

Projektdokumentation

„Ampelsteuerung“

in der Lehrveranstaltung

Wissensbasierte Systeme

SS06



Name	Vorname	Matrikelnummer
Plüß	Martin	1xxxxx
Göbel	Marco	9xxxxx
von der Brüggen	Marc	9xxxxx

Inhaltsverzeichnis

1. Projekt & Ziel	3
1.1 Projektbeschreibung	3
1.2 Problembeschreibung	3
1.3 Zieldefinition	4
2. Umsetzung	5
2.1 Eingangsvariablen	5
2.1.1 Dichte	5
2.1.2 Wartezeit Fußgänger	6
2.1.3 Wartezeit Fahrzeuge.....	7
2.1.4 Besonderes Fahrzeug.....	7
2.2 Ausgangsvariablen	8
2.2.1 Grünphase Ampel.....	8
2.2.2 Dauer Grünphase.....	9
2.2.3 Dauer Gelbphase	9
3. Einstellungen des Fuzzy-Reglers	10
3.1 Fuzzyfizierung.....	10
3.2 Inferenz.....	10
3.3 Defuzzifizierung	11
4. Darstellung der Variablen als Surfaces	12
4.1 Grünphase Ampel (Besonderes Fahrzeug1 und Besonderes Fahrzeug2)	12
4.2 Dauer der Grünphase (Straße1 und Wartezeit2)	13
5. Fazit	14
6. Anhang	15

1. Projekt & Ziel

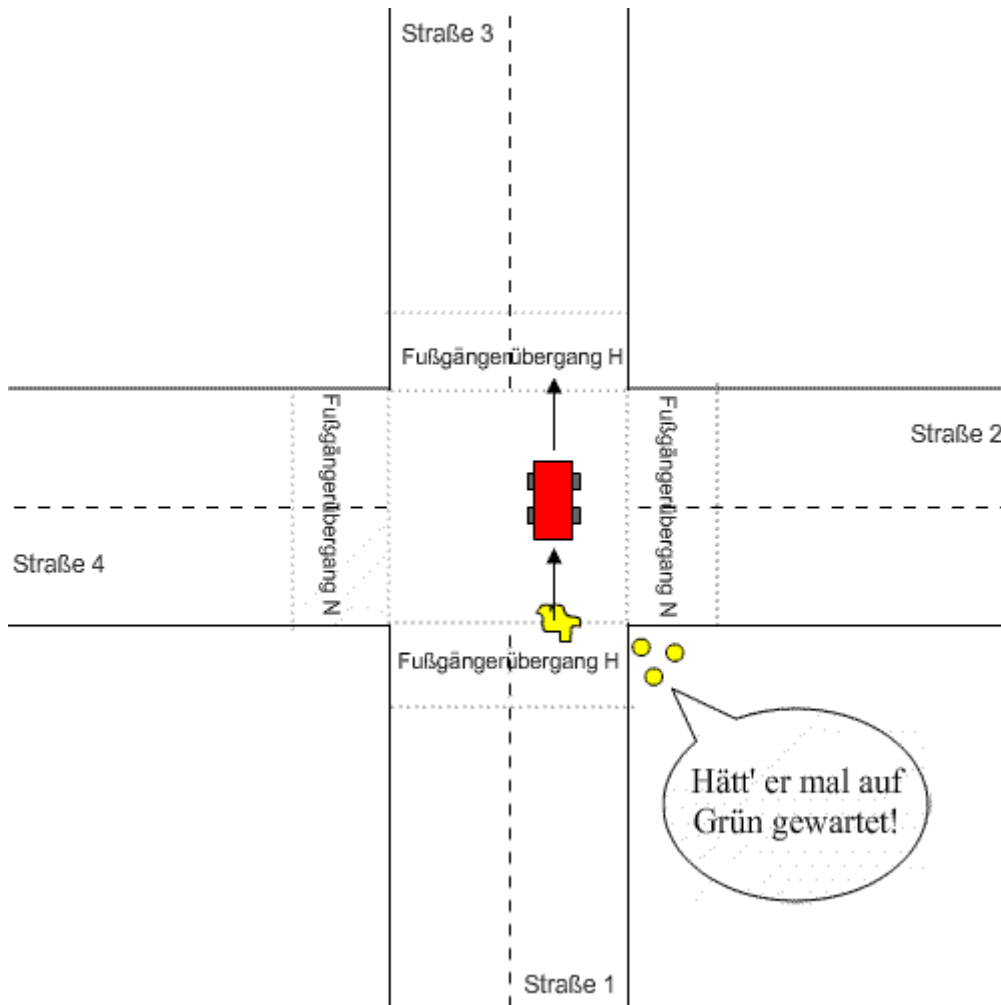
1.1 Projektbeschreibung

Innerhalb unserer Lehrveranstaltung Wissensbasierte Systeme, wurde uns die Aufgabe gestellt, ein auf Fuzzy-Logik basierendes Steuerungssystem zu entwickeln. Wir haben uns dazu entschlossen, diese Aufgabenstellung in Form einer, auf einem Fuzzy-Regler basierende Verkehrsampelsteuerung, zu realisieren.

1.2 Problembeschreibung

Unsere Ampelsteuerung hat folgende Parameter zu beachten: Sie reguliert den Verkehr auf einer vierspurigen Kreuzung mit Fußgängerampeln. Dabei handelt es sich um eine Hauptstraße, welche das größte Verkehrsaufkommen besitzt und zwei Nebenstraßen. Jede dieser Straßen, kann, reguliert durch eine Fußgängerampel, von Fußgängern überquert werden. Dabei sind die beiden, sich gegenüberliegenden Fußgängerampeln parallel geschaltet (immer gleichzeitig rot oder grün). Eines, der zu beachtenden Kriterien ist, die Dichte (Anzahl) der wartenden oder fahrenden Fahrzeuge pro Straße. Sowohl bei den Straßen, als auch bei den Fußgängerampeln wird die Dauer der Wartezeit gemessen. Ab einer bestimmten Zeitgrenze soll, trotz geringerer Dichte, die Ampel auf Grün geschaltet werden, um übermäßig lange Wartezeiten bei den Verkehrsteilnehmern zu verhindern.

In der Realität besteht das Problem, dass es – außer bei einem vernetzten Verkehrsleitsystem – zu chaotischen Verhältnissen zwischen den Verkehrsteilnehmern kommen kann, wenn Einsatzfahrzeuge (Feuerwehr, Krankenwagen, Polizei), auf dem Weg zu einem Notfall, eine Kreuzung überqueren müssen, da diese ggf. bei „rot“ über dieselbige fahren müssen. Aus diesem Grund soll berücksichtigt werden, dass Einsatzfahrzeuge, bei einem Einsatz, unbedingten Vorrang vor allen anderen Verkehrsteilnehmern haben.



