

Fuzzy control – niet zo vaag als gedacht

Corné Govers, Hans Nas, en Peter Klijn
Fontys Hogescholen - Lectoraat Automotive Control

High Tech Automotive Campus, Steenovenweg 1, Helmond
E-mail: c.govers@fontys.nl

Het onderzoek in het artikel is geïnspireerd door de casus ‘platooning’ uit de Grand Cooperative Driving Challenge. Er is een PreScan[®]/Sumulink[®] model opgesteld met daarin twee auto’s. De voorste auto volgt een vastgesteld snelheidsprofiel, de tweede auto volgt de eerste auto waarbij de tweede auto de snelheid van de eerste meet met behulp van een AIR-sensor. De besturing van het gaspedaal in beide auto’s vindt plaats met Fuzzy Logic Control in plaats van met een klassieke regelaar. Concluderend mag worden gesteld dat in dit verkennend onderzoek gebleken is dat de Fuzzy Logic Control techniek in principe werkt.

Trefwoorden: GCDC, Fuzzy Logic Control, platooning, PreScan

1. INLEIDING

Eén van de gestelde challenges in de “Grand Cooperative Driving Challenge” is platooning, ofwel het in colonne rijden van een lange rij auto’s op een aanzienlijke snelheid (zie het thema ‘Control algorithms for longitudinal cooperative behavior’ op [1]). Eén van de (ongewenste) verschijnselen bij platooning is het ontstaan van spookfiles. Spookfiles ontstaan over het algemeen als kleine fluctuaties in de onderlinge afstanden tussen de voertuigen. Deze fluctuaties worden steeds sterker met als gevolg dat op zeker moment zelfs auto’s tijdelijk stil komen te staan. Op afstand gezien lijkt het op de longitudinale voortplanting van een golf door een springveer. Een mooi filmpje welke dit illustreert is te vinden op NewScientist [2].

In wetenschappelijke termen spreekt men over ‘string stabiliteit voor heterogene voertuigketens’. Het verschijnsel is uit de praktijk al decennia bekend en er wordt dan ook al decennia lang onderzoek op gepleegd (zie o.a. de referenties in [3]). Oorspronkelijk was dit onderzoek theoretisch wiskundig van aard, later met krachtiger wordende computers werden simulatiemodellen bij het onderzoek betrokken. De laatste jaren worden tevens praktijkexperimenten uitgevoerd, ter verificatie van de modellen. Uit de onderzoeken kan samenvattend worden gesteld dat centrale aansturing van alle voertuigen in de colonne wel een stabiele string oplevert, maar dat dit praktisch onwerkbaar is. Decentrale aansturing, zoals bij CACC (cooperative adaptive cruise control) gebeurt, kan in geval van een slecht ontwerp leiden tot string instabiliteit. In de onderzoeken wordt onderscheid gemaakt tussen strings van homogene voertuigen, de voertuigdynamiek is identiek, en strings van heterogene voertuigen [4][5], waarbij de dynamiek per voertuig anders is. De laatste situatie benadert natuurlijk de praktijk zoals deze getest is (bij schrijven – wordt) in de GCDC.

In vrijwel alle onderzoeken naar string stabiliteit worden mathematische modellen opgesteld van zowel de voertuigen als de regeling. Een andere manier om deze problematiek op te pakken is door gebruik te maken van Fuzzy Logic Control, een ongelukkige, doch historisch zo gegroeide term welke niet-exactheid insinueert. Echter Fuzzy Logic is net zo deterministisch als de alom bekende binaire logica.

In dit artikel wordt toegelicht hoe Fuzzy Logic Control werkt, en is een eenvoudig model voor string stabiliteit in een CACC-systeem opgesteld met een fuzzy regeling. De simulatie omgeving waarin de experimenten zijn uitgevoerd is PreScan[®] [7] in combinatie met Matlab[®] / Simulink[®].

Sectie 2 beschrijft een aantal basisconcepten uit de experimentele opzet. In Sectie 3 wordt de toegepaste methode / experimentele opzet uitgewerkt. Sectie 4 beschrijft de resultaten van het simulatie experiment. In sectie 5 worden de resultaten geanalyseerd en sectie 6 besluit het artikel.

2. CONCEPTEN EN TERMINOLOGIE

PreScan[®] is een simulatie en verificatie omgeving voor intelligente voertuigsystemen. In deze software kunnen werelden met infrastructuur worden gecreëerd, voertuigen kunnen in deze wereld worden geplaatst en de voertuigen kunnen worden voorzien van detectiesystemen zoals radar, lidar, stereovision maar ook van communicatiesystemen als car-2-car of car-2-infrastructuur. Per voertuig kan een vaststaand traject worden vastgelegd, maar het is ook mogelijk om bijvoorbeeld voertuigen elkaar te laten volgen (hiervoor heb je dan wel een dynamisch model nodig waarover later meer). Dit geheel wordt in PreScan[®] termen een ‘experiment’

genoemd. Fig. 2.1 geeft een indruk van de 3D visualisatieomgeving van *PreScan*[®].

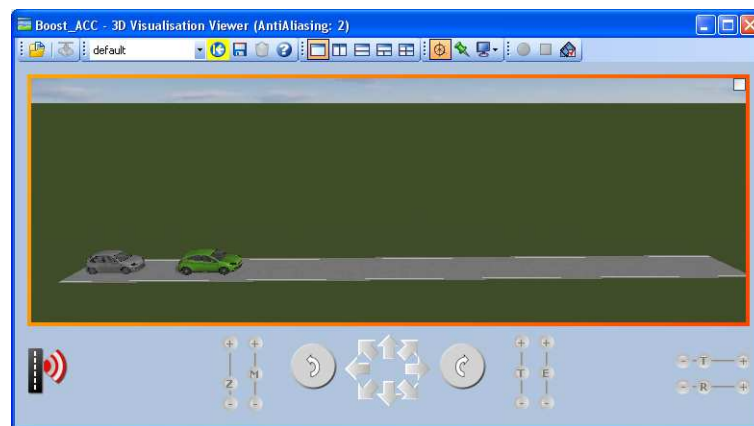


Fig. 2.1 Visualisatie in *PreScan*[®]

Voor de simulatie omgeving maakt *PreScan*[®] gebruik van *Simulink*[®], een onderdeel van *MATLAB*[®]. Daartoe dient eerst vanuit het *PreScan*[®]-experiment het *Simulink*[®]-model gebouwd te worden. Dit model heet in *PreScan*[®] termen een ‘*Compilation Sheet*’ (Fig. 2.2). Deze is te openen in *Simulink*[®] waarna het model eventueel uit te breiden is zoals je normaal een model in *Simulink*[®] bouwt, bijvoorbeeld auto 2 moet auto 1 volgen. De simulatie start je vanuit *Simulink*[®].

