

De platoon moet een opening tussen twee auto's gaan maken. De grootte van de opening moet groot genoeg zijn dat de auto soepel kan invoegen.

Om de opening tussen auto i en $i-1$ groter te maken, moet de gewenste grootte van deze opening, $r + hv_i$, een grotere waarde krijgen. Dit kan door r te verhogen met de gewenste vergroting. r verschilt dan dus per auto i en zal worden aangeduid met r_i . De veranderde r_i betekent dat e_i ook verandert. De waarde van e_i op het tijdstip dat r_i wordt aangepast zal opeens met de vergroting van r_i afnemen. e_i maakt dus een sprong. Dit kan gesimuleerd worden in matlab door de simulatie op te splitsen. Op het moment dat r_i verandert begint er een 'nieuwe' simulatie, waarbij voor de initiële conditie e_i gelijk is aan de laatste waarde in de vorige simulatie min de verandering in r .

De opening voor de invoegende auto wordt met 10 meter vergroot, zoals in figuur 4 te zien is. Hierbij is het tweede model voor de platoon gebruikt. Zoals te zien is gaat dit nogal abrupt. de auto's liggen dan ook opeens 10 meter te dicht op elkaar, wat betekent dat er heel abrupt geremd wordt. Dit is niet het gewenste gedrag, aangezien de opening ook wel wat langzamer vergroot kan worden in de meeste gevallen. De abruptheid kan verminderd worden door F

naar beneden aan te passen. Als twee auto's veel te dicht op elkaar rijden is het echter wel gewenst dat er abrupt gereageerd wordt. Om er toch voor te zorgen dat de opening langzamer verbreed wordt, kan r_i langzaam vergroot worden. Dit wordt bereikt door van r_i een functie te maken die loopt tussen t_1 , het tijdstip dat wordt begonnen met de opening vergroten, en t_2 , het tijdstip dat r_i groot genoeg moet zijn.



(a) De afstand tussen auto 4 en auto 3 wordt van 16.87 naar 26.87 gezet.

(b) Versnelling in m/s^2

Figuur 4: In model 2 wordt r abrupt vergroot

Als r_i niet meer constant is, betekent dit dat zijn afgeleide niet wegvalt bij het berekenen van de afgeleides van e_i . Dit betekent dus dat de afgeleides van e_i uit vergelijking 2 er als volgt uit gaan zien:

$$\begin{aligned}
\dot{e}_i &= v_{i-1} - v_i - ha_i - \dot{r}_i \\
\ddot{e}_i &= a_{i-1} - a_i + \frac{h}{\tau}a_i - \frac{h}{\tau}u_i - \ddot{r}_i \\
\ddot{\ddot{e}}_i &= -\frac{1}{\tau}a_{i-1} + \frac{1}{\tau}u_{i-1} + \frac{1}{\tau}a_i - \frac{1}{\tau}u_i - \frac{h}{\tau^2}a_i + \frac{h}{\tau^2}u_i - \frac{h}{\tau}\dot{u}_i - \ddot{\ddot{r}}_i \\
&= -\frac{1}{\tau}(a_{i-1} - a_i + \frac{h}{\tau}a_i - \frac{h}{\tau}u_i - \ddot{r}_i) + \frac{1}{\tau}(u_{i-1} - u_i - h\dot{u}_i) - \ddot{\ddot{r}}_i - \frac{1}{\tau}\ddot{\ddot{r}}_i \\
&= -\frac{1}{\tau}\ddot{\ddot{e}}_i + \frac{1}{\tau}\ddot{\ddot{u}}_i - \ddot{\ddot{r}}_i - \frac{1}{\tau}\ddot{\ddot{r}}_i
\end{aligned} \tag{10}$$

Waarbij weer geldt: $\dot{u}_i(t) = \frac{1}{h}u_{i-1}(t) - \frac{1}{h}u_i(t) - \frac{1}{h}\bar{u}_i(t)$.

Dit betekent dat het model voor auto i er nu als volgt uitziet, in plaats van als in vergelijking (5):

$$\dot{x}_i(t) = Ax_i(t) + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{\tau} & -\frac{1}{\tau} & -1 \end{pmatrix} \begin{pmatrix} \bar{u}_i \\ \dot{\ddot{r}}_i \\ \ddot{\ddot{r}}_i \end{pmatrix} (t) \tag{11}$$