1.3 Zieldefinition

Ziel ist es, eine Ampelsteuerung zu realisieren, welche folgende Parameter berücksichtigt...

- Dichte der (fahrenden und wartenden) Fahrzeuge pro Straße
- Wartezeit der Fahrzeuge pro Straße
- Wartezeit der Fußgänger pro Übergang
- Ob sich ein Einsatzfahrzeug, im Einsatz befindet

... und ohne menschliches Einwirken, fehlerfrei funktioniert.

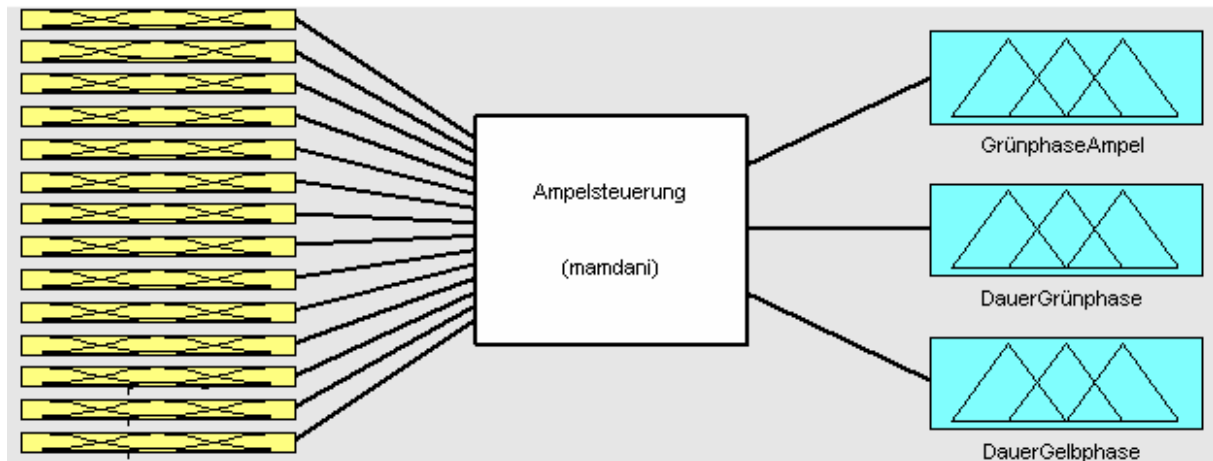
Das soll heißen, dass die Ampelsteuerung, abhängig von den gegebenen Parametern, selbständig entscheiden soll, welche Straße die Berechtigung erhält, zu fahren.

Zum Beispiel sei die Dichte auf Straße 1 und Straße 2 hoch, während die Dichte auf Straße 3 und Straße 4 nur niedrig ist. Die Fußgänger, die darauf warten, Straße 1 zu überqueren, warten bereits 30 Sekunden und zurzeit ist kein Einsatzfahrzeug im Einsatz. Die Ampelsteuerung soll nun in der Lage sein, eine Spur frei zu geben.

2 Sekunden, bevor die Ampel auf „gelb“ schaltet, wird Dichte, Wartezeit und Vorhandensein eines Einsatzfahrzeugs im Einsatz gemessen. Nun soll entweder eine neue Straße frei gegeben werden oder die Freigabe der bereits fahrenden Straße, um die Zeit x verlängert werden. Sobald ein Einsatzfahrzeug im Einsatz erscheint, soll dessen Straße sofort auf „grün“ geschaltet werden.

2. Umsetzung

Mit dem Tool MATLAB® wurde nun unter zu Hilfenahme der Toolbox „Fuzzy Logic Toolbox“ eine entsprechende Lösung veranschlagt. Dazu wurden zuerst vierzehn Eingangsvariablen und drei Ausgangsvariablen heran gezogen, um die Ampelsteuerung nach der Zieldefinition zu realisieren.



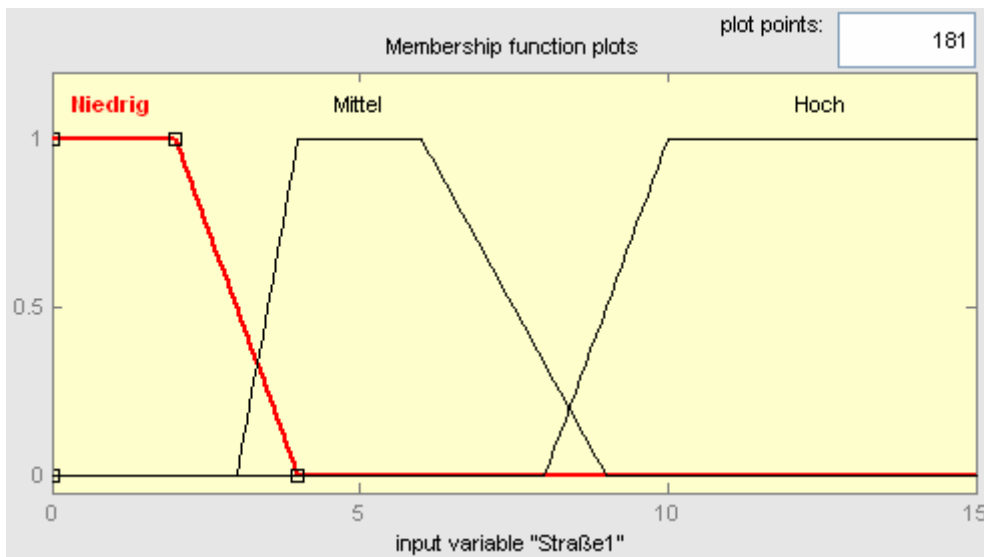
2.1 Eingangsvariablen

2.1.1 Dichte

Die Variable „Dichte“ kann drei Werte annehmen: niedriges Verkehrsaufkommen, mittleres Verkehrsaufkommen und hohes Verkehrsaufkommen. Der Wertebereich erstreckt sich von 0 bis 15 (Fahrzeuge):

- Niedriges Verkehrsaufkommen: 0 - 4
- Mittleres Verkehrsaufkommen: 3 - 9
- Hohes Verkehrsaufkommen: 8 - 15

Eine Anzahl Fahrzeuge, welche über den Wertebereich von 15 hinausgeht, wird als 15 angenommen. Wir haben die Trapez-Funktion (trapmf) benutzt. Diese Variable wurde für alle 4 Straßen (Straße1 bis Straße4) übernommen.