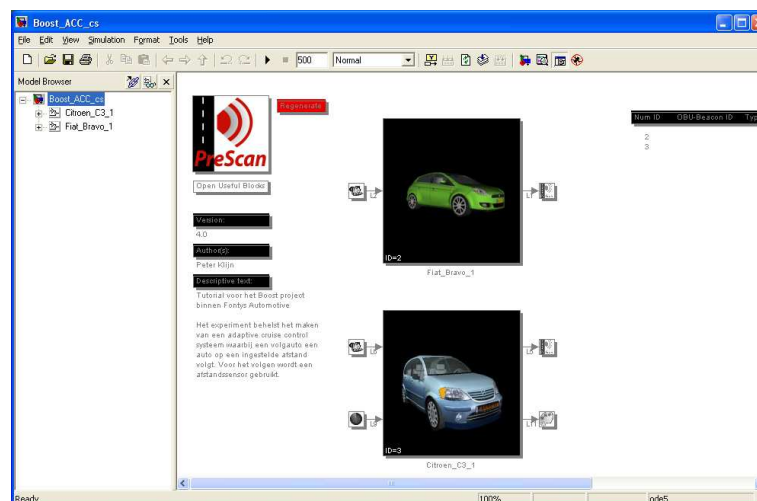


Fig. 2.2 Simulink compilation sheet

Fuzzy Logic Control. In de context van regelsystemen is Fuzzy Logic een probleemoplossende systeemaanpak welke zich leent voor implementatie in systemen variërend van eenvoudige embedded microcontrollers tot grote PC of workstation gebaseerde data-acquisitie en controlsystemen. *Matlab*[®]/*Simulink*[®] heeft een Fuzzy Logic Toolbox waarmee Fuzzy Logic Control kan worden beoefend.

Het kan worden geïmplementeerd in hardware, in software of in een combinatie van beide. Fuzzy Logic Control biedt een eenvoudige manier om tot een eenduidige conclusie te komen op basis van vage, ambigue, niet-precieze, ruisvolle of ontbrekende inputinformatie. De Fuzzy Logic manier van regelproblemen benadert de mens-als-regelaar, alleen veel sneller, en beter reproduceerbaar.

De fuzzy regelaar bestaat - eenvoudig gesteld - uit een verzameling beslissingsregels in een vorm van IF X AND Y THEN Z, in plaats van een mathematisch gemodelleerd systeem. Het Fuzzy Logic model is empirisch, eerder vertrouwend op de ervaring dan op de technische systeemkennis van de operator. Bijvoorbeeld “IF (mijn snelheid is hoog) AND (het verkeerslicht voor me is rood) THEN (ik moet hard remmen)” is de manier waarop je als mens zou reageren. Natuurlijk zijn er op het betreffende moment meerdere beslissingsregels welke meespelen.

In Fig. 2.3 wordt in schema aangegeven hoe een fuzzy controller in principe de plaats kan innemen van een 'gewone' regelaar.

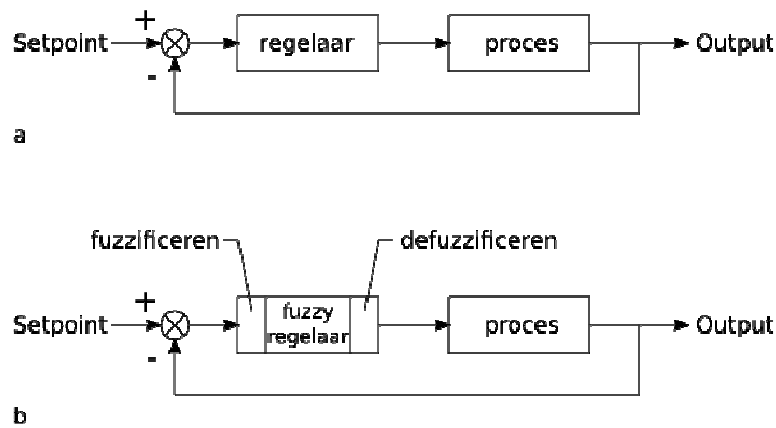


Fig. 2.3 a. Mathematische regelaar, b. Fuzzy logic regelaar

Aan de ingang worden numerieke waarden aangeboden. Er is een interface nodig om de numerieke waarden om te zetten in voor de fuzzy regelaar zinvolle waarden als 'hoog', 'korte afstand' enzovoorts (dit proces heet *fuzzificeren*). De fuzzy regelaar regelt vervolgens op basis van deze linguïstische omschrijvingen en zal bijgevolg linguïstische stuurwaarden genereren. Deze moeten weer omgezet worden in 'harde' numerieke stuurwaarden (*defuzzificeren*). Zowel het fuzzificeren als het defuzzificeren vindt plaats door mathematische relaties, dus zo 'vaag' zijn deze processen niet.

In Fig. 2.4 wordt enige typische Fuzzy Logic terminologie geïntroduceerd. De term 'crisp' (input en output) betekent in dit verband 'scherp', zie ter illustratie Fig. 2.5a. Hier zie je voor een temperatuuraanduiding dat het tussen 18 en 27 graden 'aangenaam' is, en daarbuiten is het 'niet-aangenaam'. Dit is in feite een binaire aanduiding: het is 100% 'aangenaam' of het is 0% 'aangenaam'. De Fuzzy tegenhanger wordt geïllustreerd in Fig. 2.5b door de zogenoemde membershipfunctie 'aangenaam', waarin de relatie wordt gegeven tussen temperatuur en de mate van aangenaamheid.

