

Project vage databanken

Deel I: ontwikkeling van een generiek vaag regelsysteem

Voor het eerste deel van het project vage databanken is het de bedoeling een vaag regelsysteem te ontwikkelen. Vage regelsystemen worden gebruikt voor procesregelingen door gebruik te maken van vage regels. Bij dergelijke regels kunnen zowel de condities als de beslissing vaag geformuleerd worden. Een algemene vage regel kan in zijn normaalvorm geschreven worden als:

$$ALS \left(\bigwedge_{i=1}^n \bigvee_{j=1}^m (X_{ij} \text{ IS } L_{ij}) \right) \text{ DAN } Y \text{ IS } L$$

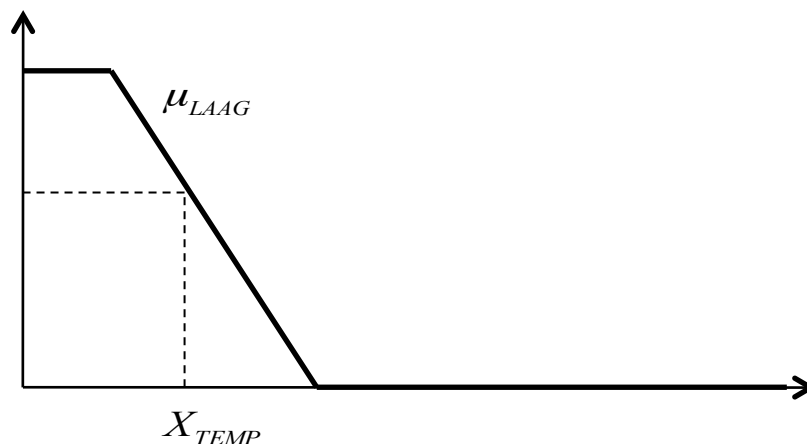
Hierbij zijn X_{ij} en Y gewone variabelen en zijn L_{ij} en L linguïstische termen voorgesteld door vaagverzamelingen. Een eenvoudig voorbeeld van een vage regel met twee condities voor het aansturen van een verwarmingstoestel is:

ALS (TEMPERATUUR IS LAAG) EN (DRUK IS LAAG) DAN GASKRAAN IS OPEN

Het nut in complexe procesregeling van vage regels wordt gehaald uit het feit dat een scherpe waarde van een variabele verschillende regels in verschillende mate kunnen geactiveerd worden. Er wordt met vage regels vermeden dat een klein verschil in de in te lezen parameters ongewenst grote gevolgen heeft. Het evalueren van een collectie vage regels is typisch een vijfstappen proces.

1. Evaluatie van de condities

Wanneer een variabele voorkomt in een conditie, wordt de waarde van die variabele aangelegd aan de lidmaatschapsfunctie van de linguïstische term uit die conditie en wordt de lidmaatschapsgraad gemeten. Voor de conditie 'temperatuur is laag' wordt dit getoond op Figuur 1. In dit project mag verondersteld worden dat lidmaatschapsfuncties die voorkomen in de specificaties van regels steeds een puntsgewijze vorm hebben. Opgelet, denk goed na over de implementatie van de lidmaatschapsfuncties en hou ook rekening met verdere berekeningen.



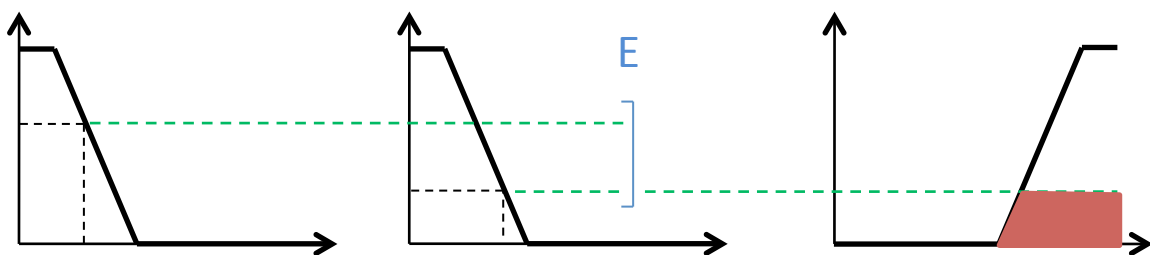
Figuur 1: evaluatie van de conditie 'Temperatuur is Laag'