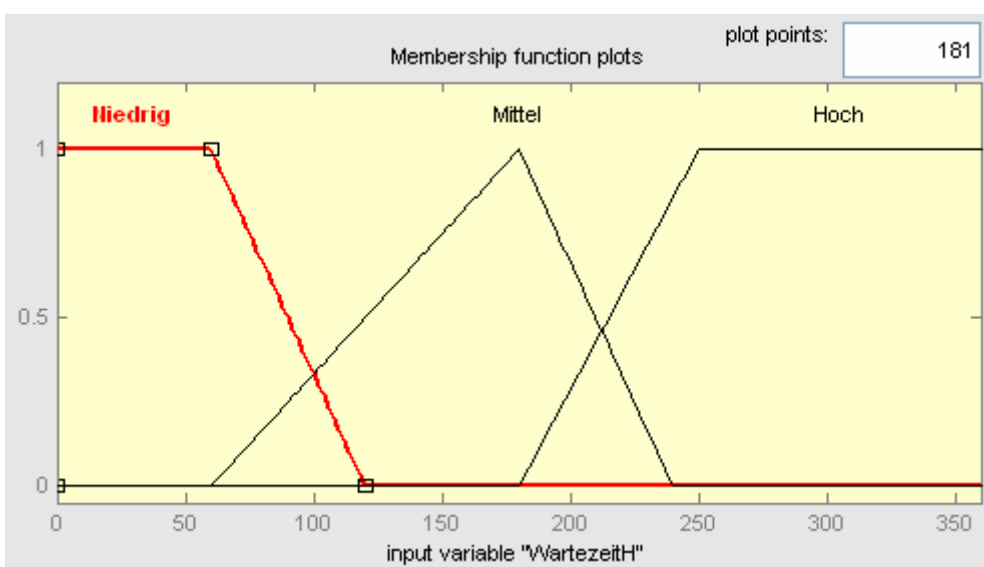


2.1.2 Wartezeit Fußgänger

Die Variable „Wartezeit Fußgänger“ kann drei Werte annehmen: niedrige Wartezeit, mittlere Wartezeit und hohe Wartezeit. Der Wertebereich erstreckt sich von 0 bis 360 (Sekunden):

- Niedriges Wartezeit: 0 - 120
- Mittleres Wartezeit: 60 - 240
- Hohe Wartezeit: 180 – 360

Eine Wartezeit, welche über den Wertebereich von 360 hinausgeht, wird als 360 angenommen. Die hohe Wartezeit wird als kritischer Wert betrachtet. Das heißt, die Ampel soll möglichst schnell freigegeben werden. Für die niedrige und hohe Wartezeit haben wir die Trapez-Funktion (trapmf) benutzt. Für den Fall einer mittleren Wartezeit wurde die Dreiecks-Funktion (trimf) verwendet. Diese Variable wurde für sämtliche Fußgängerübergänge (WartezeitH und WartezeitN) übernommen.

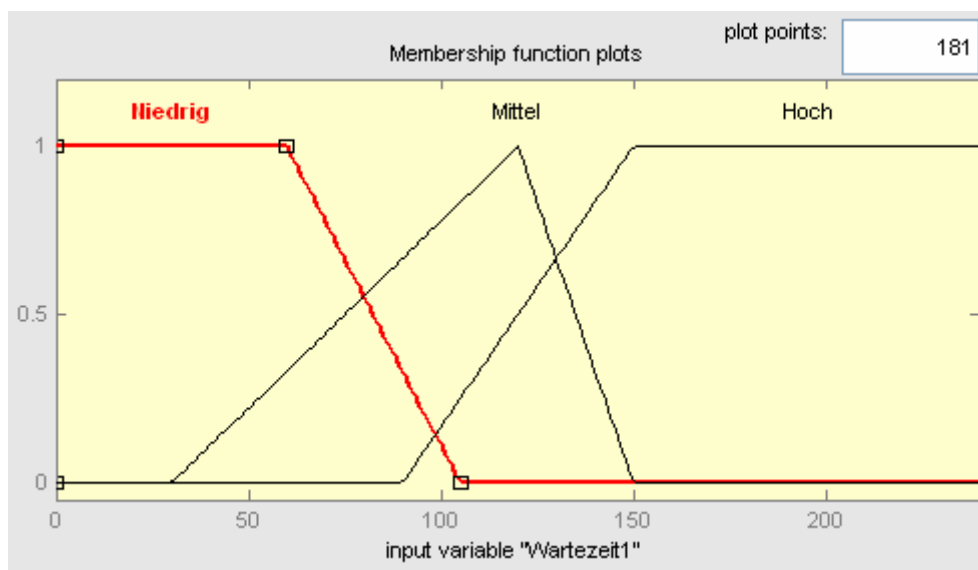


2.1.3 Wartezeit Fahrzeuge

Die Variable „Wartezeit Fahrzeuge“ kann drei Werte annehmen: niedrige Wartezeit, mittlere Wartezeit und hohe Wartezeit. Der Wertebereich erstreckt sich von 0 bis 240 (Sekunden):

- Niedrige Wartezeit: 0 - 105
- Mittleres Wartezeit: 30 - 150
- Hohe Wartezeit: 90 – 240

Eine Wartezeit, welche über den Wertebereich von 240 hinausgeht, wird als 240 angenommen. Die hohe Wartezeit wird als kritischer Wert betrachtet. Das heißt, die Ampel soll möglichst schnell freigegeben werden. Für die niedrige und hohe Wartezeit haben wir die Trapez-Funktion (trapmf) benutzt. Für den Fall einer mittleren Wartezeit wurde die Dreiecks-Funktion (trimf) verwendet. Diese Variable wurde für sämtliche Straßen (Straße1 bis Straße4) übernommen.