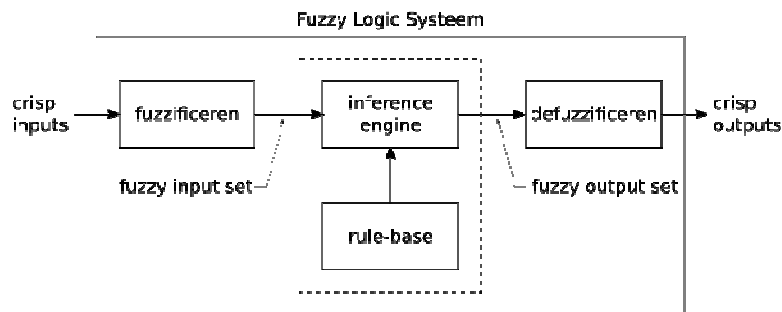


Fig. 2.4 Fuzzy Logic Systeem

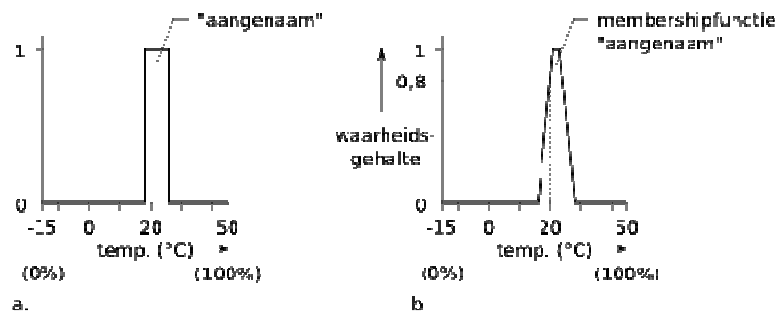


Fig. 2.5 Crisp versus Fuzzy

De vorm van een membershipfunctie hoeft zeker niet altijd de vorm van een trapezoïde te hebben. In Fig. 2.6 zijn als voorbeeld de membershipfuncties ‘koud’, ‘fris’, ‘aangenaam’, ‘warm’ en ‘heet’ weergegeven. Deze set functies zou een typische ‘fuzzy input set’ kunnen zijn. Evenzo dient er een ‘fuzzy output set’ te worden gedefinieerd, bijvoorbeeld de set ‘stevig verwarmen’, ‘verwarmen’, ‘neutraal’, ‘koelen’, ‘stevig koelen’.

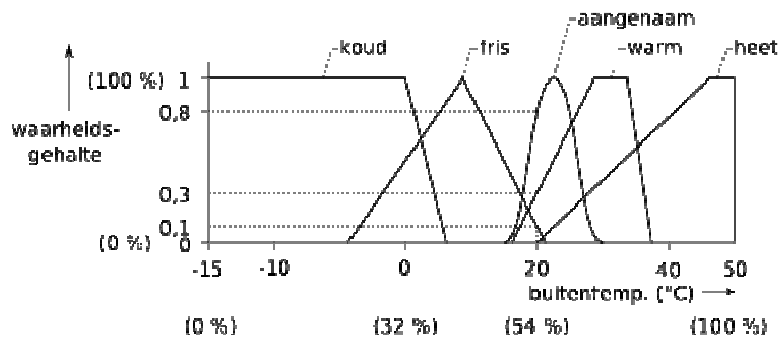


Fig. 2.6 Voorbeelden van membershipfuncties

Vorm en plaats van de membershipfuncties worden in feite door de ontwerper van het Fuzzy Logic Systeem bepaald.

Het *fuzzificeren* van de harde signalen vindt plaats aan de hand van membershipfuncties uit de input set. Uit bovenstaand voorbeeld blijkt dat het bij 20°C voor 10% ‘fris’, voor 80% ‘aangenaam’ en voor 30% ‘warm’ is. ‘koud’ en ‘heet’ scoren 0%. (Sommigen vinden 20° fris, velen aangenaam en anderen weer warm, niemand vindt 20° koud of heet – volgens de ontwerper van dit systeem).

De *rule-base* in Fig. 2.4 is de verzameling ‘if-then’-statements waarmee van de gefuzzificeerde ingangssignalen van de regelaar de fuzzy uitgangssignalen worden gemaakt. Het zal duidelijk zijn dat elke statement uit de rule-base een fuzzy waarde van een of meer uitgangssignalen kan opleveren; dit manifesteert zich in de vorm van een bepaald waarheidsgehalte van een of meer membershipfuncties van elk uitgangssignaal. De rule base dient een afdoende hoeveelheid statements te bevatten zó dat de regelaar goed functioneert. In de rule base opnemen van statements van situaties die zich niet zullen voordoen dient vermeden te worden.

Fuzzy inference (gevolgtrekking, conclusie) is in elk fuzzy geregeld systeem de op één na laatste stap die in de fuzzy regelaar wordt uitgevoerd. Dat wil zeggen dat van alle membershipfuncties van alle uitgangssignalen van de regelaar (dit zijn dus de stuursignalen voor het proces) wordt bepaald hoe groot hun uiteindelijk aandeel is in het betreffende uitgangssignaal.

Defuzzificeren ten slotte is de laatste stap die wordt uitgevoerd in de fuzzy regelaar. Alle fuzzy stuurwaarden van de regelaar worden omgezet in een numerieke waarde waarmee het proces daadwerkelijk kan worden aangestuurd. Deze operatie vormt dus de interface tussen de uitgang van de fuzzy regelaar en de ingang van het proces. Een veel gebruikte methode hierbij is zwaartepuntmethode.

3. TOEGEPASTE METHODE / EXPERIMENTELE SETUP

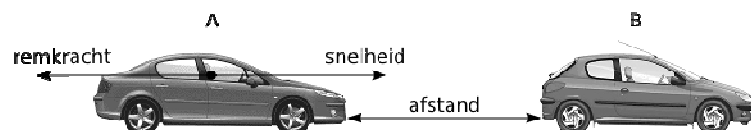


Fig. 3.1 Schematische weergave experiment

De situatie zoals die in het simulatie experiment (zie Fig. 2.1, Fig. 3.1) is uitgevoerd betreft een systeem waarbij 2 auto’s op een recht traject achter elkaar aan rijden. De twee auto’s zijn achtereenvolgens PreScan® modellen van een Fiat Bravo (auto B in Fig. 3.1) en een Citroen C3. Elk van deze auto’s heeft een in de modelleromgeving vastgelegde (eenvoudige) voertuigdynamica. De Fiat Bravo rijdt volgens een bepaald gewenst snelheidsprofiel (zie Fig. 3.2), dit is in feite een reeks setpoints als functie van de tijd. De Citroen probeert de Fiat op gepaste afstand (bij stilstand 10 meter) te volgen. Afstand tot de voorganger en relatieve snelheid ten opzichte van de voorganger wordt gemeten met behulp van een AIR-sensor (Actor Information Receiver, niet gebonden aan bepaalde technologie, komt uit PreScan®). Vanuit stilstand vertrekt de Fiat Bravo, en accelereert tot de 50 km/h is bereikt. Na 50 seconden accelereert de Fiat verder tot 60 km/h. Enzovoorts.