Voor het tweede model geldt nu de volgende vergelijking voor e_i :

$$e_i = l_1(q_{i-1} - q_i - r_i - hv_i) + l_2(q_i - q_{i+1} - r_{i+1} - hv_i)$$

Dit betekent voor \ddot{e}_i :

$$\ddot{e}_i = \frac{1}{\tau}\ddot{e}_i + \frac{1}{\tau}\bar{u}_i - \frac{l_1}{\tau}\ddot{r}_i - l_1\ddot{r}_i - \frac{l_2}{\tau}\ddot{r}_{i+1} - l_2\ddot{r}_{i+1}$$

In het geval dat r_i verandert, zullen alleen de afgeleides van r_i ongelijk aan 0 zijn, de rest blijft gelijk aan 0, maar dit betekent dat in het tweede model voor \ddot{e}_{i-1} ook een nieuwe formule geldt:

$$\ddot{e}_{i-1} = \frac{1}{\tau}\ddot{e}_{i-1} + \frac{1}{\tau}\bar{u}_{i-1} - \frac{l_2}{\tau}\ddot{r}_i - l_2\ddot{r}_i$$

En dus is er in dit geval een aanpassing in het model voor x_i en x_{i-1} .

3.3 De functie r

Aan de functie r tussen t_1 en t_2 zitten een aantal eisen. De volgende gelijkheden moeten namelijk gelden (waarbij de opening met 10 meter vergroot zal worden):

$$r(t_1) = 0.2$$

$$r(t_2) = 5.2$$

$$\dot{r}(t_1) = 0$$

$$\dot{r}(t_2) = 0$$

$$\ddot{r}(t_1) = 0$$

$$\ddot{r}(t_2) = 0$$

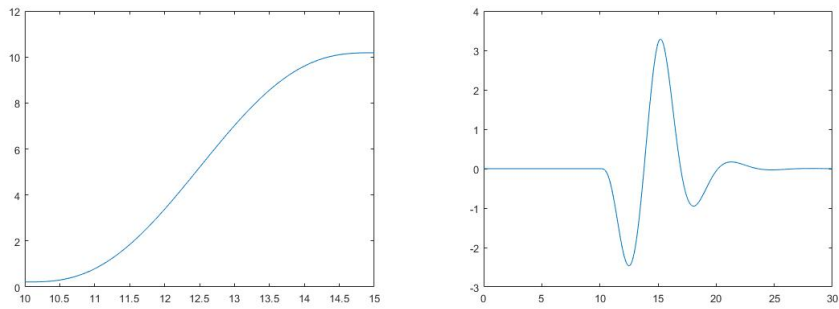
De eerste twee gelijkheden zijn overeenkomstig met het doel om de functie te laten starten op de originele waarde van r , 0.2, en te laten lopen tot de nieuwe waarde, 5.2. Verder moet de derde afgeleide van r bestaan, aangezien deze nu voorkomt in de differentiaalvergelijking. Hiervoor moeten de eerste en tweede afgeleide geen sprong maken en dus aansluiten aan de afgeleides van r in de tijd voor en na de functie. r is hier constant en dus moeten de afgeleides van de functie aan het begin en het einde 0 zijn.

Met deze zes voorwaarden en het begin- en eindtijdstip kan nu als functie een 5de graads polynoom gevonden worden: $a + bt + ct^2 + dt^3 + et^4 + ft^5$. Dit wordt met matlab gedaan door een stelsel voor de 6 coëfficiënten met voorwaarden op te lossen:

$$\begin{pmatrix} 1 & t_1 & t_1^2 & t_1^3 & t_1^4 & t_1^5 \\ 1 & t_2 & t_2^2 & t_2^3 & t_2^4 & t_2^5 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} 0.2 \\ 5.2 \end{pmatrix}$$

subject to: $\begin{pmatrix} 0 & 1 & 2t_1 & 3t_1^2 & 4t_1^3 & 5t_1^4 \\ 0 & 0 & 2 & 6t_1 & 12t_1^2 & 20t_1^3 \\ 0 & 1 & 2t_2 & 3t_2^2 & 4t_2^3 & 5t_2^4 \\ 0 & 0 & 2 & 6t_2 & 12t_2^2 & 20t_2^3 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

In figuur 5 zijn de resulterende functie voor r en de versnelling van de auto achter de opening te zien.



(a) r stijgt als functie tussen $t = 10$ en $t = 15$ (b) Versnelling van auto achter de opening (m/s^2)

Figuur 5: In model 2 wordt r als functie vergroot

Er kan nog gekozen worden in hoeveel seconden de r vergroot moet worden. In het geval van deze resultaten is r in 5 seconden vergroot. De keuze van de duur kan ook bijvoorbeeld afhangen van de schatting van het invoegtijdstip of van de lengte van de oprit, waardoor de platoon sneller reageert als er minder tijd of ruimte beschikbaar is om de opening te vergroten.