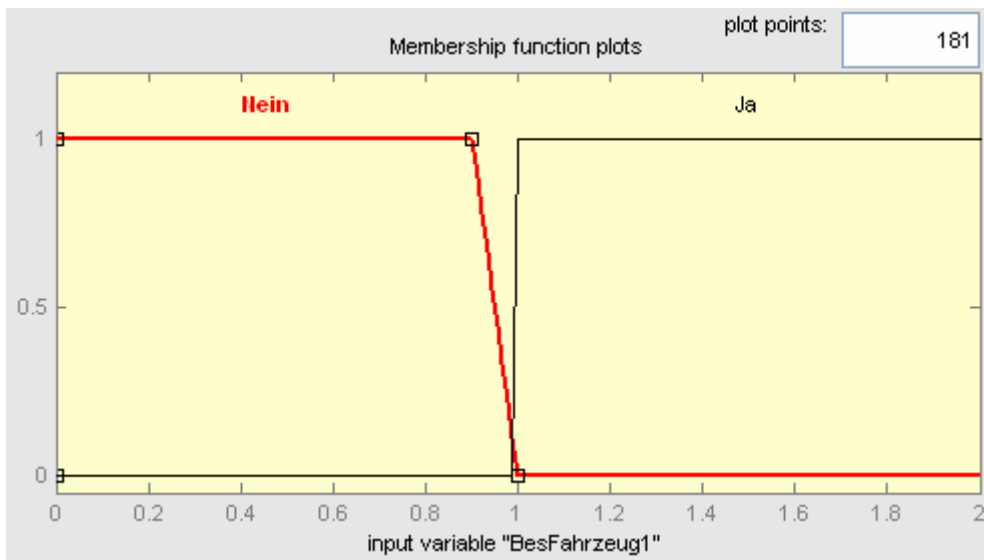


2.1.4 Besonderes Fahrzeug

Die Variable „Besonderes Fahrzeug“ kann zwei Werte annehmen: Kein besonderes Fahrzeug (nein) und ein oder mehrere besondere Fahrzeuge (ja). Der Wertebereich erstreckt sich von 0 bis 2. Hiermit simulieren wir eine boolesche Variable:

- Kein besonderes Fahrzeug: 0 - 1
- Ein oder mehrere besondere Fahrzeuge: 1 - 2

Die Anwesenheit eines besonderen Fahrzeugs wird als kritischer Wert betrachtet. Das heißt, die Ampel soll unverzüglich freigegeben werden. Hierfür nutzen wir die Trapez-Funktion (trapmf). Diese Variable wurde für sämtliche Straßen (Straße1 bis Straße4) übernommen.



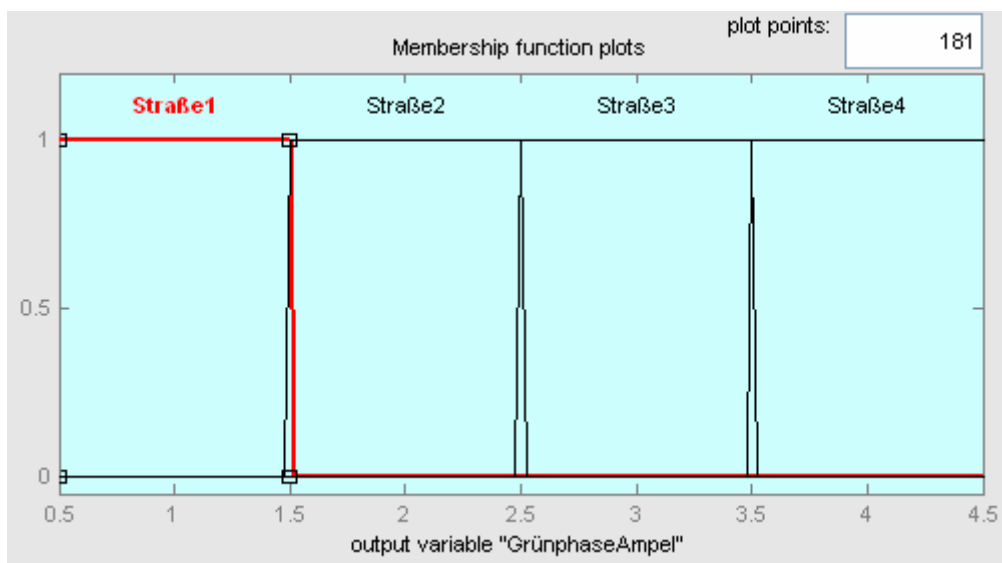
2.2 Ausgangsvariablen

2.2.1 Grünphase Ampel

Die Variable „Grünphase Ampel“ kann vier Werte annehmen, die angeben, welche Straße als nächstes „grün“ geschaltet wird: Straße1, Straße2, Straße3 oder Straße4. Der Wertebereich erstreckt sich von 0,5 bis 4,5. Hiermit simulieren wir einen scharfen integer Wert:

- Straße1: 0,5 – 1,5
- Straße2: 1,5 – 2,5
- Straße3: 2,5 – 3,5
- Straße4: 3,5 – 4,5

Diese Werte müssen scharf (eindeutig) sein, da unmissverständlich, genau eine Straße identifiziert werden muss. Hierfür nutzen wir die Trapez-Funktion (trapmf).

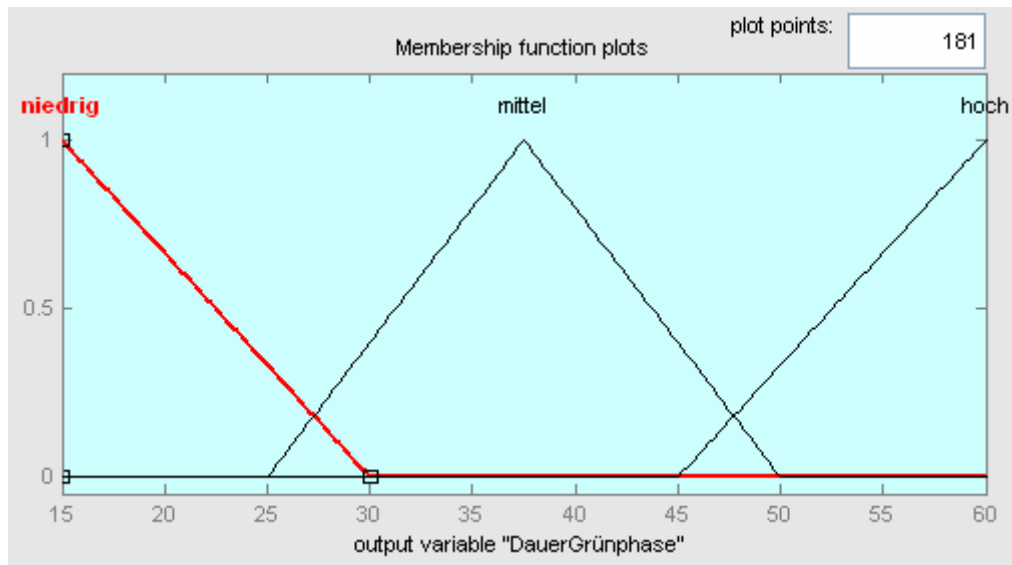


2.2.2 Dauer Grünphase

Die Variable „Dauer Grünphase“ kann drei Werte annehmen, die angeben, wie lange die Ampel auf Grün geschaltet wird: niedrig, mittel und hoch. Der Wertebereich erstreckt sich von 15 bis 60 (Sekunden):

- Niedrig: 15 – 30
- Mittel: 25 – 50
- Hoch: 45 – 60

Hierfür nutzen wir die Dreiecks-Funktion (trimf).