2. Aggregatie van de condities

Wanneer een regel meerdere condities bevat, dan moeten de lidmaatschapsgraden die verkregen zijn na evaluatie van de condities geaggregeerd worden. In de vage logica wordt een t-norm gebruikt voor conjunctie en een t-conorm voor disjunctie. Binnen dit project mag u zich beperken tot (min,max), het probabilistische paar en het Lukasiewics paar. U mag veronderstellen dat binnen eenzelfde regel, steeds hetzelfde paar van t-norm en t-conorm gebruikt wordt.

3. Bepaling van de regel output

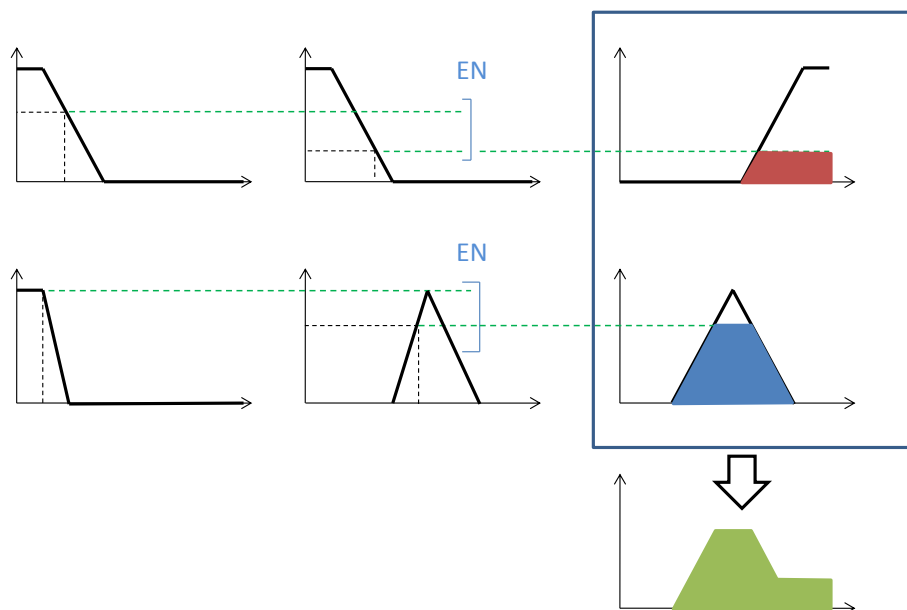
Na aggregatie van condities is voor elke regel een geaggregeerde lidmaatschapsgraad bekend die aangeeft in welke mate de regel van toepassing is. Met dit getal wordt de lidmaatschapsfunctie behorend bij het gevolg van de regel afgesneden. Dit wordt getoond op Figuur 2.



Figuur 2: bepaling van de regel output

4. Samenvoeging van de regels

Wanneer meerdere regels aanwezig zijn die eenzelfde variabele aansturen, dan moeten deze regels samengebracht worden. Deze operatie wordt gedaan door de verkregen vaagverzamelingen te combineren met een unie operator. Als implementatie wordt het puntsgewijze maximum van lidmaatschapsfuncties gebruikt. Dit wordt getoond in Figuur 3.