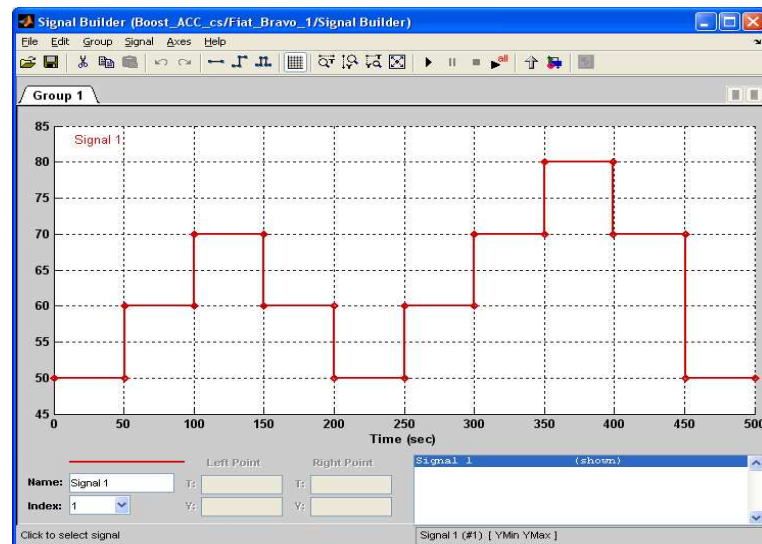


Fig. 3.2 Gewenst snelheidsprofiel van het experiment

De regeling van de Fiat Bravo is bewust eenvoudig gehouden en is te vinden in Fig. 3.3. Linksboven in deze figuur staan de sensorgegevens zoals deze uit de gesimuleerde Fiat Bravo komen. Uit dit blok zijn de volgende parameters te halen: positie in XYZ-coördinaten (m) en de xyz-rotatie (graden) van het voertuig in het assenstelsel van het experiment, de GPS-coördinaten, de absolute snelheid (m/s) en de kompas heading (graden). Voor de eenvoud nemen we in ons model alleen de snelheid als parameter mee, daar we deze willen vergelijken met de gewenste snelheid en vervolgens daarop de voertuigregeling afstemmen. De feitelijke regeling vindt plaats in het Fuzzy Logic Control blok. Deze heeft twee inputs: het snelheidsverschil en de verandering in het snelheidsverschil. Output van de regelaar is een maat voor de hoeveelheid gas (Throttle %) welke gegeven wordt. Dit gaat via enig signaalnivellering het 'Simple_Dynamics' blok van de gesimuleerde Fiat in. Mogelijke ingangen bij dit Simple_Dynamics blok zijn: stuurhoek (graden), gaspedaal (%), remdruk (bar), versnelling (auto of hand). In ons model zijn de automodellen automaten en laten we ze rechtdoor rijden. Remmen doen we niet, alleen gas geven (ahum).

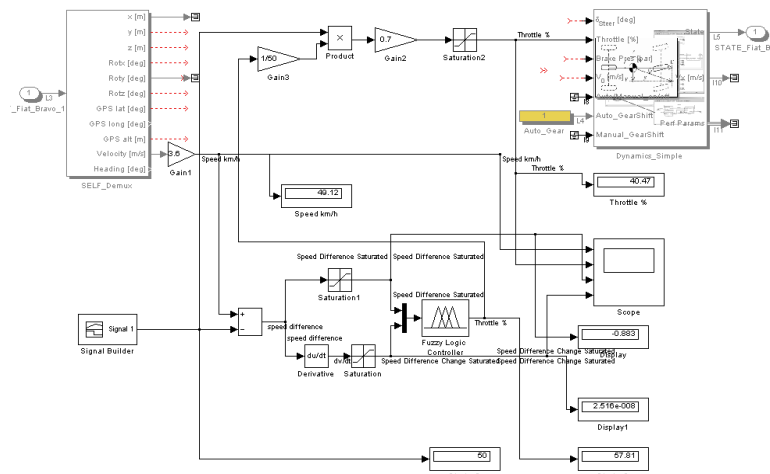


Fig. 3.3 De Fuzzy Logic Controller toegepast in de regeling van de Fiat Bravo

Het Matlab/Simulink model van de regeling in de volgauto is vrijwel identiek aan de regeling in de Fiat, met als grootste verschil de input vanuit de AIR-sensor. Het blok van de AIR sensor geeft de volgende waarden van het gedetecteerde object: afstand (m), azimuth (graden) en elevation (graden), snelheid (m/s), heading (graden). Van deze inputs willen we in dit eenvoudige model alleen snelheid van het voorgaande object (de Fiat) weten. Zie Fig. 3.4.