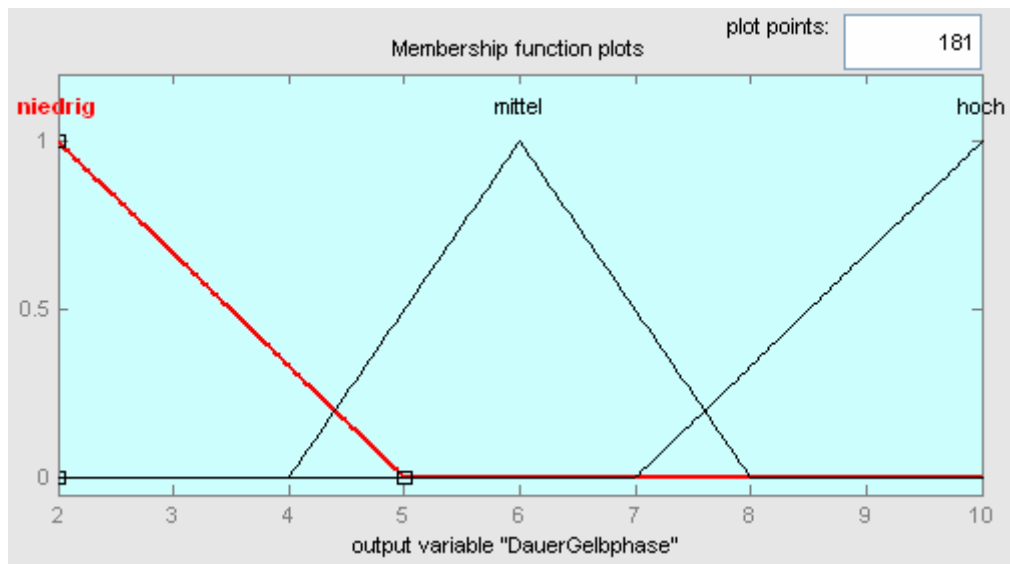


2.2.3 Dauer Gelbphase

Die Variable „Dauer Gelbphase“ kann drei Werte annehmen, die angeben, wie lange die Ampel auf Gelb geschaltet wird: niedrig, mittel und hoch. Der Wertebereich erstreckt sich von 2 bis 10 (Sekunden):

- Niedrig: 2 – 5
- Mittel: 4 – 8
- Hoch: 7 – 10

Hierfür nutzen wir die Dreiecks-Funktion (trimf).



3. Einstellungen des Fuzzy-Reglers

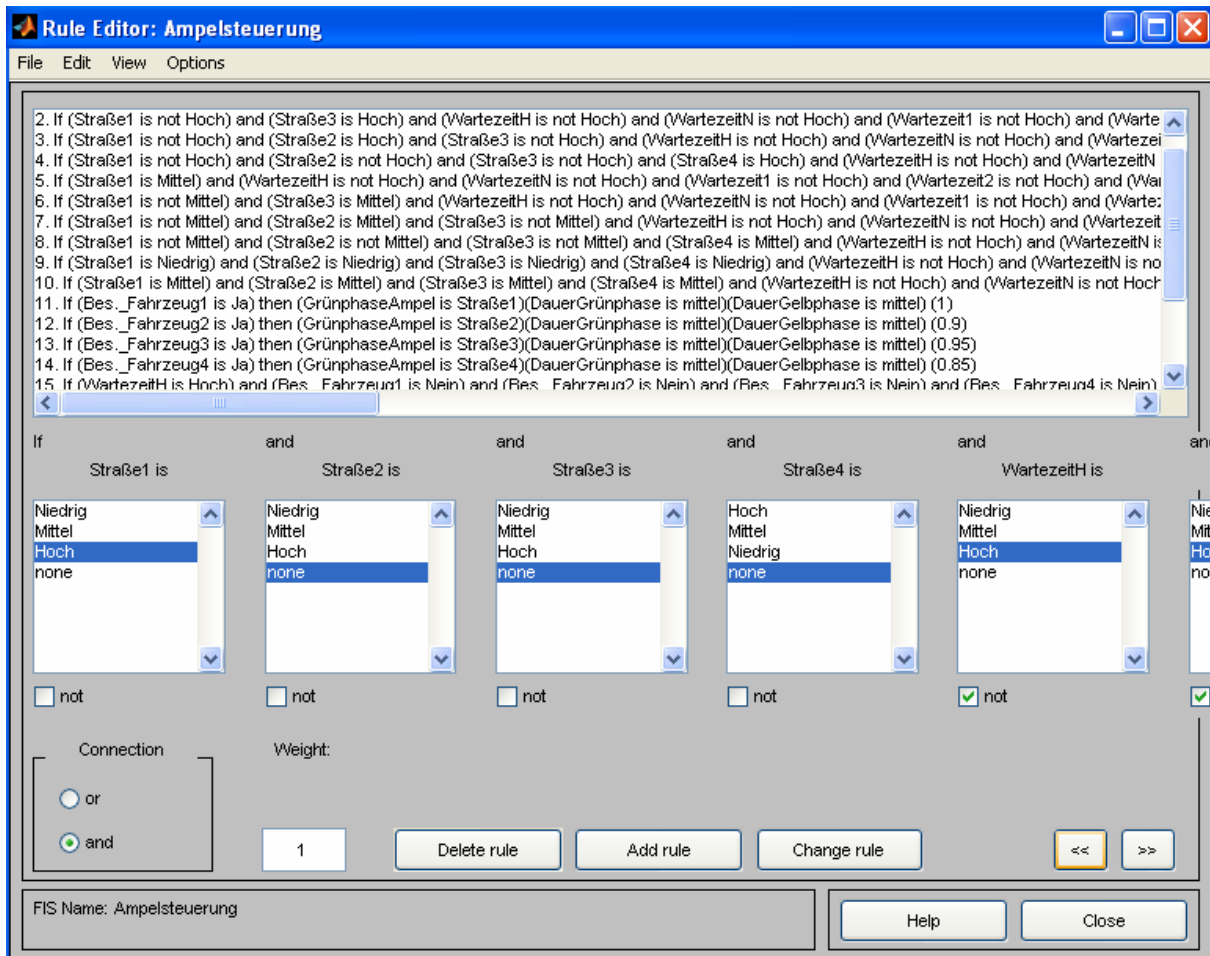
3.1 Fuzzyfizierung

Um scharfe Eingangsgrößen in Zugehörigkeitsgrade für Ausprägungen in linguistische Variablen zu überführen, werden die oben beschriebenen Zugehörigkeitsfunktionen verwendet; siehe 2.1 und 2.2.