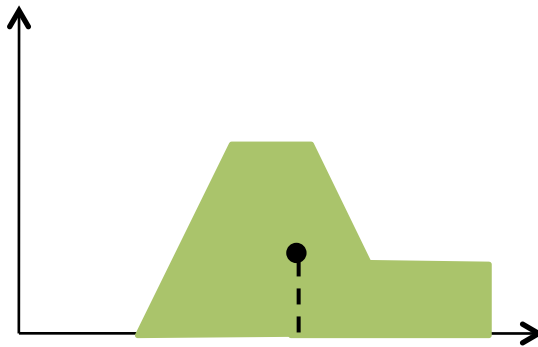
Figuur 3: combinatie van regels

5. Verscherping

In de laatste stap wordt voor elke aan te sturen variabele een scherpe waarde berekend uit de verkregen vaagverzameling. Een veelgebruikte methode voor verscherping in de context van vage regelsystemen, is het berekenen van de x-coördinaat van het zwaartepunt van het oppervlak onder de verkregen lidmaatschapsfunctie. De x-coördinaat van het zwaartepunt van het oppervlak onder een functie $f(x)$ wordt berekend als:

$$x_Z = \frac{\int_{-\infty}^{\infty} x \cdot f(x) \cdot dx}{\int_{-\infty}^{\infty} f(x) \cdot dx}$$

Dit wordt getoond in Figuur 4.



Figuur 4: verscherping van de regel output

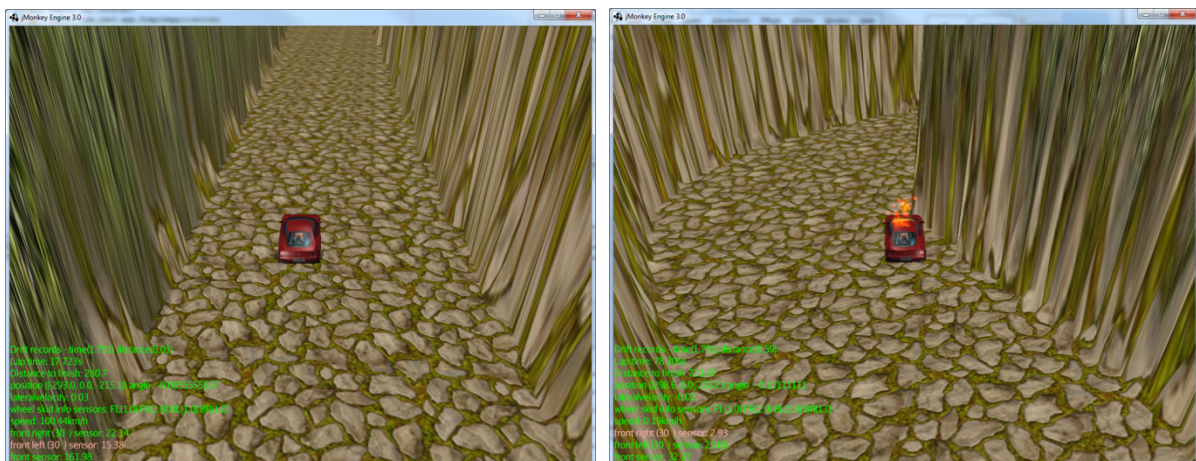
Deel II: ontwerp van een vage controller

In het tweede deel van dit project, wordt gevraagd een vage controller te ontwerpen voor de aansturing van een racewagen. De wagen kent aanvankelijk een vastgelegde startplaats en moet zo snel mogelijk een ronde afleggen op de gegeven circuits. Dit begint eenvoudig op de Texas Speedway, maar de auto moet ook rond geraken op deftige circuits zoals Spa Francorchamps.



Figuur 5: Texas Speedway (links) - Spa Francorchamps (rechts)

Voor de rendering van dit project maken we gebruik van de JMonkey Game Engine en een omgeving die door hen voorzien wordt. Dit stelt ons in staat een realistischere omgeving te voorzien waarbij de wagen slipt, wheelspin heeft, kan crashen, kan bijgestuurd worden door tegen te sturen etc.



Figuur 6: Race in Jmonkey Omgeving (links) - crash (rechts)

U krijgt bij deze opgave enkele parcours (Interlagos, Silverstone, Spa Francorchamps, Texas Speedway) ter beschikking en een jar die deze bevat alsook de jmonkey engine en alle nodige bibliotheken. Deze vereist OpenGL2.0. Indien uw pc hier problemen mee heeft kan u terecht in de pc zalen van de plateau.

