

Studienarbeit

# **AUTONOMES FAHRZEUG** Dokumentation

Version 1.0



## **Artificial Neural Network Application**

Roland Fehlmann Rico Steffen

6. Juli 2006

Betreut durch Herr A. Kurka





# Aufgabenstellung

## Einführung

In der Lagerhaltung oder in den Montageabläufen eines Industrie-Unternehmens werden oft selbständig agierende Montage- oder Transport-Einheiten gewünscht. Die einfachste Variante einer solche Transport-Einheit wäre ein Fahrzeug (etwa Hubstaplergrösse) der innerhalb einer Industriehalle selbständig agierend den Weg von A nach B suchen würde. Selbstverständlich müssen auch Hindernisse erkannt werden und umgangen werden. Die Suchalgorithmen sollen aus bekannten Abmessungen des Fahrbereichs und zum voraus unbekanntem Hindernissen den kürzesten Weg suchen. Zur Erkennung der Hindernisse können mehrere Laserentfernung-Messgeräte verwendet werden. Es sollen dabei keine «tray and error» Methoden verwendet werden, sondern KNN-Methoden wie z.B. Kohonen- oder Hopfield Netz evt. mit Kombination von Fuzzy-Logik.

## Aufgabe

- Erstellen des detaillierten Pflichtenheftes
- Softwarekonzept erarbeiten
- Softwaremodul programmieren gemäss nachfolgender Beschreibung
- Virtuelle Simulation der Bewegungen eines Autonomes-Fahrzeugs gemäss Punkt 1. Entsprechende GUI zur Eingabe der Zielkoordinaten, Hallengrundrisses und Hindernisse. Darstellung der berechnete- sowie aktuell befahrene - Strecke mit Anzeige der erkannten Hindernisse.

## Vorgehen

Im Rahmen der allgemeinen Richtlinien zur Durchführung von Studien- & Diplomarbeiten gemäss eigenem Projektmanagementplan. Dieser Projektmanagementplan ist als Erstes von den Studenten zu erstellen und enthält insbesondere:

- Die Beschreibung des dem Projektcharakter angepassten Vorgehensmodells.



- Eine erste Aufteilung der Aufgabe in gemeinsam und einzeln zu bearbeitende Teile unter Berücksichtigung der vorgegebenen Teilaspekte. Die genaue Aufteilung erfolgt nach der Anforderungsanalyse.
- Den Projektplan (Zeitplan) und die Meilensteine.

## Randbedingungen

- Plattform: Windows XP Professional
- Programmiersprache: Java / C++/C#, freie Wahl mit Begründung

## Infrastruktur

Die Arbeiten werden auf den zugeteilten Rechnern mit der Standardinstallation durchgeführt.

## Erwartete Resultate

**Was soll realisiert werden** Siehe das unter Punkt 2 beschriebene System. Das System wird von den Studenten in der Anforderungsspezifikation detailliert beschrieben und nach Genehmigung durch den Auftraggeber realisiert.

## Was muss abgegeben werden

- In elektronischer Form:
  - Installationskit (alle Dateien für eine Installation und Installationsanweisung)
  - kompletter Source Code
  - JavaDoc / C++Doc / C# Doc
  - alle Dokumente
  - alle Protokolle
  - Zeiterfassung
- Auf Papier:
  - Beschreibung der Arbeit (1 Blatt)
  - Zusammenfassung der Ergebnisse
  - Anforderungsspezifikation
  - Analyse- und Design- Dokumentation
  - Projektmanagementplan und Projektplan



- Zusammenfassung der Studienarbeit mit den persönlichen Projektberichten und der zusammengefassten Aufwandserfassung

Es muss aus den abgegebenen Dokumenten klar hervorgehen, wer für welchen Teil der Arbeit und der Dokumentation verantwortlich war.

## Termine

**Start:** 03.04.2006

**Ende:** Freitag 07.07.2006, 17:00 Uhr

**Zeitplan und Meilensteine:** Zeitplan und Meilensteine für das Projekt sind von den Studenten selber zu erarbeiten und zusammen mit dem Projektmanagementplan abzuliefern. Die Meilensteine sind bindend. Der erste Meilenstein ist vorgegeben. Mit den Betreuern werden regelmässige Sitzungen zur Fortschrittskontrolle durchgeführt.

Abgabetermin Projektmanagementplan:

## Betreuung

**Auftraggeber** HSR / NDS

**Betreuer** A.Kurka, HSR, Zi 6.013

Tel: 055 22 24 682

E-mail: anton.kurka@hsr.ch

Privat : 055 266 19 00

E-mail: akurka@active.ch





## Abstract

Das Ziel der Studienarbeit ist es, eine Simulation eines Fahrzeuges zu erstellen, welches fähig ist, autonom von einem beliebigen Startpunkt in einem Warenhaus zu einem festgelegten Ziel zu fahren. Dabei soll es Hindernisse, welche im Weg sind, erkennen und umfahren. Die Bestimmung des Weges darf nicht durch Ausprobieren erfolgen, sondern soll mit «intelligenten» Methoden aus dem Gebiet der künstlichen Intelligenz berechnet werden.

Um die gestellte Aufgabe erledigen zu können, mussten wir uns zuerst in das Gebiet der künstlichen Intelligenz einarbeiten. Nach einem mehrwöchigen Studium und ersten Testimplementierungen entschieden wir uns, die Wegsuche mit einem Hopfield-Netz, einer Variante eines künstlichen neuronalen Netzes, zu realisieren. Die Fahrzeugkontrolle, genauer das Beschleunigen, Bremsen und Steuern, um dem gefundenen Weg entlang zu fahren, sollte per Fuzzy-Logik erfolgen.

Danach ging es an die Realisierung. Die Nachbildung der Fahrphysik sowie die Simulation der Sensoren zur Hinderniserkennung bildeten die Knackpunkte. Schlussendlich entwickelten wir eine Applikation, welche ein dynamisches Laden von künstlichen neuronalen Netzen ermöglicht; neben unserer Implementierung des Hopfield-Netzes wäre es problemlos möglich, weitere Varianten zu entwickeln.

Die erreichten Resultate sind zufriedenstellend. Das künstliche neuronale Netz findet fast immer den optimalen Weg und das Fahrzeug ist fähig, ihm zu folgen, neue Hindernisse zu erkennen und kann diese an das neuronale Netz weitergeben, um einen angepassten Weg zu erhalten. Das Hauptproblem bildet die Performance; die verschiedenen Tätigkeiten des Hopfield-Netzes, des Fahrzeuges und der Sensoren sind, da sie alle simuliert werden, ziemlich rechenaufwändig.







# Management Summary

## Ausgangslage

In der Lagerhaltung oder in den Montageabläufen eines Industrie-Unternehmens werden oft selbständig agierende Montage- oder Transport-Einheiten gewünscht. Die einfachste Variante einer solchen Transport-Einheit wäre ein Fahrzeug (etwa Hubstaplergrösse), das innerhalb einer Industriehalle selbständig agierend den Weg von A nach B suchen würde. Selbstverständlich müssen allfällige Hindernisse berücksichtigt werden. Ideal wäre natürlich, wenn das Fahrzeug den optimalen Weg finden würde und ohne blindes Ausprobieren, zielgenau auf den Zielpunkt steuern könnte. Hindernisse könnten per Sensor erkannt und umfahren werden.

## Aufgabe

Im Rahmen der Studienarbeit soll eine Simulation erstellt werden, welche das oben beschriebene Problem realitätsnah repräsentiert und löst. Dazu wird eine grafische Benutzeroberfläche erstellt, von welcher aus bequem Simulationen durchgeführt werden können. Um den optimalen Weg zu finden und somit ein intelligentes Verhalten des Fahrzeuges sicherzustellen, werden Methoden aus der künstlichen Intelligenz eingesetzt.

Ziel ist es, das Fahrzeug an einem beliebigen Punkt in einem unbekanntem Lager positionieren zu können und einen Zielpunkt festzulegen. Das Fahrzeug fährt dann autonom vom Startpunkt zum Ziel und erkennt dabei Hindernisse. Diese muss es umfahren, optimalerweise ohne sie zu berühren.

## Vorgehen

In einer ersten Phase haben wir uns in das Gebiet der künstlichen Intelligenz eingearbeitet. Dabei ging es darum, geeignete Techniken und Methoden zu finden, mit welchen die gestellte Aufgabe zu lösen war. Mit Hilfe dieses Grundwissens wurde bestimmt, dass für die Wegsuche ein künstliches neuronales Netz, genauer



ein Hopfield-Netz, eingesetzt werden soll. Ein künstliches neuronales Netz kann als Simulation des Gehirns betrachtet werden – natürlich nur rudimentär und stark eingeschränkt. Ein Hopfield-Netz kann dazu verwendet werden, ein in Abbildung 0.1 dargestelltes Gebilde zu berechnen.