3.2 Inferenz

Zur Beschreibung der notwendigen Beziehungen unter den einzelnen Eingangsvariablen, wurden 20 Regeln erstellt. Diese wurden mit bestimmten Bedingungen verknüpft, damit ihr Zusammenspiel zu entsprechenden Werten bei den Ausgangsvariablen führt.

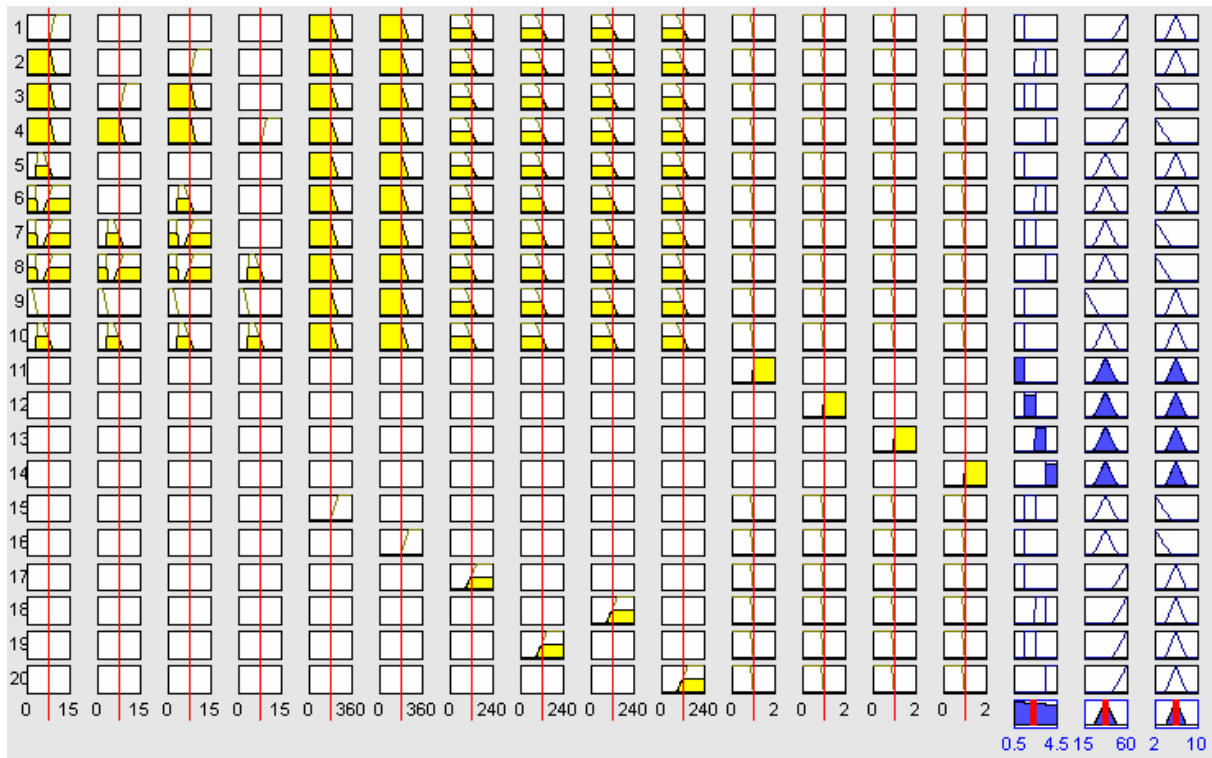
(Besonderes Fahrzeug, Wartezeiten im kritischen Bereich) vorrangig behandelt werden. Außerdem wurden diese gewichtet, um sicher zu stellen, dass die Hauptstraße Vorrang vor den Nebenstraßen hat und damit es, im Falle zweier zutreffender Regeln (z. B. besonderes Fahrzeug auf Straße3 und Straße4) nicht zu einer Kollision kommt.



3.3 Defuzzifizierung

Bei der Defuzzifizierung werden die linguistischen Ausgangsgrößen in skalare Werte überführt. Man kann sehr gut sehen, wie die 20 Bedingungen miteinander in Beziehung stehen und wie sämtliche Entscheidungen am Ende die Ampel steuern.

Hierbei ist zu beachten, dass die von MATLAB® verwendete Schwerpunktsfunktion für die Output-Variablen DauerGelbphase und DauerGrünphase sehr gut funktioniert aber für GrünphaseAmpel nicht geeignet ist. Aufgrund der besonderen Struktur (siehe oben) wäre eine Maximum-Funktion besser geeignet um, die Umschaltende Straße anzeigen zu lassen. Aber auch so ist es mit MATLAB® möglich, einen guten Überblick zu bekommen.