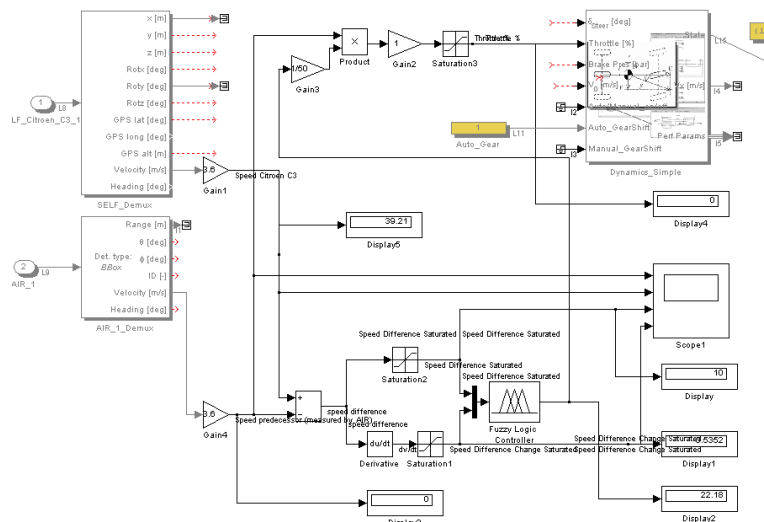


Fig. 3.4 De Fuzzy Logic Controller toegepast in de regeling van de Citroen C3

Het instellen van de Fuzzy Regelaar in Matlab gaat in een aantal stappen. De tool die daartoe gebruikt dient te worden is de Fuzzy Inference System (FIS) editor (zie Fig. 3.5). In de editor kan via [Edit] extra inputs en outputs worden toegevoegd. De benamingen van de inputs en outputs kunnen naar wens worden aangepast. Ook kan aangegeven worden hoe de operatoren zoals 'and' en 'or' uit de rule-base geïnterpreteerd dienen te worden. We gebruiken de default instellingen.

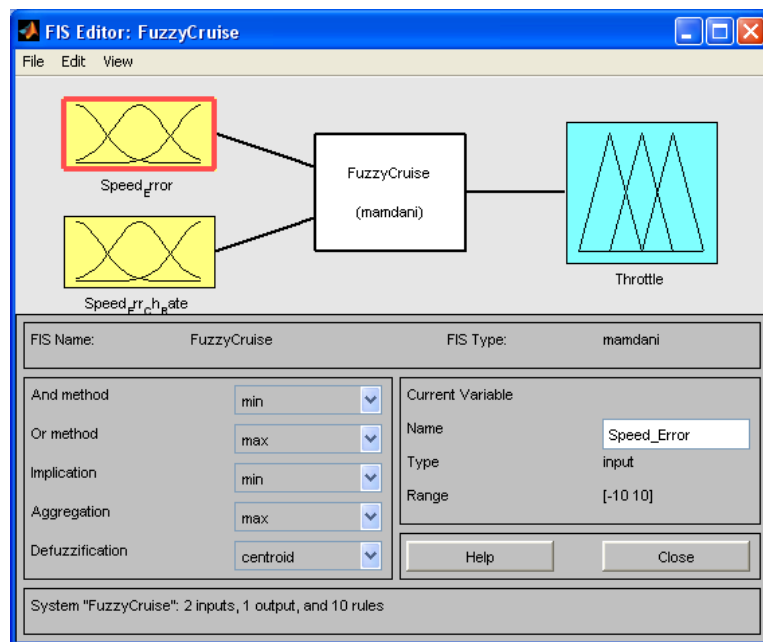


Fig. 3.5 De FIS editor van Matlab

Door dubbelklikken op het plaatje van de membershipfunctie speed_error wordt het venster van Fig. 3.6 geopend. Met de kleine icoontjes van de membershipfuncties linksboven in het scherm kan de te wijzigen input- of outputvariabele worden gekozen (zie ook Fig. 3.7 en Fig. 3.8). Naam en vorm van elke membershipfunctie kan in dit scherm naar wens worden aangepast. Het kiezen van goede benamingen is handig bij het vullen van de rule-base in de rule-editor.

Uit het simulink model valt af te leiden dat een negatieve speed_error betekent dat de actuele snelheid lager ligt dan de gewenste snelheid. Negatieve speed_error_change_rate betekent het lager worden van het snelheidsverschil.