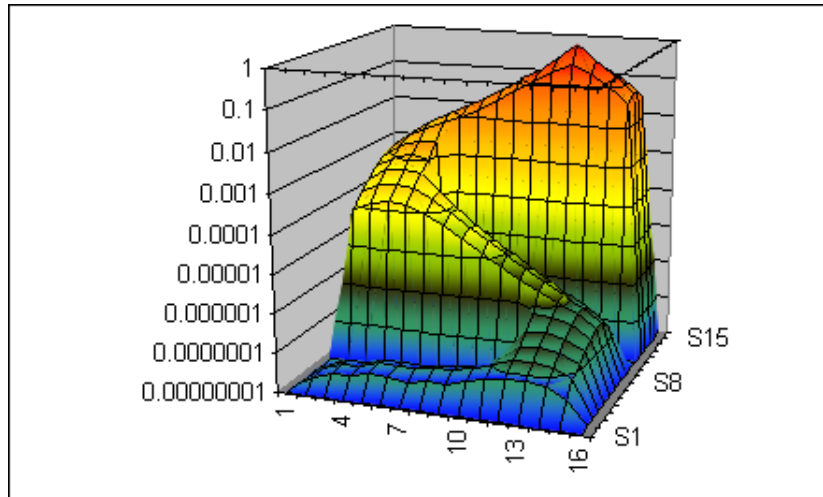


Abbildung 0.1: Resultat eines Hopfield-Netzes

Diese Figur abstrahiert das Warenhaus und gibt den kürzesten Weg zum Ziel vor: Der höchste Punkt ist der Zielpunkt – alles was getan werden muss um zu ihm zu gelangen ist, der stärksten Steigung zu folgen. Die Täler spiegeln die Hindernisse und dürfen nicht durchfahren werden.

Wenn der Weg erst einmal bestimmt ist, soll Fuzzy-Logik verwendet werden, um ihm zu folgen. Fuzzy-Logik ist eine Methode, die im Computer vorherrschende zweiwertige Logik (wahr oder falsch) durch eine mehrwertige Variante zu ersetzen. Damit ist es möglich, dem Fahrzeug nicht nur mitzuteilen, ob es links oder rechts steuern soll, sondern man kann ihm weiche Befehle erteilen: Ein bisschen nach links, stark nach rechts, etc. Auch das Beschleunigen und Bremsen soll mit Fuzzy-Logik realisiert werden – stark beschleunigen oder ein bisschen verlangsamen wären hier Beispiele für mögliche Befehle.

Als erst einmal die Technologien festgelegt und die genauen Anforderungsspezifikationen beschrieben waren, ging es an die Realisierung der Applikation. Beim Erstellen der Software galt es einige Knackpunkte zu lösen: Die Nachbildung eines möglichst realistischen Fahrverhaltens des Fahrzeuges war relativ komplex. So war zum Beispiel zu beachten, dass bei hoher Geschwindigkeit keine zu engen Kurven gefahren werden konnten oder dass das Beschleunigen und Bremsen nicht zu abrupt geschah. Eine weitere Schwierigkeit waren die vielen Tätigkeiten, welche gleichzeitig stattfanden – während das Fahrzeug am Manövrieren ist, führt der Sensor eine Messung aus und das künstliche neuronale Netz berechnet gerade den aktuellen Weg. All das musste aufeinander abgestimmt werden und erforderte ein gut durchdachtes Design. Hinzu kam, dass wir die künstlichen neuronalen



Netze als Plugins implementierten – so dass auch andere Varianten einfach in die Applikation eingebunden werden können.

Als nächster Schritt stand das Testen und Verfeinern der Einstellungen auf dem Programm. Fährt das Fahrzeug wirklich immer in die richtige Richtung? Müsste es in dieser Situation nicht bloss ein wenig nach links steuern, anstatt eine starke Kurve zu fahren? All die vielen Konfigurationsmöglichkeiten machten dieses Unterfangen zu einer richtigen Challenge und nahmen viel Zeit in Anspruch.

Als die Applikation so funktionierte wie sie sollte – Zwischenergebnisse wurden immer mit unserem Betreuer besprochen – ging es im letzten Abschnitt der Arbeit noch darum, die Applikation für die Benutzer verwendbar zu machen. Dazu wurde ein Installations-Assistent erstellt und es wurden Anleitungen für die Anwender geschrieben. Auch allfällige Entwickler, welche das Produkt in der Zukunft weiterentwickeln möchten, wurden berücksichtigt und mit den entsprechenden Dokumentationen ausgestattet.

## Ergebnisse

Entstanden ist eine gut dokumentierte Applikation, welche sich, wie in Abbildung 0.2 dargestellt, präsentiert.

Zu sehen ist die Benutzeroberfläche. Das Fenster oben links dient zum Starten und Stoppen der Simulation sowie zum Starten des Trainings für das künstliche neuronale Netz (falls vom verwendeten Netz benötigt). Im Fenster unten links sieht man das Lager mit fahrendem Fahrzeug: Die grüne Linie repräsentiert den zurückgelegten Weg, die roten Punkte bilden den vorausberechneten Weg. Der rote Kreis ist der Zielpunkt.

Wie man sehen kann, wählt das Fahrzeug einen fast optimalen Weg durch das Lager. Dieses Ziel wurde erreicht. Einziger Negativpunkt ist, dass es Situationen geben kann, in welchen das Fahrzeug stecken bleibt. Dieses Problem der lokalen Minima ist bekannt im Zusammenhang mit neuronalen Netzen, einige Lösungsansätze sind bekannt, konnten im Rahmen dieser Arbeit aber leider nicht umgesetzt werden.

Etwas kritischer ist das Verhalten des Fahrzeuges. Nicht immer kann es die Ideallinie halten und neigt teilweise zum Übersteuern. Nach einer Kurve kann es auch vorkommen, dass sich das Fahrzeug ein wenig einpendeln muss. Dieses Problem ist bedingt durch die Fuzzy-Logik und kann möglicherweise durch weitere Anpassungen an der Konfiguration verringert werden. Da die Konfiguration der Fuzzy-Logik in einem externen Programm passiert, aus welchem anschliessend

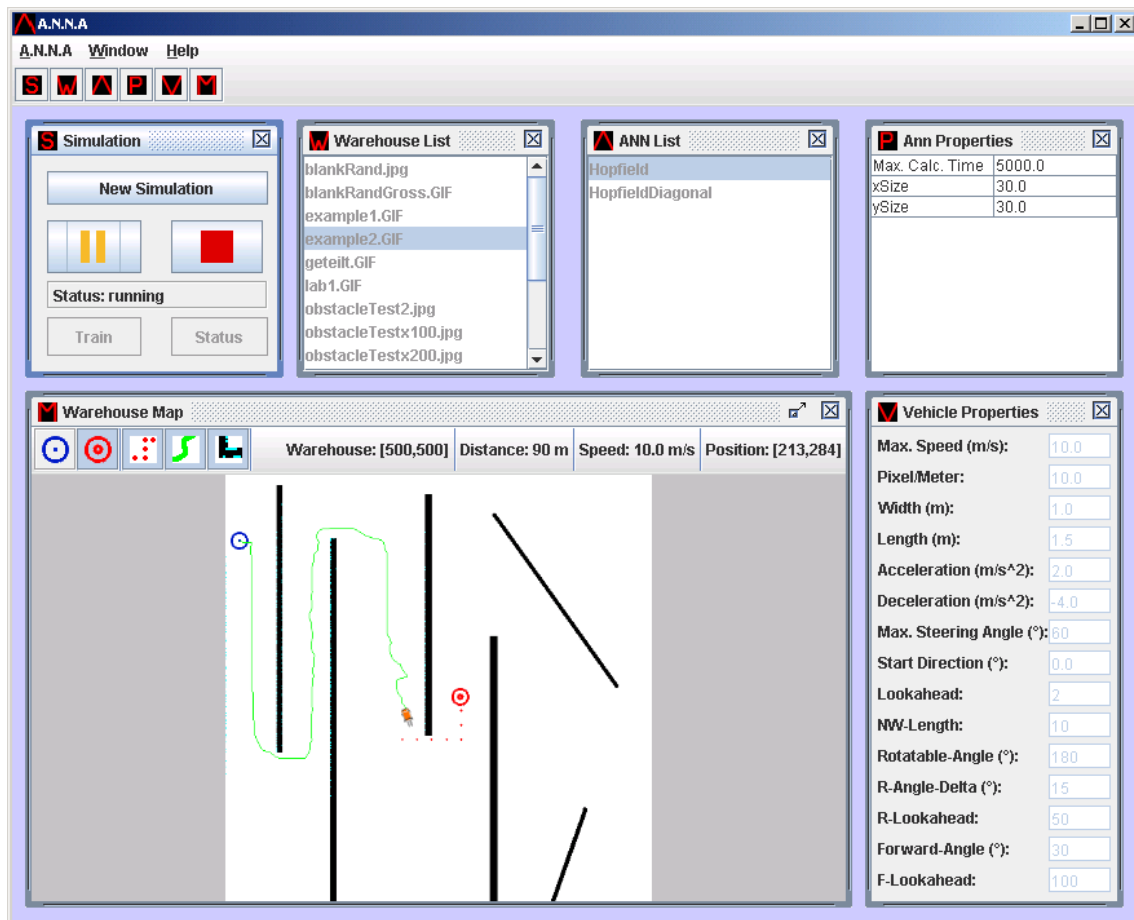


Abbildung 0.2: Screenshot der Applikation

der Programmcode generiert wird, ist das bequem machbar, eine optimale Lösung ist aber fast nicht zu erreichen.