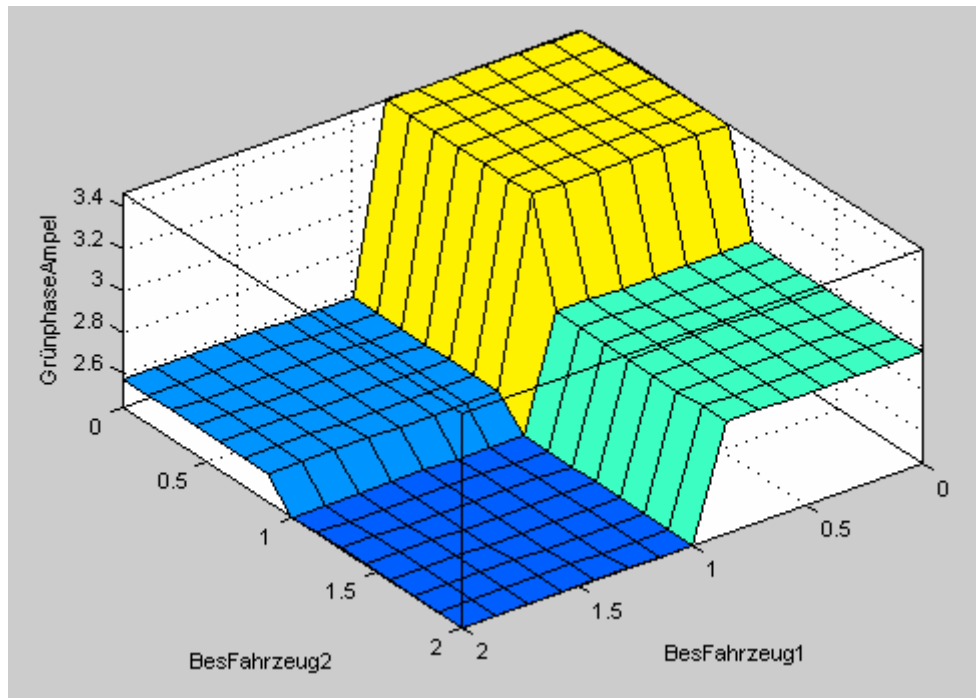


4. Darstellung der Variablen als Surfaces

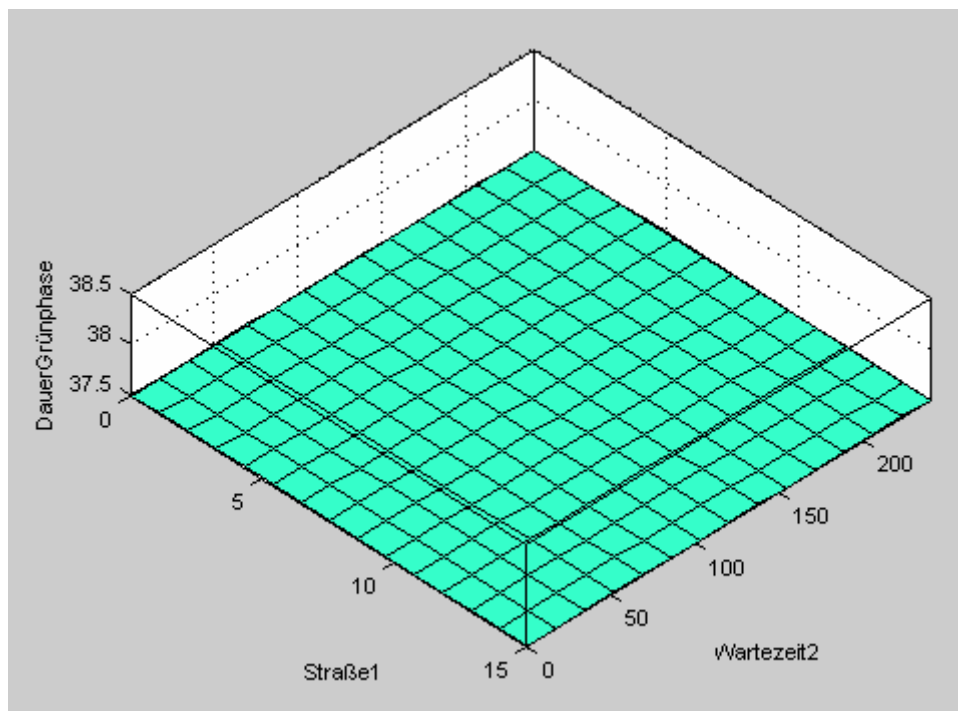
Im folgendem wird nun eine Auswahl der Variablen in dreidimensionalen Gebilden, so genannten Surfaces, dargestellt. Dieses dient der Übersichtlichkeit unter den Eingangsvariablen und den Ausgangsvariablen.

4.1 Grünphase Ampel (Besonderes Fahrzeug1 und Besonderes Fahrzeug2)

Dieses Surface zeigt, welche Ampel beim aufkommen vom Besonderen Fahrzeug1 und Besonderem Fahrzeug2 auf „grün“ geschaltet wird.



4.2 Dauer der Grünphase (Straße1 und Wartezeit2)



5. Fazit

Mit der Fuzzy-Logik lässt sich die Problemstellung der Ampelsteuerung nur bedingt lösen. Oft ist das ungewohnte Hineinversetzen in die Denkweise unscharfer Grenzen, schwierig. Dies mag ggf. daran liegen, dass wir auf unserem bisherigen Weg zum Dipl. Informatiker ausschließlich mit konkreten Fakten und Werten konfrontiert wurden.

Die Umsetzung des Projekts in MATLAB® erwies sich als schwierig, da eine Ampelsteuerung zu viele konkrete Werte (z. B. welche Ampel als nächstes auf „grün“ geschaltet wird), wofür das Programm nicht ausgelegt ist. Daher mussten wir dieses durch annähernd konkrete Funktionen „austricksen“ und erhielten größtenteils „flache“ Surfaces. Ebenso erwies sich der Umgang mit vielen Inputvariablen, welche allerdings alle notwendig waren, als schwierig. Die benutzerfreundliche Handhabung von MATLAB® stößt hierbei schnell an seine Grenzen.

Zu Beginn des Projekts erwies sich die Fülle an Eingangsvariablen außerdem als schwierig, da beim Versuch, alle miteinander zu verbinden, zu viele Regeln entstanden sind, welche erst gegen Ende des Projekts zu einem annehmbaren Maß zusammengekürzt werden konnten.

Trotz dieser Probleme gelang es der Projektgruppe, alle Zielsetzungen zu erfüllen und das Projekt zu einem positiven Abschluss zu bringen. Außerdem sind wir der Meinung, durch dieses Projekt viele positive Lerneffekte und Erfahrungen gesammelt zu haben.

Abschließend ist zu sagen, dass sich eine Ampelsteuerung, aufgrund der genannten, vielen scharfen Werte nicht als ein Fuzzy-Logik-Projekt eignet.