3.4 Controller voor de invoegende auto

Als de invoegende auto weet naast welke opening hij moet komen te rijden, kan, op dezelfde manier dat de auto's in de platoon een constante afstand van elkaar houden, het verschil tussen de positie van de opening en de positie van de invoegende auto geminimaliseerd worden. Het midden van de opening ligt voor auto q_{i+1} en de positie hiervan is dan gelijk aan $q_{i+1} + \frac{1}{2}(q_i - q_{i+1})$.

Neem q_I als de positie van de invoegende auto.

$$\begin{aligned}
e_I &= q_I - (q_{i+1} + \frac{1}{2}(q_i - q_{i+1})) \\
&= q_I - \frac{1}{2}(q_i + q_{i+1})
\end{aligned}$$

Hiervan worden de afgeleides bepaald:

$$\begin{aligned}
\dot{e}_I &= v_I - \frac{1}{2}(v_i + v_{i+1}) \\
\ddot{e}_I &= a_I - \frac{1}{2}(a_i + a_{i+1}) \\
\dddot{e}_I &= -\frac{1}{\tau}a_I + \frac{1}{\tau}u_I - \frac{1}{2\tau}(-a_i - a_{i+1} + u_i + u_{i+1}) \\
&= -\frac{1}{\tau}\ddot{e}_I + \frac{1}{\tau}(u_I - \frac{1}{2}u_i - \frac{1}{2}u_{i+1})
\end{aligned}$$

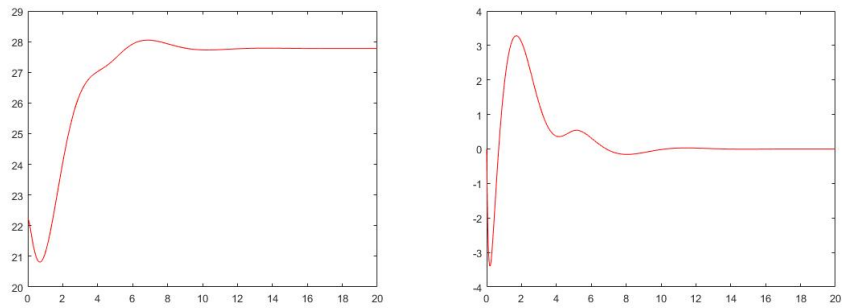
Hier kan weer een \bar{u} gedefinieerd worden:

$$\bar{u}_I = u_I - \frac{1}{2}u_i - \frac{1}{2}u_{i+1}$$

En dat geeft:

$$\dddot{e}_I = -\frac{1}{\tau}\ddot{e}_I + \frac{1}{\tau}\bar{u}_I$$

Hiermee kan voor de invoegende auto vergelijking (5) gebruikt worden, waarmee e_I , \dot{e}_I en \ddot{e}_I gevonden kunnen worden en dan a_I , v_I en q_I bepaald kunnen worden.



Figuur 6: Voorbeeld waarbij de auto eerst abrupt afremd en dan weer veel moet versnellen om de opening bij te houden

Het nadeel aan deze manier is dat de invoegende auto gelijk zo dicht mogelijk bij de opening probeert te komen. Hierdoor kan de auto in een keer heel erg

gaan versnellen of vertragen afhankelijk van waar de opening zich op dat moment bevindt. Dit is te zien in figuur 6, hier bevindt de opening zich aan het begin achter de invoegende auto, waardoor deze gaat remmen. Het kan zelfs voorkomen dat de auto achteruit gaat rijden. Dit gebeurt in het geval dat de opening zich aan het begin ver achter de auto bevindt. Dit is natuurlijk niet een goede strategie. Als de opening op het begin naast de invoegende auto wordt gekozen gaat het redelijk goed, zolang de beginsnelheid van de invoegende auto niet al te laag is, want anders zal de versnelling al snel boven de reële waarden uitkomen. Als maatregel tegen onrealistische versnellingen zou er een grens op de controller moeten worden gelegd. Al met al geeft deze controller niet het gedrag dat je van een invoegende auto zou verwachten.

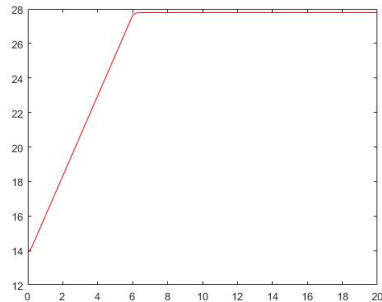
3.5 Verbetering voor de invoegende auto

De gewenste manier dat de auto naast de opening komt te rijden is dat de auto op de plaats van invoegen met de juiste snelheid bij de opening rijdt. Dit betekent dus dat de auto niet vanaf het begin van de oprit het doel hoeft te hebben om naast de opening te rijden, maar moet proberen om op hetzelfde tijdstip bij het invoegpunt te komen en de snelheid van de platoon te bereiken.