Die Hauptziele konnten erfüllt werden, Verbesserungen sind aber noch möglich. Die Applikation erlaubt eine erste Abschätzung, was mit künstlicher Intelligenz möglich ist; bevor aber ein realer Hubstapler mit dieser Technik ausgerüstet werden kann, sind sicherlich zusätzliche Tests nötig.

Diese Arbeit ermöglichte uns einen Einblick in die künstliche Intelligenz und gab uns die Gelegenheit, einige Methoden davon zu implementieren. Die Möglichkeit, die Technologie praktisch einsetzen zu können, erlaubte es uns, unsere Kenntnisse zu vertiefen und wertvolle Erfahrungen zu sammeln.



## Ausblick

Durch den Einsatz der Plugin-Technik zur Implementierung der künstlichen neuronalen Netze wäre die Applikation zur Erweiterung anderer Arten von Netzen bereit. So könnte man die Leistungsfähigkeit der Netze vergleichen. Zur genauen Analyse müssten ev. zusätzliche Statistiken eingebaut werden.

Beim bestehenden Plugin des Hopfield-Netzes wäre es interessant, eine Lösung für das Problem der lokalen Minima zu finden. Eventuell läge auch eine Steigerung der Performance im Bereich des Möglichen. Um die Performance der gesamten Applikation zu verbessern, wäre auch eine Portierung von Java in eine andere Programmiersprache denkbar.

Ein weiterer Punkt ist die Anpassung der Fuzzy-Logik: Mit einer Verfeinerung der Regeln könnte wahrscheinlich noch eine Verbesserung im Verhalten des Fahrzeuges erreicht werden.





# Inhaltsverzeichnis

|                                       |            |
|---------------------------------------|------------|
| <b>Aufgabenstellung</b>               | <b>iii</b> |
| <b>Abstract</b>                       | <b>vii</b> |
| <b>Management Summary</b>             | <b>ix</b>  |
| <b>I Projektplan</b>                  | <b>1</b>   |
| 1 Projekt Übersicht                   | 3          |
| 2 Projektorganisation                 | 5          |
| 2.1 Projektteam . . . . .             | 5          |
| 2.2 Organisation . . . . .            | 5          |
| 3 Planung                             | 7          |
| 3.1 Zeitplan . . . . .                | 7          |
| 3.2 Projektphasen . . . . .           | 9          |
| 3.3 Meilensteine . . . . .            | 9          |
| 3.4 Arbeitspakete . . . . .           | 9          |
| 4 Risiko Management / Risiko Analyse  | 13         |
| 5 Infrastruktur                       | 17         |
| <b>II Anforderungsspezifikation</b>   | <b>19</b>  |
| 6 Allgemeine Beschreibung             | 21         |
| 6.1 Zweck . . . . .                   | 21         |
| 6.2 Produkt-Perspektive . . . . .     | 21         |
| 6.3 Produkt-Funktion . . . . .        | 21         |
| 6.4 Benutzer Charakteristik . . . . . | 22         |
| 6.5 Einschränkungen . . . . .         | 22         |
| 6.6 Abhängigkeiten . . . . .          | 22         |



|            |                                       |           |
|------------|---------------------------------------|-----------|
| <b>7</b>   | <b>Nichtfunktionale Anforderungen</b> | <b>23</b> |
| 7.1        | Functionality . . . . .               | 23        |
| 7.2        | Usability . . . . .                   | 24        |
| 7.3        | Reliability . . . . .                 | 24        |
| 7.4        | Performance . . . . .                 | 24        |
| 7.5        | Working Environment . . . . .         | 24        |
| 7.6        | Interfaces . . . . .                  | 25        |
| 7.7        | Applicable Standards . . . . .        | 25        |
| <b>8</b>   | <b>User Interface Anforderungen</b>   | <b>27</b> |
| <b>9</b>   | <b>Benutzerrollen</b>                 | <b>29</b> |
| 9.1        | Entwickler . . . . .                  | 29        |
| 9.2        | Anwender . . . . .                    | 29        |
| <b>10</b>  | <b>Use Case Diagram</b>               | <b>31</b> |
| <b>11</b>  | <b>Use Cases</b>                      | <b>33</b> |
| 11.1       | Programm starten / beenden . . . . .  | 33        |
| 11.2       | Simulation durchführen . . . . .      | 34        |
| 11.3       | KNN auswählen . . . . .               | 35        |
| 11.4       | KNN-Parameter einstellen . . . . .    | 35        |
| 11.5       | Lager auswählen . . . . .             | 35        |
| 11.6       | Punkt setzen . . . . .                | 36        |
| 11.7       | Startpunkt setzen . . . . .           | 36        |
| 11.8       | Endpunkt setzen . . . . .             | 37        |
| 11.9       | KNN trainieren . . . . .              | 37        |
| 11.10      | Simulation ausführen . . . . .        | 38        |
| 11.11      | Programm konfigurieren . . . . .      | 38        |
| 11.12      | KNN-Details anzeigen . . . . .        | 38        |
| 11.13      | Weg speichern . . . . .               | 39        |
| 11.14      | KNN speichern . . . . .               | 39        |
| <b>III</b> | <b>Analyse</b>                        | <b>41</b> |
| <b>12</b>  | <b>Domain Model</b>                   | <b>43</b> |
| <b>13</b>  | <b>System Sequenz Diagramme</b>       | <b>47</b> |
| 13.1       | Programm konfigurieren . . . . .      | 47        |
| 13.2       | Simulation durchführen . . . . .      | 47        |
| 13.3       | KNN-Details anzeigen . . . . .        | 49        |
| 13.4       | Weg speichern . . . . .               | 49        |
| 13.5       | KNN speichern . . . . .               | 50        |





|           |                                     |           |
|-----------|-------------------------------------|-----------|
| <b>IV</b> | <b>Architektur &amp; Design</b>     | <b>51</b> |
| 14        | Architektur                         | 53        |
| 15        | Design                              | 55        |
| 15.1      | Persistency . . . . .               | 55        |
| 15.2      | Problem Domain . . . . .            | 55        |
| 15.3      | User Interface . . . . .            | 58        |
| 15.4      | Fahrzeug . . . . .                  | 60        |
| 15.5      | Plugins . . . . .                   | 62        |
| 15.6      | Hopfield Netz . . . . .             | 63        |
| 15.7      | Threads . . . . .                   | 67        |
| 15.8      | Fuzzy-Logik . . . . .               | 70        |
| 15.9      | Koordinatensystem . . . . .         | 71        |
| 16        | Entscheide                          | 73        |
| 16.1      | Design-Entscheide . . . . .         | 73        |
| 16.2      | Software-Entscheide . . . . .       | 75        |
| <b>V</b>  | <b>Resultate &amp; Erkenntnisse</b> | <b>77</b> |
| 17        | Fuzzy-Logik                         | 79        |
| 17.1      | Aufbau . . . . .                    | 79        |
| 17.2      | Übersteuern . . . . .               | 79        |
| 17.3      | Konfiguration . . . . .             | 80        |
| 18        | Hopfield                            | 81        |
| 18.1      | Double . . . . .                    | 81        |
| 18.2      | Lokaler Maximumwert . . . . .       | 81        |
| 18.3      | Performance . . . . .               | 82        |
| 18.4      | Netzdichte . . . . .                | 82        |
| 19        | A.N.N.A                             | 83        |
| 19.1      | Performance . . . . .               | 83        |
| 19.2      | Implementierung . . . . .           | 83        |
| 19.3      | Lookahead des Fahrzeuges . . . . .  | 84        |
| <b>VI</b> | <b>Qualitätssicherung</b>           | <b>85</b> |
| 20        | Methoden zur Qualitätssicherung     | 87        |
| 20.1      | Protokolle . . . . .                | 87        |
| 20.2      | Pair Programming . . . . .          | 87        |
| 20.3      | Coderichtlinien . . . . .           | 87        |
| 20.4      | Eclipse . . . . .                   | 88        |