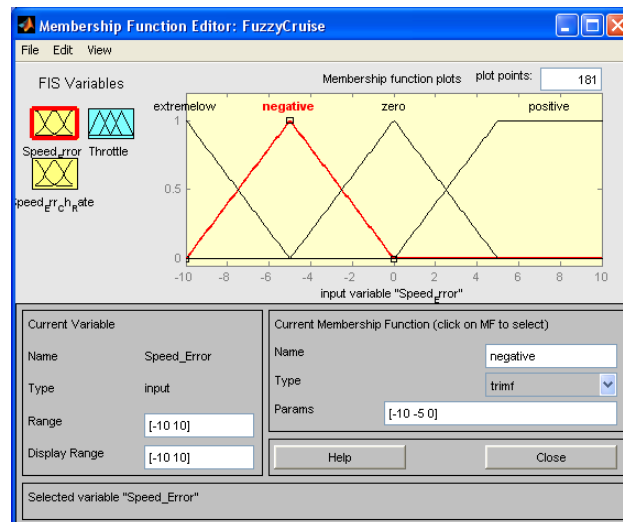


Fig. 3.6 Membership functions speed_error

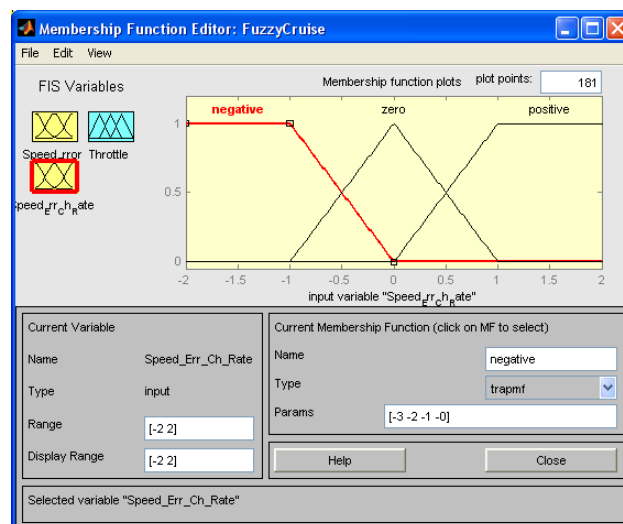


Fig. 3.7 Membership functions speed_error_change_rate

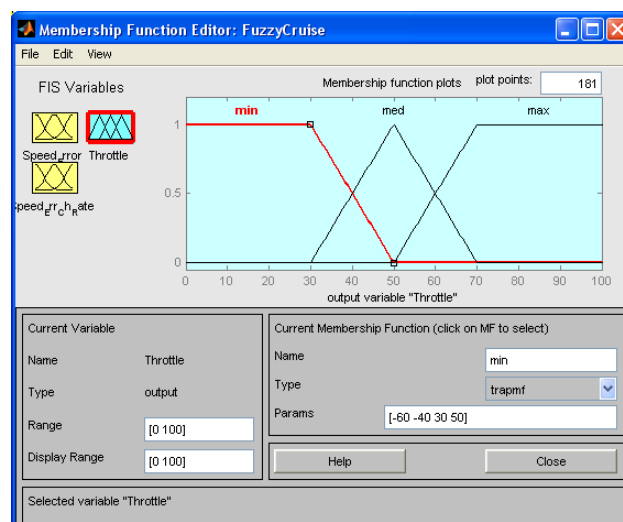


Fig. 3.8 Membership functions Throttle

Nadat alle membershipsfuncties zijn gedefinieerd kan de rule-base worden gevuld. Dubbelklikken op het de middelste figuur in de FIS-editor (Fig. 3.5) opent de rule-base editor (Fig. 3.9). Natuurlijk dient van te voren goed over de rules nagedacht te worden. Zo zal, indien de speed_error 'extreme low' is (ofwel de actuele snelheid is flink langzamer dan de gewenste), de gaspedaal ver open gezet moeten worden teneinde het snelheidsverschil kleiner te maken.

Zodra de rule-base is opgezet kunnen de instellingen worden geëxporteerd naar file. Deze file dient vervolgens te worden geïmporteerd in het Fuzzy Logic Control blok uit het Simulink model.

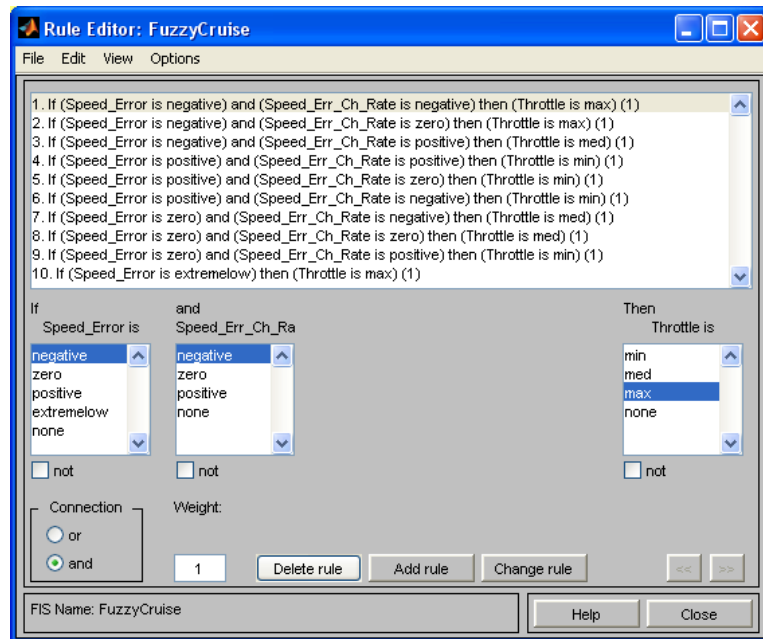


Fig. 3.9 De rule-base editor

De ingevoerde rules zijn op een tweetal manieren te inspecteren. Eén van die mogelijke manieren is via het rule-viewer scherm (Fig. 3.10). In dit scherm is snel de invloed van de inputs op de output te beoordelen. In het scherm is de speed_error op 1.6 [km/h] ingesteld en de speed_error_changerate op 0.5 [km/h/s] ingesteld. Het throttle% wordt dan 32.3%. Het instellen van de inputs kan met de muis door de verticale lijnen te verslepen of onderaan in het tekstveld in te voeren. Alle wijzigingen van de inputs zijn direct in de output te zien. De mate, per rule, waarin een input in de output mee doet is goed te zien.

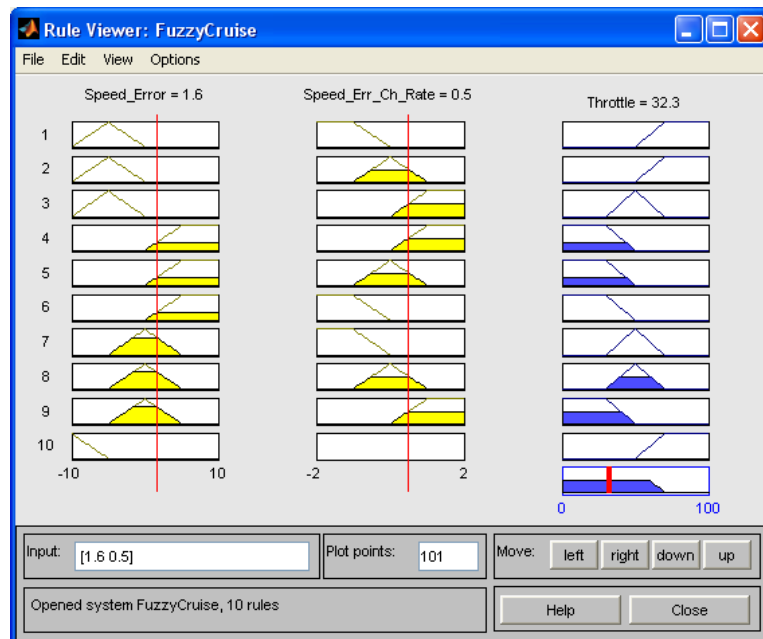


Fig. 3.10 De rule viewer

Naast deze manier van inspectie is het eveneens mogelijk om een surface view te genereren. Op de 'onafhankelijke' assen zijn dan inputs te kiezen, en op de 'afhankelijke' as een output. Met onze instellingen verkrijgen we de surface view uit Fig. 3.11. Hierin is goed te zien dat bijvoorbeeld bij een speed_error van -10 km/h (minimale, begrensde waarde in het model) de throttle maximaal wordt aangestuurd.

