

Fachbereich
Mathematik, Naturwissenschaften und Informatik

Ausarbeitung Projektvortrag **Fuzzy-Regelung**

Im Rahmen der Veranstaltung:
Praktikum Künstliche Intelligenz(CS5330)

Autoren:

Joel Bartelheimer
joel.bartelheimer@mni.thm.de

Nico Müller
nico.mueller@mni.thm.de

Eingereicht bei:

Prof. Dr. Wolfgang Henrich

Abgabedatum:

14.02.2017

II Inhaltsverzeichnis

| | | |
|-----------|---------------------------|-----------|
| II | Inhaltsverzeichnis | ii |
| 1 | Problemstellung | 1 |
| 2 | GUI | 2 |
| 3 | Konfiguration | 2 |
| 4 | Implementierung | 5 |
| 5 | Probleme | 5 |

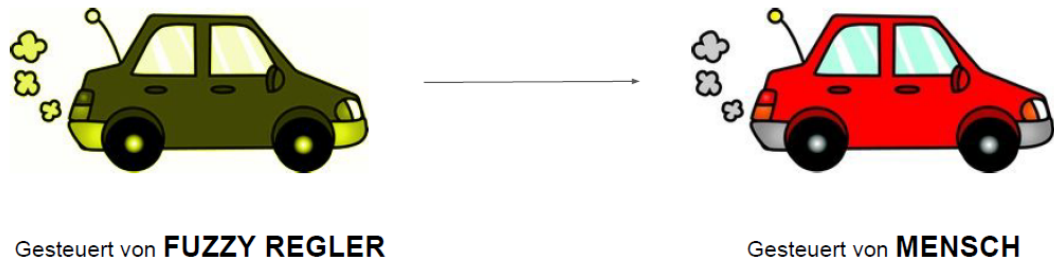


Abbildung 1: Problemstellung

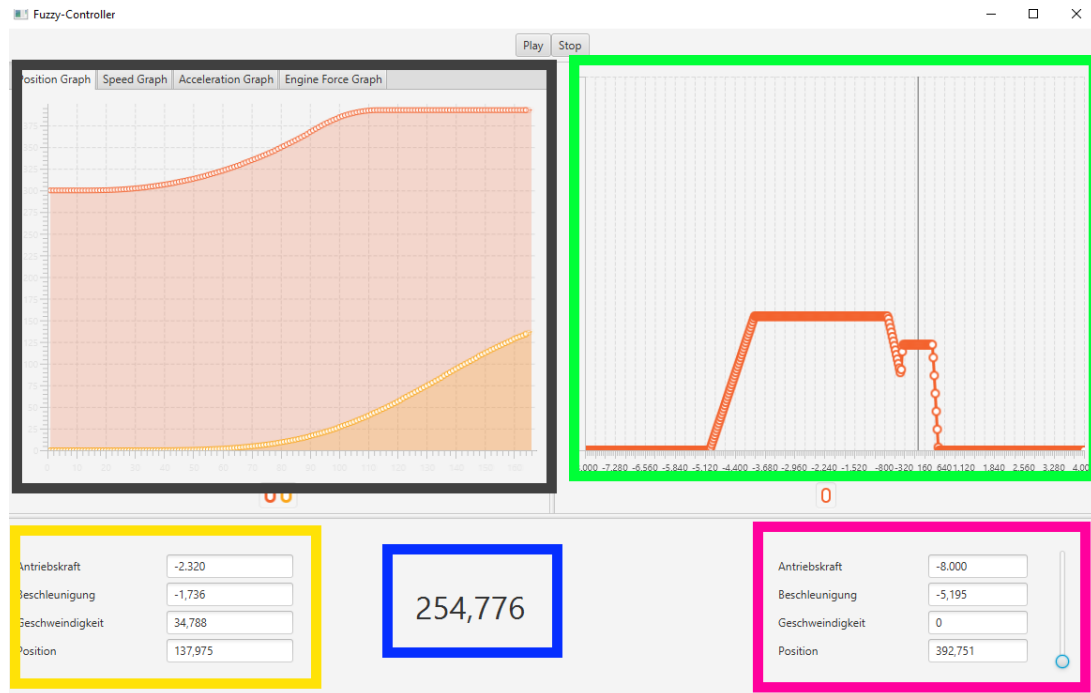


Abbildung 2: Die GUI des Regelung-Programms

1 Problemstellung

In dem Problemszenario verfolgen sich zwei Autos. Das vordere Auto A wird von einem Menschen geregelt. Das Verfolger Auto B soll durch eine Fuzzy-Regelung gesteuert werden und Auto in einem angemessenen Abstand folgen. Auto B kennt lediglich die eigenen Messwert, dh. nicht die Geschwindigkeit des Auto A, sowie die Entfernung zu Auto A. Auto B und Auto A haben die gleichen physikalischen Eigenschaften. Die Simulation soll den physikalischen Gesetzen folgen, dh. Luftwiderstand, Höchstgeschwindigkeiten und ein insgesamt realistisches Verhalten ist gefordert.