|             |  |            |
|-------------|--|------------|
| <b>21</b>   | <b>Getestete Ebenen</b>                  | <b>89</b>  |
| 21.1        | Unit Tests . . . . .                     | 89         |
| 21.2        | Ad-hoc Tests . . . . .                   | 89         |
| <b>22</b>   | <b>Management</b>                        | <b>91</b>  |
| 22.1        | Software-Verwaltung . . . . .            | 91         |
| 22.2        | ToDo's & Bug Tracking . . . . .          | 91         |
| 22.3        | Änderungsmanagement . . . . .            | 91         |
| 22.4        | Arbeitszeit . . . . .                    | 91         |
| <b>VII</b>  | <b>Persönliche Erfahrungen</b>           | <b>93</b>  |
| <b>23</b>   | <b>Roland Fehlmann</b>                   | <b>95</b>  |
| <b>24</b>   | <b>Rico Steffen</b>                      | <b>97</b>  |
| <b>VIII</b> | <b>Benutzerhandbuch</b>                  | <b>99</b>  |
| <b>25</b>   | <b>Installationsanleitung</b>            | <b>101</b> |
| <b>26</b>   | <b>Bedienungsanleitung</b>               | <b>103</b> |
| <b>27</b>   | <b>Entwickler-Dokumentation</b>          | <b>109</b> |
| 27.1        | Plugin-Anleitung . . . . .               | 109        |
| 27.2        | Building-Anleitung . . . . .             | 110        |
| 27.3        | Fuzzy-Erweiterungs-Anleitung . . . . .   | 111        |
| <b>IX</b>   | <b>Anhang</b>                            | <b>113</b> |
| <b>A</b>    | <b>Änderungen</b>                        | <b>115</b> |
| A.1         | Projektplan . . . . .                    | 115        |
| A.2         | Anforderungsspezifikation . . . . .      | 116        |
| A.3         | Analyse . . . . .                        | 117        |
| A.4         | Architektur & Design . . . . .           | 118        |
| A.5         | Resultate & Erkenntnisse . . . . .       | 119        |
| A.6         | Qualitätssicherung . . . . .             | 120        |
| A.7         | Persönliche Erfahrungen . . . . .        | 120        |
| A.8         | Benutzerhandbuch . . . . .               | 121        |
| <b>B</b>    | <b>Fuzzy-Logik Einstellungen</b>         | <b>123</b> |
| B.1         | Systemstruktur . . . . .                 | 123        |
| B.2         | Eingangs- & Ausgangs-Variablen . . . . . | 123        |
| B.3         | Regelblöcke . . . . .                    | 128        |



|          |                             |            |
|----------|-----------------------------|------------|
| <b>C</b> | <b>Sitzungsprotokolle</b>   | <b>139</b> |
| C.1      | 03.04.2006 . . . . .        | 139        |
| C.2      | 11.04.2006 . . . . .        | 139        |
| C.3      | 18.04.2006 . . . . .        | 140        |
| C.4      | 25.04.2006 . . . . .        | 141        |
| C.5      | 02.05.2006 . . . . .        | 142        |
| C.6      | 09.05.2006 . . . . .        | 143        |
| C.7      | 16.05.2006 . . . . .        | 144        |
| C.8      | 23.05.2006 . . . . .        | 145        |
| C.9      | 30.05.2006 . . . . .        | 145        |
| C.10     | 06.06.2006 . . . . .        | 146        |
| C.11     | 13.06.2006 . . . . .        | 146        |
| C.12     | 20.06.2006 . . . . .        | 147        |
| C.13     | 27.06.2006 . . . . .        | 148        |
| <b>D</b> | <b>Verwendete Software</b>  | <b>151</b> |
|          | <b>Glossar</b>              | <b>153</b> |
|          | <b>Literaturverzeichnis</b> | <b>155</b> |





# Abbildungsverzeichnis

|       |   |     |
|-------|---|-----|
| 0.1   | Resultat eines Hopfield-Netzes . . . . .            | x   |
| 0.2   | Screenshot der Applikation . . . . .                | xii |
| 3.1   | Projektplan . . . . .                               | 7   |
| 8.1   | erste UI-Skizze . . . . .                           | 27  |
| 8.2   | zweite UI-Skizze . . . . .                          | 28  |
| 10.1  | Use-Case Diagramm . . . . .                         | 31  |
| 12.1  | DomainModel . . . . .                               | 43  |
| 13.1  | SSD-Programm konfigurieren . . . . .                | 47  |
| 13.2  | SSD-Simulation durchführen . . . . .                | 48  |
| 13.3  | SSD-KNN-Details anzeigen . . . . .                  | 49  |
| 13.4  | SSD-Weg speichern . . . . .                         | 49  |
| 13.5  | SSD-KNN speichern . . . . .                         | 50  |
| 14.1  | Architektur . . . . .                               | 53  |
| 15.1  | Persistencey . . . . .                              | 55  |
| 15.2  | Problem Domain . . . . .                            | 56  |
| 15.3  | User Interface . . . . .                            | 59  |
| 15.4  | Fahrzeug von Oben . . . . .                         | 61  |
| 15.5  | Interface für Plugins . . . . .                     | 62  |
| 15.6  | Hopfield . . . . .                                  | 65  |
| 15.7  | Hopfield Grundposition . . . . .                    | 65  |
| 15.8  | Hopfield nach Berechnung . . . . .                  | 66  |
| 15.9  | Hopfield Grundposition mit Hindernissen . . . . .   | 66  |
| 15.10 | Hopfield nach Berechnung mit Hindernissen . . . . . | 67  |
| 15.11 | Threads . . . . .                                   | 68  |
| 15.12 | Fuzzy-Logik . . . . .                               | 70  |
| 15.13 | Lookahead . . . . .                                 | 70  |
| 15.14 | Koordinatensystem . . . . .                         | 72  |
| 17.1  | Fuzzy-Fahrzeugsteuerung . . . . .                   | 80  |
| 25.1  | Installation Schritte 1-3 . . . . .                 | 101 |
| 25.2  | Installation Schritte 4-7 . . . . .                 | 102 |



|      |                                      |     |
|------|--------------------------------------|-----|
| 26.1 | Screenshot der Applikation . . . . . | 103 |
| B.1  | Fuzzy Systemstruktur . . . . .       | 123 |



# Tabellenverzeichnis

|      |  |     |
|------|--|-----|
| 2.1  | Projektteam . . . . .                              | 5   |
| 3.1  | Projekt Management . . . . .                       | 10  |
| 3.2  | Einarbeiten KI . . . . .                           | 10  |
| 3.3  | Requirements . . . . .                             | 10  |
| 3.4  | Analyse . . . . .                                  | 10  |
| 3.5  | Design . . . . .                                   | 11  |
| 3.6  | Implementation . . . . .                           | 11  |
| 3.7  | Test . . . . .                                     | 11  |
| 3.8  | Dokumentation . . . . .                            | 12  |
| 3.9  | Benutzerdokumentation . . . . .                    | 12  |
| 3.10 | Qualitätssicherung . . . . .                       | 12  |
| 4.1  | Arbeitsunfähigkeit oder Beeinträchtigung . . . . . | 13  |
| 4.2  | Ausstieg eines Teammitglieds . . . . .             | 13  |
| 4.3  | Computerausfall . . . . .                          | 14  |
| 4.4  | Datenverlust . . . . .                             | 14  |
| 4.5  | Definierte Anforderungen zu hoch . . . . .         | 14  |
| 4.6  | Unverständliche Dokumentation . . . . .            | 15  |
| 4.7  | Qualitätssicherung . . . . .                       | 15  |
| B.1  | Eingangs-Variablen . . . . .                       | 124 |
| B.2  | Ausgangs-Variablen . . . . .                       | 124 |
| D.1  | Software . . . . .                                 | 151 |





# I

## Projektplan

Verantwortlich: Rico Steffen

| <b>Datum</b> | <b>Version</b> | <b>Änderung</b> |
|--------------|----------------|-----------------|
| 06.04.2006   | 0.1            | Erstellt        |
| 06.04.2006   | 0.2            | Risikoanalyse   |
| 10.04.2006   | 0.3            | Planung         |
| 13.04.2006   | 0.4            | Infrastruktur   |
| 17.04.2006   | 0.5            | Review          |
| 06.07.2006   | 1.0            | Release         |

Der Projektplan umfasst die Planung des Projektes. Eine genauere Auflistung der Änderungen ist im Anhang A.1 zu finden.





# 1 Projekt Übersicht

Bei der Studienarbeit «Autonomes Fahrzeug» geht es darum, eine Simulation eines Lagerfahrzeuges zu erstellen. Das Fahrzeug soll in der Lage sein, autonom zu einem bestimmten Punkt im Lager zu fahren. Dabei muss es selbständig Hindernisse erkennen und umfahren.