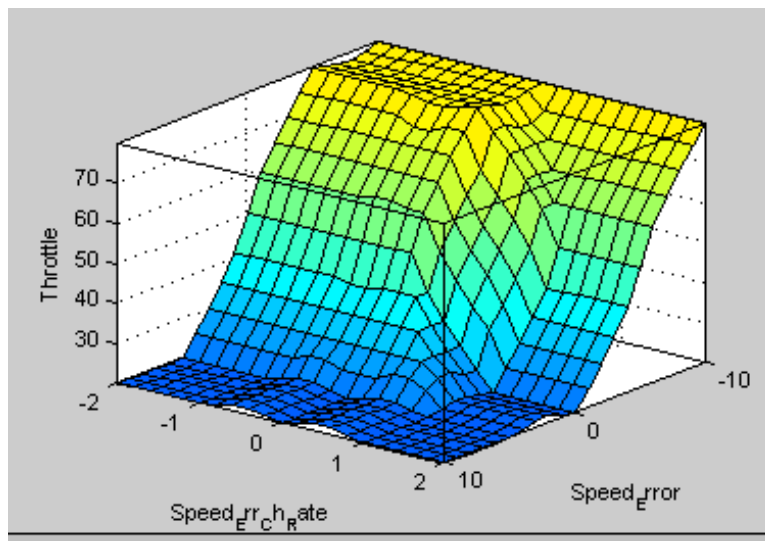


Fig. 3.11 De rule viewer in surface view

4. RESULTATEN

In Fig. 3.2 staat het gewenste snelheidsprofiel gegeven van de te volgen auto, de Fiat Bravo. Fig. 4.1 toont van boven naar beneden achtereenvolgens de snelheid, het throttle percentage, het (begrensd) verschil tussen de actuele en de gewenste snelheid en de (begrensd) verandering in het snelheidsverschil. De laatste twee zijn de inputs voor de Fuzzy Control Logic en de tweede is de output naar het voertuigmodel toe.

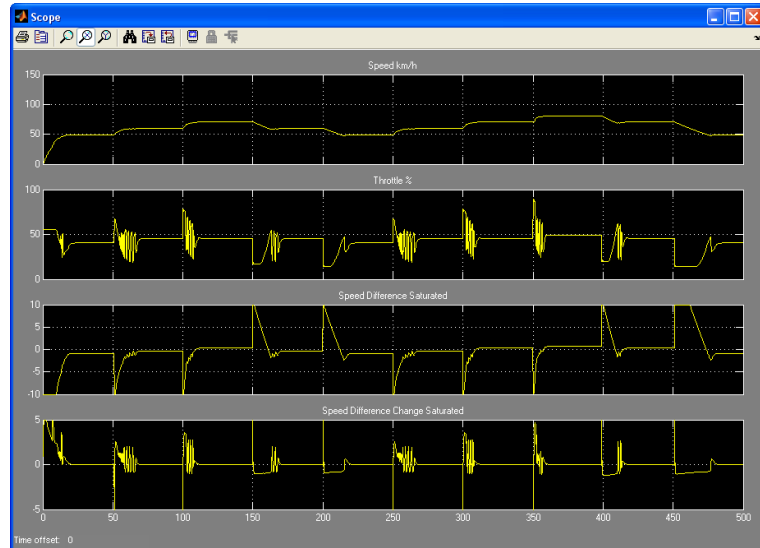


Fig. 4.1 Meetgegevens van de Fiat Bravo

Van het control model van de volgauto (de Citroen C3) zijn in Fig. 4.2 achtereenvolgens de volgende grootheden zichtbaar gemaakt: door de AIR sensor gemeten snelheid van de Fiat Bravo, de eigen snelheid, het snelheidsverschil (begrensd), de (begrensd) verandering in het snelheidsverschil en de door de AIR sensor gemeten afstand (meter) tot de Fiat.

Gedurende een groot deel van de simulatietijd weet de Citroen de Fiat op gepaste afstand te volgen. Op zeker moment begint de AIR sensor vreemd gedrag te vertonen, alsof de AIR sensor op de Citroen de Fiat niet meer ziet.

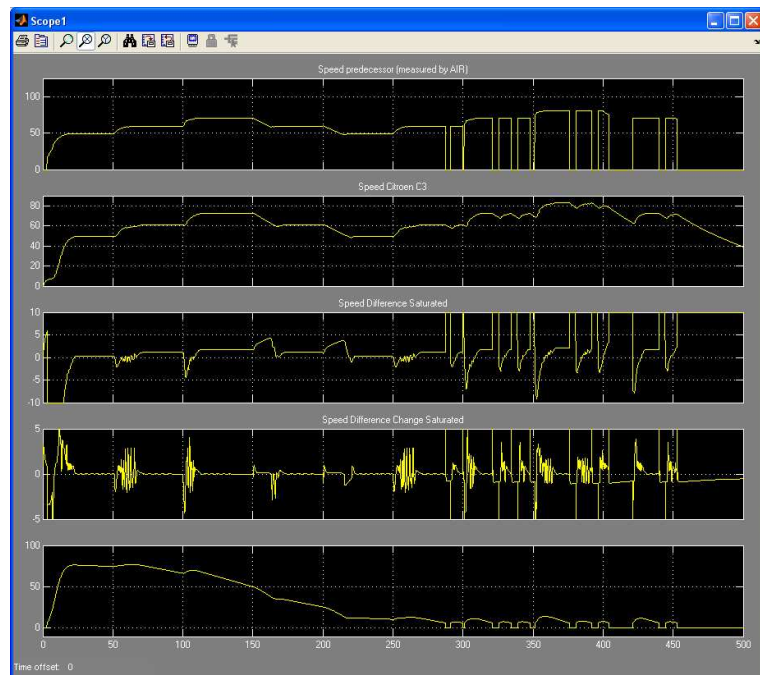


Fig. 4.2 Meetgegevens van de volgauto Citroen C3

5. DISCUSSIE

Er zijn in dit (eenvoudige) experiment vele parameters waarmee gevarieerd kan worden. Naast het model voor de voertuigbesturing is dat natuurlijk de rijkelijke keuze die er ligt bij het vastleggen van de membershipfuncties van de Fuzzy Logic Control. Er is voor een bepaalde set membershipfuncties in combinatie met een rule-base gekozen welke redelijk aannemelijk leek. Het onderzoek heeft niet de intentie gehad een uitputtende zoektocht te zijn naar alle mogelijke combinaties van membershipfuncties en rule-bases voor de inputs en output van de FLC en de effecten daarvan op het gedrag van de voertuigen.