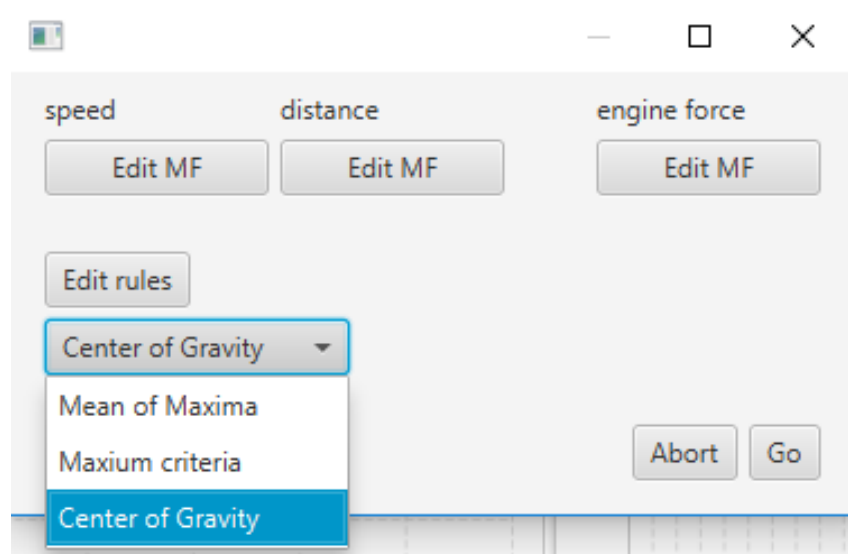


Abbildung 3: Konfigurations Übersicht

2 GUI

In Abbildung 2 ist die GUI des Programms zu sehen. In dem grünen Kasten befindet sich die Ausgangs Fuzzy-Menge die an das Defuzzifizierungsinterface gegeben wird. Die Kästchen Lila und Gelb enthalten jeweils die momentanen Daten von Auto A und B. Für Auto B kann man mit dem Slider die Antriebskraft des vorderen Autos bestimmen. Das blaue Kästchen enthält den momentanen Abstand zwischen den beiden Autos in Metern. Die Tabs in dem schwarzen Kästchen zeigen Echtzeit Informationen über die Simulation. So wird ständig die Position, die Geschwindigkeit, die Beschleunigung und die Antriebskraft geplottet (rot = Auto A und orange = Auto B).

3 Konfiguration

Mögliche Konfigurationen sind:

- Partitionierung der Eingangsvariablen
- Partitionierung der Ausgangsvariablen
- Erstellung der linguistischen Regeln
- Wahl der Defuzzifizierungsmethode

Es wurden folgende Voreinstellungen für die Regelung gemacht:

Wertebereiche Distanz: 0-500 Meter

Geschwindigkeit: 0-250 km/h

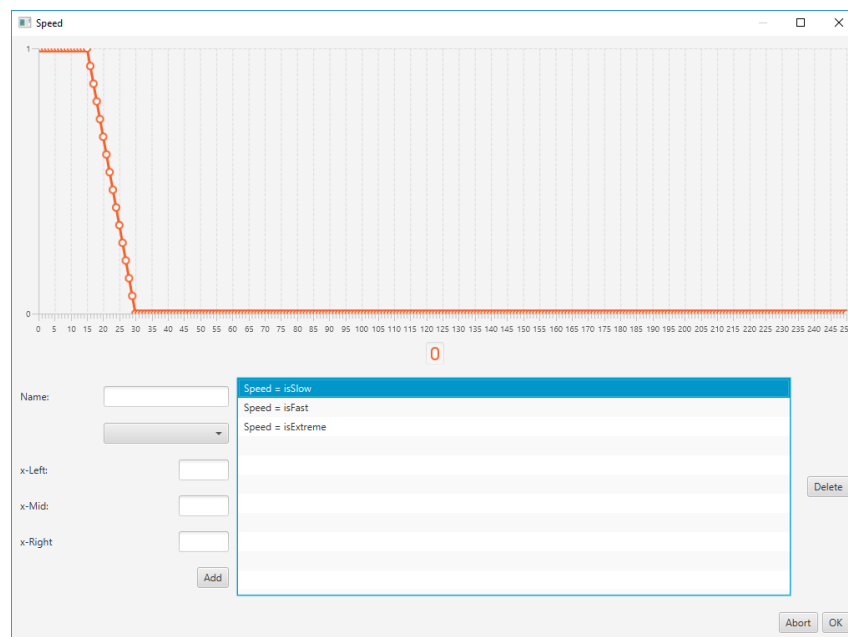


Abbildung 4: Partitionierung der Fuzzy-Mengen

Abbildung 5: Bestimmung der Linguistischen Regeln

Beschleunigung: -8000-4000 N

Partitionierung Distanz

isVeryClose = $-\infty$, 100, 200

isClose = 100, 200, 300

isNormal = 250, 300, 350

isFar = 300, 300, 500

isVeryFar = 400, 500, ∞

Partitionierung Geschwindigkeit

isSlow = $-\infty$, 15, 30

isFast = 15, 50, 100

isExtreme = 50, 150, 350

Partitionierung Beschleunigung

fullBrake = $-\infty$, -8000, -7000

medBrake = -8000, -5000, -2000

brake = -5000, -2000, 0

roll = -500, 0, 500

speed = 0, 1000, 2000

medSpeed = 1000, 2000, 3000

fullSpeed = 2000, 3000, ∞

Regeln

-
- 1 IF Distance = isVeryFar AND Speed = isSlow THEN Force = fullSpeed
 - 2 IF Distance = isVeryFar AND Speed = isFast THEN Force = fullSpeed
 - 3 IF Distance = isVeryFar AND Speed = isExtreme THEN Force = fullSpeed
 - 4 IF Distance = isFar AND Speed = isSlow THEN Force = fullSpeed
 - 5 IF Distance = isFar AND Speed = isFast THEN Force = medSpeed
 - 6 IF Distance = isFar AND Speed = isExtreme THEN Force = speed
 - 7 IF Distance = isNormal AND Speed = isSlow THEN Force = roll
 - 8 IF Distance = isNormal AND Speed = isFast THEN Force = roll
 - 9 IF Distance = isNormal AND Speed = isExtreme THEN Force = roll
 - 10 IF Distance = isClose AND Speed = isSlow THEN Force = brake
 - 11 IF Distance = isClose AND Speed = isFast THEN Force = medBrake

```
12 IF Distance = isClose AND Speed = isExtreme THEN Force = fullBrake
13 IF Distance = isVeryClose AND Speed = isSlow THEN Force = fullBrake
14 IF Distance = isVeryClose AND Speed = isFast THEN Force = fullBrake
15 IF Distance = isVeryClose AND Speed = isExtreme THEN Force = fullBrake
```

4 Implementierung

Für die Implementierung wird die Sprache **Scala** verwendet. Folgende Klassen werden verwendet.

Car.scala Enthält das physikalische Modell eines Autos.

FuzzyCarController.scala Implementierung eines Fuzzy-Controllers nach Mamdami Ansatz.

FuzzyConfig.scala Enthält die Klassen: FuzzyTerm.scala, FuzzyRule.scala, FuzzyDefuzzyFunc.scala, FuzzyValueConnector.scala

MembershipFunctions.scala Enthält Funktionen wie die Dreiecksfunktion

5 Probleme

Leider ist dem Team zu spät aufgefallen, dass es sinnvoll gewesen wäre eine Ableitung des Abstands in die Regelung mit einzubringen. Die gesamte Regelung erscheint nun ein wenig Träge.