## Ziel und Zweck

- Eines der Ziele der Studienarbeit ist es, sich über die verschiedenen künstlichen neuronalen Netze (KNN) zu informieren.
- Ein weiteres Ziel ist es, ein oder mehrere KNN's zu implementieren.
- Es soll eine Software entwickelt werden, welche das Verhalten des autonomen Fahrzeuges visualisiert.





## 2 Projektorganisation

### 2.1 Projektteam

Das Projektteam (Tabelle 2.1) besteht aus zwei Personen und wird von Anton Kurka betreut.

| Mitglied        | Kürzel | E-Mail          |
|-----------------|--------|-----------------|
| Roland Fehlmann | RF     | rfehlman@hsr.ch |
| Rico Steffen    | RS     | rsteffen@hsr.ch |

Tabelle 2.1: Projektteam

### 2.2 Organisation

|                      |  |
|----------------------|--|
| <b>Arbeitszeit</b>   | Es wird jeweils am Montag, Dienstag und Donnerstag Nachmittag gemeinsam am Projekt gearbeitet. Der Zeitpunkt weiterer Arbeiten ist jedem Projektmitglied freigestellt. Die Arbeitszeit wird im Projekt-Wiki erfasst und am Ende der Woche in den Projektplan übernommen. |
| <b>Aufgaben</b>      | Eine grobe Auflistung der Aufgaben ist im Zeitplan (Kapitel 3.1) zu finden. Im Wiki befindet sich eine aktuelle ToDo-Liste, welche laufend angepasst wird.   |
| <b>Dokumentation</b> | Für jeden Teil der Dokumentation ist ein Projektmitglied verantwortlich. Die Änderungen der Dokumententeile werden jeweils auf dem Deckblatt der jeweiligen Teile kurz aufgeführt. Die genauen Änderungen sind im Anhang A auf Seite 115 zu finden.                      |
| <b>Meetings</b>      | Zur Überprüfung des Arbeitsfortschrittes und der Klärung allfälliger Fragen ist ein wöchentliches Meeting mit dem Betreuer vorgesehen.   |





# 3 Planung

## 3.1 Zeitplan

### Grobübersicht

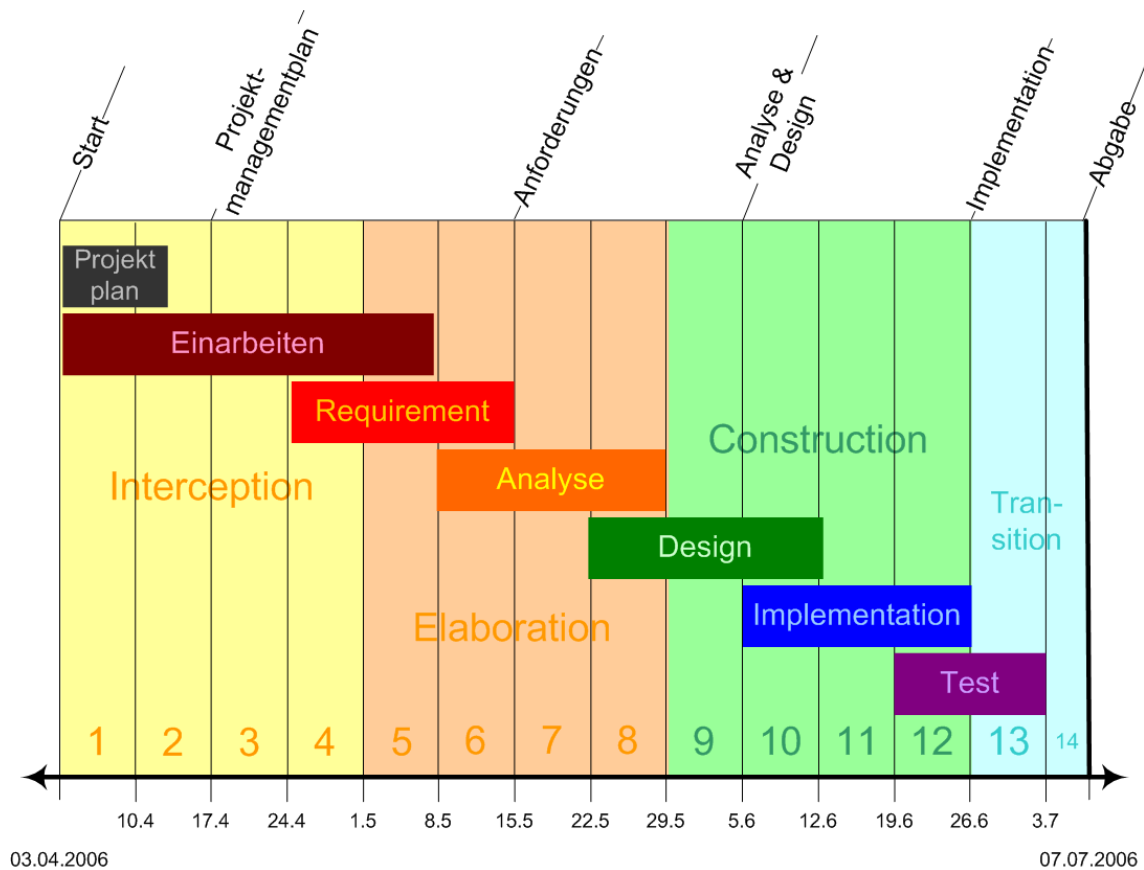


Abbildung 3.1: Projektplan

### Detailzeitplan

Auf der folgenden Seite ist der genaue Zeitplan zu sehen.







## 3.2 Projektphasen

|                     |  |
|---------------------|--|
| <b>Inception</b>    | In der Inception-Phase wird der Projektplan erstellt und mit der Anforderungsspezifikation begonnen. Zusätzlich fällt die Einarbeit in das Gebiet der KNN's in diese Phase.  |
| <b>Elaboration</b>  | Hier findet die Analyse und das Design der zu entwickelnden Simulation statt. Use-Cases und Domainmodels werden erstellt, sowie komplexere Abläufe beschrieben.              |
| <b>Construction</b> | Die eigentliche Implementation wird hier vorgenommen, wobei bereits früh getestet werden muss. Bis zum Ende dieser Phase soll die gesamte Implementation abgeschlossen sein. |
| <b>Transition</b>   | Die Applikation wird getestet und die Leistungsfähigkeit der KNN's wird bewertet. Es sollen alle Teile vollständig dokumentiert und bereit für die Schlussabgabe sein.       |

## 3.3 Meilensteine

Die jeweiligen Daten der Meilensteine sind dem Kapitel 3.1 zu entnehmen.

|                              |  |
|------------------------------|--|
| <b>Projektmanagementplan</b> | Am Ende dieses Meilensteines ist der Projektplan fertiggestellt. Der Zeitplan wird laufend weiter angepasst.   |
| <b>Anforderungen</b>         | Zu diesem Zeitpunkt muss die Anforderungsspezifikation fertiggestellt sein. In dieser wird festgelegt, was im Rahmen der Studienarbeit umgesetzt wird. |
| <b>Analyse &amp; Design</b>  | Abschluss der Analyse und des Designs. Das Design wird im Laufe der Implementation noch verfeinert und verbessert.                                     |
| <b>Implementati-on</b>       | Die Applikation muss, bis auf allfällige Fehlerkorrekturen, zu diesem Meilenstein fertig implementiert sein.   |
| <b>Abgabe</b>                | Die Arbeit am Projekt ist abgeschlossen, die Applikation ist lauffähig und alle Dokumente sind fertiggestellt.   |

## 3.4 Arbeitspakete



### Projekt Management

|                      |                              |
|----------------------|------------------------------|
| Projektplan          | Bearbeitung dieses Dokuments |
| Risikomanagement     | Analysieren der Risiken      |
| Entwicklungsumgebung | Einrichtung der IDE          |

Tabelle 3.1: Projekt Management

### Einarbeiten KI

|               |  |
|---------------|--|
| Allgemein     | Einarbeitung in neuronale Netze            |
| Kohonen-Netz  | Erarbeiten des Stoffes über Kohonen-Netze  |
| Hopfield-Netz | Erarbeiten des Stoffes über Hopfield-Netze |
| Fuzzy-Logik   | Einarbeiten in Fuzzy-Logik                 |

Tabelle 3.2: Einarbeiten KI

### Requirements

|                             |  |
|-----------------------------|--|
| Allgemein                   | Allgemeine Analyse der Requirements              |
| Supplementary Specification | Ergänzende systemweite Anforderungen zum Projekt |
| Use Case Model              | Definieren und erstellen der Use-Cases           |

Tabelle 3.3: Requirements

### Analyse

|                |  |
|----------------|--|
| Allgemein      | Allgemeine Analyse-Arbeiten                |
| Domain Model   | Erstellen des Domain-Models                |
| Plugin Analyse | Analyse für die Erstellung der KNN-Plugins |