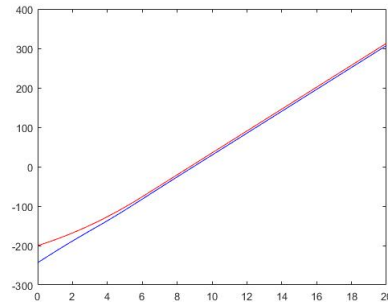
Dit kan bijvoorbeeld bereikt worden door de snelheid van de invoegende auto met een constante versnelling te laten toenemen totdat deze de snelheid van de platoon bereikt heeft. Op dat moment wordt de snelheid constant. Hierbij moet er ook voor gezorgd worden dat de invoegende auto op het juiste tijdstip, t_{invoeg} , op de invoegpositie is.

Bij deze manier wordt de controller u_I , die de versnelling aanpast, zo gekozen dat hij eerst gelijk is aan een constante, a , en vanaf een tijdstip, t_a , gelijk is aan 0. Het tijdstip t_a en de waarde voor a moeten bepaald worden door te kijken naar de bovengenoemde voorwaarden: De auto moet de snelheid van de platoon bereiken en op t_{invoeg} op de invoegpositie P_I zijn.

Voor het invoegtijdstip, het tijdstip dat de opening bij P_I is, geldt de volgende formule:



(a) De snelheid van de invoegende auto in m/s , beginnend bij t_1 met $50 km/u$



(b) De positie van de invoegende auto, beginnend bij t_1 aan het begin van de oprit, en de opening in de platoon

$$t_{invoeg} = \frac{P_I - \frac{1}{2}(q_i(t_1) + q_{i-1}(t_1))}{v_p}$$

Op t_1 komt de invoegende auto aan op het begin van de invoegstrook. Er gelden dan de volgende formules voor $v_I(t)$ en $q_I(t)$, de snelheid en positie van de invoegende auto:

$$v_I(t) = v_I(t_1) + a(t - t_1) \quad (12)$$

$$q_I(t) = q_I(t_1) + v_I(t_1)(t - t_1) + \frac{1}{2}a(t - t_1)^2 \quad (13)$$

De volgende relaties moeten gelden:

$$v_I(t_{invoeg}) = v_I(t_a) = v_p$$

$$q_I(t_{invoeg}) = P_I$$

Hieruit kan met vergelijkingen (12) en (13) worden afgeleid:

$$v_p = v_I(t_1) + a(t_a - t_1)$$

en:

$$\begin{aligned} P_I &= q_I(t_a) + v_p(t_a - t_1) \\ &= q_I(t_1) + v_I(t_1)(t_a - t_1) + \frac{1}{2}a(t_a - t_1)^2 + v_p(t_a - t_1) \end{aligned}$$

Dus:

$$\begin{aligned} a(t_a - t_1) &= v_p - v_I(t_1) \\ \rightarrow P_I &= q_I(t_1) + (t_a - t_1)\left(\frac{3}{2}v_p + \frac{1}{2}v_I(t_1)\right) \end{aligned}$$

Hieruit kan een uitdrukking voor $t_a - t_1$ worden afgeleid en hieruit volgt weer de formule voor a :

$$\begin{aligned} t_a - t_1 &= \frac{P_I - q_I(t_1)}{\frac{3}{2}v_p + \frac{1}{2}v_I(t_1)} \\ a &= \frac{(\frac{3}{2}v_p + \frac{1}{2}v_I(t_1))(v_p - v_I(t_1))}{P_I - q_I(t_1)} \end{aligned}$$

Dit systeem kan als een „open loop” of als een „closed loop” worden geïmplementeerd. Bij de open loop wordt t_{invoeg} eenmalig berekend als de invoegende auto is gesignaleerd, waarna t_a en a worden berekend. Hiermee wordt de controller u_I voor de invoegende auto gedefinieerd als gelijk aan a tussen t_1 en t_a en als gelijk aan 0 voor alle tijd na t_a .