6. Anhang

```
[System]
Name='Ampelsteuerung'
Type='mamdani'
Version=2.0
NumInputs=14
NumOutputs=3
NumRules=20
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='Straße1'
Range=[0 15]
NumMFs=3
MF1='Niedrig': 'trapmf',[0 0 2 4]
MF2='Mittel': 'trapmf',[3 4 6 9]
MF3='Hoch': 'trapmf',[8 10 15 15]

[Input2]
Name='Straße2'
Range=[0 15]
NumMFs=3
MF1='Niedrig': 'trapmf',[0 0 2 4]
MF2='Mittel': 'trapmf',[3 4 6 9]
MF3='Hoch': 'trapmf',[8 10 15 15]

[Input3]
Name='Straße3'
Range=[0 15]
NumMFs=3
MF1='Niedrig': 'trapmf',[0 0 2 4]
MF2='Mittel': 'trapmf',[3 4 6 9]
MF3='Hoch': 'trapmf',[8 10 15 15]

[Input4]
Name='Straße4'
Range=[0 15]
NumMFs=3
MF1='Hoch': 'trapmf',[8 10 15 15]
MF2='Mittel': 'trapmf',[3 4 6 9]
MF3='Niedrig': 'trapmf',[0 0 2 4]

[Input5]
Name='WartezeitH'
Range=[0 360]
NumMFs=3
MF1='Niedrig': 'trapmf',[0 0 60 120]
MF2='Mittel': 'trimf',[60 180 240]
MF3='Hoch': 'trapmf',[180 250 360 360]

[Input6]
Name='WartezeitN'
Range=[0 360]
NumMFs=3
MF1='Niedrig': 'trapmf',[0 0 60 120]
MF2='Mittel': 'trimf',[60 180 240]
MF3='Hoch': 'trapmf',[180 250 360 360]
```

WBS Projekt: Ampelsteuerung

```
[Input7]
Name='Wartezeit1'
Range=[0 240]
NumMFs=3
MF1='Niedrig':'trapmf',[0 0 60 105]
MF2='Mittel':'trimf',[30 120 150]
MF3='Hoch':'trapmf',[90 150 240 240]
```

```
[Input8]
Name='Wartezeit2'
Range=[0 240]
NumMFs=3
MF1='Niedrig':'trapmf',[0 0 60 105]
MF2='Mittel':'trimf',[30 120 150]
MF3='Hoch':'trapmf',[90 150 240 240]
```

```
[Input9]
Name='Wartezeit3'
Range=[0 240]
NumMFs=3
MF1='Niedrig':'trapmf',[0 0 60 105]
MF2='Mittel':'trimf',[30 120 150]
MF3='Hoch':'trapmf',[90 150 240 240]
```

```
[Input10]
Name='Wartezeit4'
Range=[0 240]
NumMFs=3
MF1='Niedrig':'trapmf',[0 0 60 105]
MF2='Mittel':'trimf',[30 120 150]
MF3='Hoch':'trapmf',[90 150 240 240]
```

```
[Input11]
Name='BesFahrzeug1'
Range=[0 2]
NumMFs=2
MF1='Nein':'trapmf',[0 0 0.9 1]
MF2='Ja':'trapmf',[1 1 2 2]
```

```
[Input12]
Name='BesFahrzeug2'
Range=[0 2]
NumMFs=2
MF1='Nein':'trapmf',[0 0 0.9 1]
MF2='Ja':'trapmf',[1 1 2 2]
```

```
[Input13]
Name='BesFahrzeug3'
Range=[0 2]
NumMFs=2
MF1='Nein':'trapmf',[0 0 0.9 1]
MF2='Ja':'trapmf',[1 1 2 2]
```

```
[Input14]
Name='BesFahrzeug4'
Range=[0 2]
NumMFs=2
MF1='Nein':'trapmf',[0 0 0.9 1]
MF2='Ja':'trapmf',[1 1 2 2]
```

```
[Output1]
Name='GrünphaseAmpel'
```

WBS Projekt: Ampelsteuerung

```
Range=[0.5 4.5]
NumMFs=4
MF1='Straße1':'trapmf',[0.5 0.5 1.5 1.5]
MF2='Straße3':'trapmf',[2.5 2.5 3.5 3.5]
MF3='Straße4':'trapmf',[3.5 3.5 4.5 4.5]
MF4='Straße2':'trapmf',[1.5 1.5 2.5 2.5]
```

```
[Output2]
Name='DauerGrünphase'
Range=[15 60]
NumMFs=3
MF1='niedrig':'trimf',[15 15 30]
MF2='mittel':'trimf',[25 37.5 50]
MF3='hoch':'trimf',[45 60 60]
```

```
[Output3]
Name='DauerGelbphase'
Range=[2 10]
NumMFs=3
MF1='niedrig':'trimf',[2 2 5]
MF2='mittel':'trimf',[4 6 8]
MF3='hoch':'trimf',[7 10 10]
```

```
[Rules]
3 0 0 0 -3 -3 -3 -3 -3 -3 1 1 1 1, 1 3 2 (1) : 1
-3 0 3 0 -3 -3 -3 -3 -3 -3 1 1 1 1, 2 3 2 (1) : 1
-3 3 -3 0 -3 -3 -3 -3 -3 -3 1 1 1 1, 4 3 1 (1) : 1
-3 -3 -3 1 -3 -3 -3 -3 -3 -3 1 1 1 1, 3 3 1 (1) : 1
2 0 0 0 -3 -3 -3 -3 -3 -3 1 1 1 1, 1 2 2 (0.6) : 1
-2 0 2 0 -3 -3 -3 -3 -3 -3 1 1 1 1, 2 2 2 (0.6) : 1
-2 2 -2 0 -3 -3 -3 -3 -3 -3 1 1 1 1, 4 2 1 (0.6) : 1
-2 -2 -2 2 -3 -3 -3 -3 -3 -3 1 1 1 1, 3 2 1 (0.6) : 1
1 1 1 3 -3 -3 -3 -3 -3 -3 1 1 1 1, 1 1 2 (1) : 1
2 2 2 2 -3 -3 -3 -3 -3 -3 1 1 1 1, 1 2 2 (1) : 1
0 0 0 0 0 0 0 0 0 0 2 0 0 0, 1 2 2 (1) : 1
0 0 0 0 0 0 0 0 0 0 0 2 0 0, 4 2 2 (0.9) : 1
0 0 0 0 0 0 0 0 0 0 0 0 2 0, 2 2 2 (0.95) : 1
0 0 0 0 0 0 0 0 0 0 0 0 0 2, 3 2 2 (0.85) : 1
0 0 0 0 3 0 0 0 0 0 1 1 1 1, 4 2 1 (0.75) : 1
0 0 0 0 0 3 0 0 0 0 1 1 1 1, 1 2 1 (0.7) : 1
0 0 0 0 0 0 3 0 0 0 1 1 1 1, 1 3 2 (1) : 1
0 0 0 0 0 0 0 0 3 0 1 1 1 1, 2 3 2 (0.95) : 1
0 0 0 0 0 0 0 0 3 0 0 1 1 1 1, 4 3 2 (0.9) : 1
0 0 0 0 0 0 0 0 0 3 1 1 1 1, 3 3 2 (0.85) : 1
```