Tabelle 3.4: Analyse



## Design

|                  |   |
|------------------|---|
| Allgemein        | Sonstige Design-Arbeiten  |
| Architektur      | Entwurf der Software-Architektur (Layers, Packages)             |
| Design Model     | Erstellen des Design-Modells inklusive Beschreibung der Klassen |
| Schnittstellen   | Beschreibung der Schnittstellen                                 |
| Sequenzdiagramme | Definieren der Abläufe  |
| User Interface   | Entwurf des User Interfaces                                     |
| KNN              | Design der künstlichen neuronalen Netze                         |
| Datenhaltung     | Entscheid über Art der Datenhaltung sowie Design der Daten      |

Tabelle 3.5: Design

## Implementation

|               |  |
|---------------|--|
| Grundgerüst   | Implementation der Grundfunktionen der Applikation |
| Kommunikation | Nachrichtenaustausch zwischen den Schichten        |
| GUI           | Erstellung des GUI's                               |
| KNN           | Entwicklung der KNN's                              |
| Datenhaltung  | Implementierung der Datenhaltung                   |

Tabelle 3.6: Implementation

## Test

|                     |  |
|---------------------|--|
| Usability           | Prüfen auf Benutzerfreundlichkeit        |
| Testen              | Test der Funktionalität                  |
| Performance Analyse | Analyse der Leistungsfähigkeit der KNN's |

Tabelle 3.7: Test



### **Dokumentation**

|                   |  |
|-------------------|--|
| Dokumentation     | Pflegen der Dokumentation                                |
| Projektauswertung | Auswertung des Projektes und Schlussbericht              |
| Codestatistik     | Statistik über den erstellten Programmcode               |
| Meetings          | Wöchentliche Meetings (Erstellen von Protokollen)        |
| Erfahrungsbericht | Verfassung eines persönlichen Berichtes über das Projekt |

Tabelle 3.8: Dokumentation

### **Benutzerdokumentation**

|                        |  |
|------------------------|--|
| Installationsanleitung | Manual zur Installation des Programmes |
| Bedienungsanleitung    | Manual zur Bedienung                   |

Tabelle 3.9: Benutzerdokumentation

### **Qualitätssicherung**

|                      |   |
|----------------------|---|
| Code Walkthrough     | Periodische Codereviews im Team           |
| Dokument Walkthrough | Periodisches Überprüfen der Dokumentation |

Tabelle 3.10: Qualitätssicherung



## 4 Risiko Management / Risiko Analyse

In den Tabellen 4.1 bis 4.7 werden die Risiken aufgezeichnet, welche während des Projektverlaufes auftreten könnten.

|                            |   |
|----------------------------|---|
| <b>Risiko:</b>             | <b>Arbeitsunfähigkeit oder Beeinträchtigung eines Teammitglieds</b>                             |
| <b>Beschreibung:</b>       | Ein Teammitglied wird Arbeitsunfähig oder Beeinträchtigt infolge eines Unfalles oder Krankheit. |
| <b>Wahrscheinlichkeit:</b> | mittel - klein  |
| <b>Auswirkungen:</b>       | sehr gross  |
| <b>Gegenmassnahmen:</b>    | Redundantes Wissen  |
| <b>Folgen:</b>             | Restrukturierung der Arbeitseinteilung. Evtl. Abstriche beim Funktionsumfang.                   |

Tabelle 4.1: Arbeitsunfähigkeit oder Beeinträchtigung

|                            |                                       |
|----------------------------|---------------------------------------|
| <b>Risiko:</b>             | <b>Ausstieg eines Teammitglieds</b>   |
| <b>Beschreibung:</b>       | Ein Teammitglied verlässt das Team    |
| <b>Wahrscheinlichkeit:</b> | extrem klein                          |
| <b>Auswirkungen:</b>       | sehr gross                            |
| <b>Gegenmassnahmen:</b>    | siehe Gegenmassnahmen der Tabelle 4.1 |
| <b>Folgen:</b>             | siehe Folgen der Tabelle 4.1          |

Tabelle 4.2: Ausstieg eines Teammitglieds



|                            |   |
|----------------------------|---|
| <b>Risiko:</b>             | <b>Computerausfall</b>  |
| <b>Beschreibung:</b>       | Ausfall eines Arbeitsrechners.                                |
| <b>Wahrscheinlichkeit:</b> | klein   |
| <b>Auswirkungen:</b>       | gering  |
| <b>Gegenmassnahmen:</b>    | Jedes Teammitglied hat einen Zweit-PC. Backup auf SVN-Server. |
| <b>Folgen:</b>             | Verlust von Daten, welche noch nicht im SVN abgelegt wurden.  |

Tabelle 4.3: Computerausfall

|                            |  |
|----------------------------|--|
| <b>Risiko:</b>             | <b>Datenverlust</b>  |
| <b>Beschreibung:</b>       | Ausfall eines Servers.   |
| <b>Wahrscheinlichkeit:</b> | klein  |
| <b>Auswirkungen:</b>       | mittel   |
| <b>Gegenmassnahmen:</b>    | Tägliches Backup   |
| <b>Folgen:</b>             | Verlust eines Arbeitstages. Mehraufwand für Systemwiederherstellung. |

Tabelle 4.4: Datenverlust

|                            |   |
|----------------------------|---|
| <b>Risiko:</b>             | <b>Definierte Anforderungen zu hoch</b>                                       |
| <b>Beschreibung:</b>       | Unterschätzen der Komplexität   |
| <b>Wahrscheinlichkeit:</b> | klein   |
| <b>Auswirkungen:</b>       | gross   |
| <b>Gegenmassnahmen:</b>    | Gutes Einarbeiten und Analyse der Anforderungsspezifikation mit dem Betreuer. |
| <b>Folgen:</b>             | Mehraufwand in Realisierung oder unvollständige Abgabe                        |

Tabelle 4.5: Definierte Anforderungen zu hoch



|                            |   |
|----------------------------|---|
| <b>Risiko:</b>             | <b>Unverständliche Dokumentation</b>  |
| <b>Beschreibung:</b>       | Die Dokumentation ist nicht oder nur schwer verständlich für Nicht-Teammitglieder.  |
| <b>Wahrscheinlichkeit:</b> | klein   |
| <b>Auswirkungen:</b>       | mittel  |
| <b>Gegenmassnahmen:</b>    | Einfache/klare Formulierungen verwenden, welche für Fachpersonen gut verständlich/nachvollziehbar sind und durchführen von Dokumentenreviews.   |
| <b>Folgen:</b>             | Bei einer allfälligen Erweiterung des Produktes durch einen Aussenstehenden entsteht ein enormer Zeitaufwand, da die Dokumentation nicht weiterhilft und Code analysiert werden muss. |

Tabelle 4.6: Unverständliche Dokumentation

|                            |  |
|----------------------------|--|
| <b>Risiko:</b>             | <b>Qualitätssicherung</b>  |
| <b>Beschreibung:</b>       | Die Qualitätssicherung wird von den Teammitgliedern nicht genügend ernst genommen. Es wird zu wenig Zeit dafür investiert. |
| <b>Wahrscheinlichkeit:</b> | klein  |
| <b>Auswirkungen:</b>       | mittel   |
| <b>Gegenmassnahmen:</b>    | Dokument-/Codereviews  |
| <b>Folgen:</b>             | Die Simulation enthält Bugs/Fehler welche nicht entdeckt werden und sich auf das Verhalten des Fahrzeuges auswirken.       |

Tabelle 4.7: Qualitätssicherung







## 5 Infrastruktur

- Arbeitsrechner** Jedes Teammitglied verfügt über einen von der Schule gestellten Rechner. Zusätzlich besitzt jedes Mitglied ein Notebook.
- Server** Für das Projekt wird ein Server benutzt, welcher von einem Teammitglied privat zur Verfügung gestellt wird. Auf dem Server ist eine Versionskontrolle, ein Webserver und ein Wiki installiert.
- Software** Zum Entwickeln der Applikation wird voraussichtlich Eclipse verwendet [Fou05]. UML Diagramme erstellen wir mit JUDE. Für die Dokumentation wird  $\LaTeX$  [Hel02] [Koh04], JUDE und Microsoft Office verwendet. Eine detaillierte Auflistung der verwendeten Software ist im Anhang D zu finden.
- Organisation / Kommunikation** Das Wiki wird zur Verwaltung von Notizen, Terminen, Ideen usw. verwendet. Die Kommunikation findet über E-Mail oder Instant Messaging statt.



# II

## Anforderungsspezifikation

Verantwortlich: Roland Fehlmann

| <b>Datum</b> | <b>Version</b> | <b>Änderung</b>                   |
|--------------|----------------|-----------------------------------|
| 27.04.2006   | 0.1            | Erstellt                          |
| 01.05.2006   | 0.2            | Use-Cases & UI                    |
| 02.05.2006   | 0.3            | Diverses                          |
| 04.05.2006   | 0.4            | Nichtfunktionale<br>Anforderungen |
| 08.05.2006   | 0.5            | Korrekturen                       |
| 06.07.2006   | 1.0            | Release                           |

Es werden die funktionalen Anforderungen, mit Hilfe von Use Cases, und die nichtfunktionalen Anforderungen an die Simulation beschrieben. Eine genauere Auflistung der Änderungen ist im Anhang A.2 zu finden.