Bij de open loop wordt er dus eenmalig schatting van t_{invoeg} gemaakt. Voor de closed loop geldt dat er nieuwe schattingen van t_{invoeg} worden gemaakt gedurende de rit over de oprit en invoegstrook. Met deze nieuwe schattingen veranderen t_a en a en dus zal u_I gedurende de rit aangepast worden als reactie op het gedrag van de platoon. De closed loop zorgt er dus voor dat er meer zekerheid is dat de invoegende auto en de opening in de platoon tegelijk op de goede plek zijn. Voor de closed loop is het echter nodig dat u_I niet lineair afhankelijk is van de toestand. Voor de berekening van a wordt in het geval van de closed loop bijvoorbeeld door $q_I(t)$ gedeeld. Dit is niet goed toe te voegen aan de simulatie zoals die nu is in matlab, waardoor het niet gelukt is het als een closed loop te implementeren. Hierom is het in de simulatie als open loop toegevoegd.

4 De simulaties

Er zijn verschillende simulaties in matlab gemaakt. Er zijn twee modellen voor het platoon gebruikt, een waar een auto reageert op de auto voor zich en een waar een auto reageert op zowel de auto voor als achter zich, zoals beschreven in sectie 2. Beide modellen kunnen van toepassing zijn op een auto afhankelijk van wat de auto voor mogelijkheden heeft. In alle simulaties is voor de invoegende auto de tweede controller gebruikt, beschreven in sectie 3.5.

De simulaties die gemaakt zijn voor deze twee platoon modellen zijn bijgevoegd:

```
simulatie_model1_r_functie  
simulatie_model2_r_functie
```

Beide laten r als functie toenemen om de opening te vergroten. De variant waarin r abrupt groter wordt gemaakt is ook bijgevoegd:

```
simulatie_model1_r_abrupt  
simulatie_model2_r_abrupt
```

5 Conclusie

Er zijn meerdere simulaties in matlab gemaakt om de mogelijkheid tot invoegen voor een aankomende auto te onderzoeken. Voor deze simulaties zijn lineaire modellen en controllers gebruikt. Verder is er nog een onderscheiding in de platoon of auto's wel of niet op de auto achter zich reageren.

Voor de invoegende auto zijn meerdere controllers gemaakt. De eerste was geen succes, aangezien deze de auto liet vertragen of achteruit rijden terwijl het de bedoeling was dat de auto snelheid ging maken. De tweede controller gaf de auto gedrag dat meer lijkt op het gewenste menselijke gedrag: versnellen totdat de snelheid van de auto's op de snelweg bereikt is en invoegen in een beschikbare opening. Met deze controller was te zien dat het de auto goed lukte om in te voegen in de meeste situaties. Het is dus gelukt om het invoegen van een nieuwe auto met een succesvolle controller te simuleren, maar er zijn nog heel wat verbeteringen aan deze simulatie mogelijk.

6 Discussie

De gemaakte simulatie is natuurlijk niet optimaal. Zo is er geen rekening gehouden met een vertraagde reactie van auto's, terwijl deze er in werkelijkheid wel is. Ook hebben auto's grenzen aan hun versnelling, wat niet meegenomen is in de modellen.

Voor de controller van de invoegende auto zijn ook nog talloze verbeteringen mogelijk. Nu is het zo dat de auto alleen met een constante snelheid kan versnellen totdat de juiste snelheid bereikt is. Er is dus bijvoorbeeld geen oplossing als de auto eerst te ver voor de platoon rijdt, waarbij het handig zou zijn als de auto even vertraagd of op lage snelheid doorrijdt. Ook zou door de snelheid op een andere manier dan lineair te laten toenemen het makkelijker kunnen zijn voor de invoegende auto om op de juiste snelheid te komen.

De succesvolste controller voor de platoon is nu als een open loop geïmplementeerd. Deze kan echter ook als een niet-lineaire controller voor een closed loop geïmplementeerd worden. Dit zal de kans op succesvol invoegen een stuk groter maken, vooral bij onverwachte situaties in het platoon.

Het simuleren van invoegende auto's die vlak na elkaar de oprit opkomen zou ook een goede toevoeging kunnen zijn. In werkelijkheid komt het namelijk ook voor dat er veel auto's tegelijk de snelweg op willen en het is nog de vraag hoe goed de platoon hiermee om kan gaan.

Referenties

- [1] Zegers, Jeroen C, Semsar-Kazerooni, Elham, Ploeg, Jeroen, van de Wouw, Nathan en Nijmeijer, Henk. *Consensus-based bi-directional CACC for vehicular platooning*. American Control Conference (ACC): pagina's 2578–2584, 2016. <http://ieeexplore.ieee.org/document/7525305/>
- [2] Ploeg, Jeroen en Semsar-Kazerooni, Elham. *Interaction protocols for cooperative merging and lane reduction scenarios*. 18th International Conference on Intelligent Transportation Systems: pagina's 1964–1970, 2015.