Zowel de Fiat als de Citroen worden 'bestuurd' door een Fuzzy Logic Controller met identieke parameters. Welbeschouwd zou deze parametrisatie verschillend moeten zijn daar de voertuigtypen niet identiek rijgedrag hebben. Desondanks weet enerzijds de Fiat Bravo het opgelegde snelheidsprofiel (Fig. 3.2 vs Fig. 4.1 a) goed te volgen en weet anderzijds de Citroen C3 de snelheid van de Fiat goed te volgen (Fig. 4.2 a, b). Daarnaast weet de Citroen C3, na een aanvankelijke achterstand van circa 70 meter, deze te verminderen tot de in het model gewenste ingestelde achterstand van 10 meter (Fig. 4.2 e). Bij 72 km/h komt deze afstand overeen met een tijd van 0.5 sec tussen de voertuigen.

Op zeker moment (op $t=287s$) begint de AIR-sensor een niet te verklaren gedrag te vertonen. Het lijkt alsof de sensor de Fiat Bravo niet meer waarneemt. Een eerste gedachte hierbij was dat de sensor de Fiat uit het oog verloren was, de sensor heeft namelijk een maximaal bereik van 100 m, maar na toevoegen van de afstand-output van de AIR-sensor aan de meetgegevens lijkt dit uitgesloten.

6. CONCLUSIE EN AANBEVELINGEN

Er mag worden gesteld dat de technologie Fuzzy Control Logic in principe toepasbaar is in Adaptive Cruise Control. Immers de volgauto weet de voorste auto bij te houden, en de voorste auto weet het opgelegde snelheidsprofiel te handhaven.

De simulatie omgeving *PreScan*[®] icm *Simulink*[®] is behoorlijk pittig. Enerzijds is dat *PreScan*[®] met een immense hoeveelheid aan mogelijkheden en anderzijds het modelleren van de voertuig control in *Simulink*[®].

Qua onderzoek kan worden gesteld dat in de regeling de afstand tot de voorganger niet is meegenomen. En dat de rem niet bekrachtigd kan worden. Deze beide factoren meenemen maakt het model, en ook de Fuzzy Logic Control behoorlijk complexer en zal ook de rule-base doen toenemen. Naar verwachting zal het volggedrag dan nog beter werken en hopelijk verdwijnt dan ook het geobserveerde vreemde gedrag van de AIR-sensor.

In de GCDC wordt uitgegaan van Cooperative Adaptive Cruise Control. Dit hebben we niet onderzocht, immers hiertoe dienen de voertuigen actief informatie uit te zenden (meestal acceleratie) en dat is niet meegenomen in het model. *PreScan*[®] / *Simulink*[®] biedt deze mogelijkheid echter wel.

Laatste punt dat uitgezocht kan worden in een mogelijk vervolgonderzoek betreft het naast elkaar zetten van CACC modellen met enerzijds een klassieke regeling en anderzijds een FLC regeling.

REFERENTIES

- [1] Grand Cooperative Driving Challenge. De onderdelen.
http://www.gcdc.net/mainmenu/Home/Challenging_aspects
- [2] M. Glaskin, "Shockwave traffic jam recreated for first time". <http://www.newscientist.com/article/dn13402>
- [3] Elaine Shaw and J. Karl Hedrick, "String Stability Analysis for Heterogeneous Vehicle Strings" in *Proceedings of the 2007 American Control Conference*, New Orleans, USA, December 2007
- [4] G. Naus, R. Vugts, J. Ploeg, R. Vd Molengraft and M. Steinbuch, "Towards on-the-road implementation of cooperative adaptive cruise control".
- [5] G. Naus, R. Vugts, J. Ploeg, R. Vd Molengraft and M. Steinbuch, "String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach" in *IEEE transactions on vehicular technology*, vol. 59, no. 9, november 2010.
- [6] Zadeh, L.A. (1965). "Fuzzy sets", *Information and Control* 8 (3): 338–353
- [7] TNO, "*PreScan*[®]", <http://www.tno.nl/prescan>

Simulatie software

PreScan:

www.tno.nl/prescan

Modeleren en simuleren met MatLab Simulink:

www.mathworks.com

Fuzzy control systems

http://en.wikipedia.org/wiki/Fuzzy_control_system

Relevante Youtube filmpjes via www.youtube.com:

Filmpje over VeHIL laboratorium TNO:

Trefwoorden "TNO vehil hoezo teleac"

Filmpje over Japanse string stabiliteit

Trefwoorden "shockwave traffic jams first time"

Filmpje over test A-CC en C-ACC Tue en TNO

Trefwoorden "string stability tests"

Geraadpleegde / geciteerde literatuur

- Boek: Regeltechniek voor het HTO deel 1 analoge systemen
Auteurs: Jaap Schrage, Hans van Daal, Jan Stroeken
Uitgever: HB uitgevers
ISBN: 90-5574-275-9
- Paper: Fuzzy Logic Vehicle Intelligent Cruise Control Simulation
Auteur: Ing. Ondřej Láník
Czech Technical University in Prague, Faculty of Mechanical Engineering
- Website: Fuzzy Logic for "Just Plain Folks"
URL: <http://www.fuzzy-logic.com/>
Auteur: Thomas E. Sowell, Published on the Web 2003, revised September, 2008.
- Boek: The Fuzzy Systems Handbook – A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems
Auteur: Earl Cox
Uitgever: Academic Press, Inc.
ISBN: 0-12-194270-8
- Boek: Fuzzy Control
Auteurs: Kevin M. Passino and Stephen Yurkovich
Uitgever: Addison Wesley
ISBN: 0-201-18074-X
- Fuzzy Logic Systems for Engineering - A Tutorial
Proceedings of the IEEE 0018-9219/95\$04.00 (c) 1995 IEEE
Jerry M. Mendel
- Boek: Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems
Auteurs: Guanrong Chen and Trung Tat Pham
Uitgever: CRC Press
ISBN: 0-8493-1658-8
- Boek: Fuzzy sets and fuzzy logic: theory and applications
Auteurs: George J Klir and Bo Yuan
Uitgever: Prentice Hall
ISBN: 0-13-101171-5