## 6 Allgemeine Beschreibung

### 6.1 Zweck

A.N.N.A<sup>1</sup> soll es ermöglichen, die Leistungsfähigkeit von neuronalen Netzen, beim Finden eines Weges durch eine Lagerhalle, zu untersuchen. Die Software ermöglicht es dem Anwender, unterschiedliche Arten von neuronalen Netzen zu konfigurieren und zu testen. Der gefundene Weg wird anschliessend im Bild des Lagers eingezeichnet.

### 6.2 Produkt-Perspektive

Es wird die Software zur Simulation erstellt. Das Verhalten des Fahrzeuges soll, mit Hilfe von Fuzzy-Logik, möglichst realitätsnah nachgebildet werden. Die verschiedenen KNN's können als Plugins geladen werden. Im Rahmen der Studienarbeit wird ein Plugin für ein Hopfield-Netz implementiert. A.N.N.A soll so realisiert werden, dass weitere KNN's problemlos hinzugefügt werden können.

### 6.3 Produkt-Funktion

Die grundsätzliche Funktionalität ist das Durchführen von Simulationen. Neben den Funktionen zum starten/stoppen einer Simulation werden Funktionen zur Auswahl eines Lagers, zur Auswahl, Konfiguration und zum Training der neuronalen Netze sowie zum Setzen eines Ausgangs- und eines Zielpunktes benötigt. Daneben ist es möglich, die Konfiguration eines neuronalen Netzes oder einen gefundenen Weg zu speichern oder sich die KNN-Details anzeigen zu lassen.

---

<sup>1</sup>A.N.N.A: Artificial Neural Network Application



## 6.4 Benutzer Charakteristik

Die Zielgruppe von A.N.N.A ist relativ gross. Sie reicht vom KI-Interessierten Anwender, welcher eine Anwendung eines neuronalen Netzes ausprobieren will, bis zum versierten Entwickler, welcher ein zusätzliches KNN implementieren möchte. Von dieser Zielgruppe wird erwartet, dass sie zumindest über Erfahrung in der Anwendung von Computern verfügt.

## 6.5 Einschränkungen

Es wird vorausgesetzt, dass das Fahrzeug vom aktuellen Standort zu einem Zielpunkt fährt. Es wird also der kürzeste Weg zwischen zwei Punkten gesucht. Das Finden eines Weges über mehrere Stationen ist nicht geplant. Das Fahrzeug kennt das Lager zu Beginn der Simulation nicht, sondern lernt die Position der Hindernisse nach und nach. Es ist nicht vorgesehen, dass sich die Hindernisse bewegen.

## 6.6 Abhängigkeiten

Da die Software in Java implementiert wird, ist die Funktionstüchtigkeit der installierten Java Virtual Machine Voraussetzung für ein fehlerfreies Ausführen.



## 7 Nichtfunktionale Anforderungen

Die nichtfunktionalen Anforderungen lehnen sich am FURPS+ von Larman [Lar04] an.

### 7.1 Functionality

|                      |  |
|----------------------|--|
| <b>Physik</b>        | <p>Die Fahrphysik des Fahrzeuges sollte in etwa der Realität entsprechen. Dazu kommt ein einfaches mathematisches Modell zum Einsatz. Im Modell werden die folgenden Eigenschaften nachgebildet:</p> <ul style="list-style-type: none"><li>• Ausmasse (Länge <math>\times</math> Breite)</li><li>• Beschleunigung / Verzögerung</li><li>• Kurvenradius</li><li>• Höchstgeschwindigkeit</li></ul> |
| <b>Resultate</b>     | <p>Als Resultat wird einerseits der vom KNN berechnete Weg eingezeichnet, andererseits wird die Strecke, welche das Fahrzeug unter Berücksichtigung der Fahrphysik zurücklegt, markiert. Zudem werden folgende statistischen Daten angezeigt:</p> <ul style="list-style-type: none"><li>• Länge des zurückgelegten Weges</li><li>• Gefundene Hindernisse</li></ul>                               |
| <b>Einstellungen</b> | <p>Einstellungen, welche der Benutzer vornehmen kann:</p> <ul style="list-style-type: none"><li>• Fahrzeugdaten</li><li>• Konfiguration Fuzzy-Logik</li><li>• KNN-Parameter</li></ul>  |
| <b>Errorhandling</b> | <p>Der Benutzer soll wenn möglich vor Fehleingaben bewahrt werden. Allfällige Fehler müssen dem Benutzer mit einer aussagekräftigen Fehlermeldung angezeigt werden.</p>  |



## 7.2 Usability

- Erlernbarkeit** A.N.N.A soll intuitiv zu bedienen sein. Ein Benutzer, welcher ähnliche Anwendungen bedienen kann, sollte sich rasch zurechtfinden können.
- Bedienbarkeit** Um A.N.N.A sinnvoll verwenden zu können, sollte der Benutzer über rudimentäre Kenntnisse im Bereich der neuronalen Netze verfügen.

## 7.3 Reliability

- Fehlertoleranz** Die Applikation soll dem Verhalten der KNN gegenüber fehlertolerant sein. Versagt ein KNN bei der Berechnung oder tritt beim Training des Netzes ein Fehler auf, darf die Stabilität der Applikation nicht gefährdet sein.

## 7.4 Performance

- Zeitverhalten** Das User Interface der Applikation soll jederzeit bedienbar sein und soweit als möglich verzögerungsfrei reagieren.

## 7.5 Working Environment

- Hardware** A.N.N.A sollte auf folgender Hardware-Konfiguration performant lauffähig sein:
- Prozessor: Intel Pentium 4 oder vergleichbar
  - Memory: 512 MB
- Software**
- A.N.N.A wird unter Microsoft Windows XP Professional SP2 entwickelt und getestet. Prinzipiell sollte es jedoch unter jedem Betriebssystem, für welches eine entsprechende Java-VM zur Verfügung steht, lauffähig sein. Dies wird jedoch nicht überprüft.





- Es wird das Java™ 2 Runtime Environment, Standard Edition (build 1.5.0\_07) von Sun verwendet.

## 7.6 Interfaces

**Benutzerschnittstelle** Der Benutzer kommuniziert über ein grafisches User Interface sowie mit Maus und Tastatur mit dem Programm.

**Softwareschnittstelle** A.N.N.A bietet ein Plugin-System zur Entwicklung weiterer KNN-Plugins.

## 7.7 Applicable Standards

**Dateiformat** Als Inputdatei für eine Lagerhalle können Standardgrafikformate (jpeg, gif, png) verwendet werden.





## 8 User Interface Anforderungen

Die folgenden Skizzen (8.1, 8.2) stellen erste Ideen eines User Interfaces dar. Das schlussendlich realisierte User Interface wird erst in der Designphase definiert und muss nicht zwingend nach den folgenden Skizzen umgesetzt werden. Die Skizzen dienen jedoch zur besseren Verständlichkeit der Use Cases.



Abbildung 8.1: erste UI-Skizze



Abbildung 8.2: zweite UI-Skizze



## 9 Benutzerrollen

Es werden grundsätzlich zwei verschiedene Benutzerrollen unterschieden. Diese werden in den folgenden Abschnitten genauer beschrieben.

### 9.1 Entwickler

**Benutzte Funktionen** Alle im Kapitel 11 beschriebenen Use Cases. Zusätzlich arbeitet der Entwickler mit der Plugin-Schnittstelle von A.N.N.A.

**Spektrum** Softwareingenieure, Softwareentwickler, Programmierer

**Erfahrung**

- Objektorientiertes Design und Programmieren
- Java
- KNN Implementierung

**Ausbildung für Benutzung** Der Entwickler sollte in der Bedienung von A.N.N.A vertraut sein. Das dazu nötige Know-How findet er im Benutzerhandbuch. Für die Pluginentwicklung muss er sich zusätzlich in die Beschreibung der Schnittstelle einlesen und sich mit der Programmarchitektur vertraut machen.

### 9.2 Anwender

**Benutzte Funktionen** Alle im Kapitel 11 beschriebenen Use Cases.

**Spektrum** PC-Anwender

**Erfahrung**

- KNN Grundlagenwissen



**Ausbildung für Benutzung** Der Anwender kann sich mit dem Benutzerhandbuch alle benötigten Kenntnisse beibringen.



# 10 Use Case Diagram

In der Abbildung 10.1 wird ein Überblick über die Use-Cases gegeben.