Er wordt verwacht dat u een (of meerdere) klasse(s) schrijft die de interface Controller implementeert (implementeren). De methode **getFrameControl(VehicleProperties vp)** is de enige te implementeren methode en moet een Framecontrol object terug geven **new FrameControl(steering, acceleration, brake)** waarin de gebruiker op basis van de huidige vehicleproperties aangeeft in hoeverre mate dat hij wil dat de wagen versnelt (acceleration 0...1600), remt (brake 0...40) en welke draaihoek hij moet proberen nemen (steering -1...1 , voorstelling van 90 graden naar links of rechts).

De vehicleproperties die ter beschikking staan zijn:

- private float carSpeedKph; (0 - 500)
- Sensoren centrum, links en rechts in vooraf in te stellen hoek in framecontrol
 - private float frontSensorDistance;
 - private float leftSensorDistance;
 - private float rightSensorDistance;
- private float roadWidth;
- frictie van de wielen (indien kleiner dan 0.1 begin je redelijk te slippen) :
 - private float frontLeftFriction;
 - private float backRightFriction;
 - private float backLeftFriction;
 - private float frontRightFriction;
- private float lateralVelocity; (stelt u in staat te meten wanneer auto effectief begint te slippen wanneer deze afwijkt van 0)
- private float frontwheelangle; (geeft de hoek van 1 van de voorwielen terug)

Het is de bedoeling aan de hand van deze methode enkele vage regelsystemen te implementeren om zo dus meerdere vage controllers te bekomen.

- Een controller die de focus legt op zo snel mogelijk de finish bereiken
- Een controller die de prioriteit op veiligheid legt en dus er voor moet zorgen dat de wagen niet crasht.
- Een controller die bochten rally style benadert: insturen met oversteer zodat het achterste van de wagen te ver gaat en je moet tegen sturen om de bocht te halen. (wat ook zonder handrem mogelijk is)

Voorbeeldje met een van de zaligste old school rally cars ooit gemaakt, de E30:

<http://www.youtube.com/watch?v=i8jEYINrCjQ>

Het is de bedoeling om bovenstaande controllers algemeen te ontwikkelen zodat deze op alle parcours werken, maar het is ook toegestaan extra verschillende setups (een per parcours) voor de wagen (andere controllers) te schrijven zodanig dat de wagen competitiever is.

Voer enkele experimenten uit met uw controllers die u een antwoord geven op de volgende vragen.

- Is de keuze van t-norm en t-conorm belangrijk voor de prestatie (i.e. de gemiddelde snelheid) van de wagen?
- Hoe robuust is uw regelsysteem voor verschillende races? Is de gemiddelde snelheid ongeveer dezelfde als u de wagen verschillende keren laat racen op een welbepaald parcours?
- Zijn er parcours waar uw controller minder goed presteert? Hoe komt dit?
- Hoe veilig is de wagen, i.e. hoe dikwijls crasht uw wagen?
- Wat zijn de verschillen met een exacte controller (beter/slechter)

Op <http://telin.ugent.be/~jnieland/scores.php> kan u uw rondetijden terugvinden, alsook drift tijden en afstanden, en de som van de totale impacten van de wagen.

Deel III: Verslag

Schrijf bij dit project een uitgebreid en verzorgd verslag waarin u de controllers (en dus de regelsystemen) voor de wagen volledig beschrijft alsook hoe performant deze controllers zijn op de verschillende parcours. Geef in dit verslag ook een antwoord op de gestelde vragen voor de drie algemene controllers. Geef duidelijk aan wanneer u andere bibliotheken hebt gebruikt. Bundel een pdf document van uw verslag, gegevens van uw groep en uw code (enkel de controllers en gebruikte extra bibliotheken) in een zip bestand en plaats dit op Minerva via de dropbox. U mag werken in groepen van twee studenten. De uiterste indiendatum voor het verslag is zondagavond 23:59, 31 mei 2015.