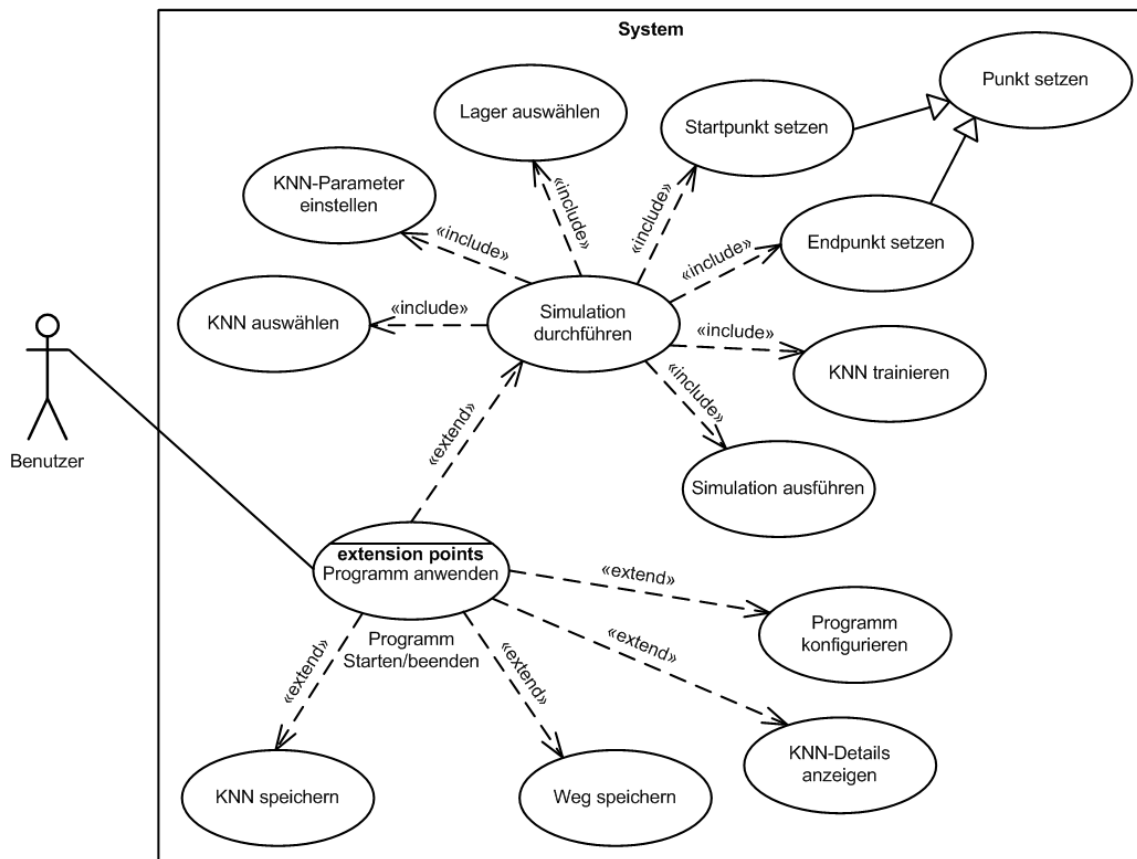


Abbildung 10.1: Use-Case Diagramm







# 11 Use Cases

Der *Primary Actor* aller nachfolgend beschriebenen Use Cases ist immer der Benutzer von A.N.N.A.

## Notation der Use Cases

Die Darstellung und Beschreibung der Use Cases sind an Larman [Lar04] angelehnt.

|                                   |  |
|-----------------------------------|--|
| <b>Stakeholders and Interests</b> | Beschreibt, wer den Use Case ausführt und welche Interessen er hat.  |
| <b>Preconditions</b>              | Definiert die Bedingungen, welche der Stakeholder erfüllen muss, um den Use Case durchführen zu können.                |
| <b>Postconditions</b>             | Beschreibt, wie die Situation aussieht, nachdem der Use Case abgeschlossen wurde.                                      |
| <b>Basic flow</b>                 | Der normale Ablauf des Use Cases.<br>Optionale Punkte stehen in [eckigen Klammern].                                    |
| <b>Alternative flows</b>          | Falls dem Stakeholder bei einzelnen Schritten andere Möglichkeiten zur Verfügung stehen, werden diese hier aufgezeigt. |
| <b>Freq. of Occur.</b>            | Frequency of Occurrence ist die Häufigkeit des Auftretens des Use Cases.   |
| <b>Further notes</b>              | Weitere Anmerkungen und Ergänzungen zum Use Case.  |

## 11.1 Programm starten / beenden

|                                   |   |
|-----------------------------------|---|
| <b>Stakeholders and Interests</b> | Der Benutzer möchte mit A.N.N.A arbeiten. |
| <b>Preconditions</b>              | A.N.N.A ist korrekt installiert.          |



|                          |  |
|--------------------------|--|
| <b>Postconditions</b>    | A.N.N.A läuft und der Benutzer kann damit arbeiten.<br>A.N.N.A ist beendet.  |
| <b>Basic flow</b>        | <ol style="list-style-type: none"><li>1. Benutzer gibt den Befehl, A.N.N.A auszuführen.</li><li>2. A.N.N.A startet.</li><li>3. Benutzer arbeitet mit A.N.N.A<br/>«Extension Point: Programm anwenden».</li><li>4. Benutzer wählt die Funktion <i>Beenden</i>.</li><li>5. A.N.N.A wird beendet.</li></ol>                                   |
| <b>Alternative flows</b> | <ol style="list-style-type: none"><li>5a) Der gefundene Weg oder die Netzkonfiguration wurden noch nicht gespeichert.<ol style="list-style-type: none"><li>i. System fragt Benutzer, ob er speichern möchte.</li><li>ii. Benutzer hat die Wahl zwischen <i>Speichern</i>, <i>Nicht speichern</i> und <i>Abbrechen</i>.</li></ol></li></ol> |
| <b>Freq. of Occur.</b>   | Gelegentlich   |

## 11.2 Simulation durchführen

|                                   |  |
|-----------------------------------|--|
| <b>Stakeholders and Interests</b> | Der Benutzer möchte eine Simulation durchführen.   |
| <b>Preconditions</b>              | A.N.N.A wurde gestartet.   |
| <b>Postconditions</b>             | Resultat der Simulation wird angezeigt.  |
| <b>Basic flow</b>                 | <ol style="list-style-type: none"><li>1. UC<sup>1</sup> 11.3 «KNN auswählen»</li><li>2. UC 11.4 «KNN-Parameter einstellen»</li><li>3. UC 11.5 «Lager auswählen»</li><li>4. UC 11.7 «Startpunkt setzen»</li><li>5. UC 11.8 «Endpunkt setzen»</li><li>6. [UC 11.9 «KNN trainieren»]</li><li>7. UC 11.10 «Simulation ausführen»</li></ol> |
| <b>Freq. of Occur.</b>            | Gelegentlich   |
| <b>Further Note</b>               | Reihenfolge des Durchführens der Use Cases teilweise egal.   |

<sup>1</sup>UC: Use Case



## 11.3 KNN auswählen

|                                   |  |
|-----------------------------------|--|
| <b>Stakeholders and Interests</b> | Benutzer möchte festlegen, mit welchem KNN der Weg berechnet werden soll.  |
| <b>Preconditions</b>              | A.N.N.A ist gestartet.   |
| <b>Postconditions</b>             | KNN wurde geladen. Einstellbare Parameter werden angezeigt.  |
| <b>Basic flow</b>                 | <ol style="list-style-type: none"><li>1. Benutzer wählt KNN aus Liste aus.</li><li>2. Parameter werden angezeigt.</li><li>3. [Konfiguration laden.]</li></ol>  |
| <b>Alternative flows</b>          | <ol style="list-style-type: none"><li>3a) Eine bereits geänderte Konfiguration wurde nicht gespeichert.<ol style="list-style-type: none"><li>i. System fragt Benutzer, ob er speichern möchte.</li><li>ii. Benutzer hat die Wahl zwischen <i>Speichern</i>, <i>Nicht speichern</i> und <i>Abbrechen</i>.</li></ol></li></ol> |
| <b>Freq. of Occur.</b>            | Gelegentlich   |

## 11.4 KNN-Parameter einstellen

|                                   |   |
|-----------------------------------|---|
| <b>Stakeholders and Interests</b> | Benutzer möchte das geladene KNN parametrisieren.   |
| <b>Preconditions</b>              | Ein KNN wurde geladen.  |
| <b>Postconditions</b>             | Das KNN ist konfiguriert.   |
| <b>Basic flow</b>                 | <ol style="list-style-type: none"><li>1. Benutzer ändert die Standardparameter oder eine vorhandene Konfiguration.</li><li>2. [Benutzer speichert die geänderte Konfiguration.]</li></ol> |
| <b>Freq. of Occur.</b>            | Gelegentlich  |

## 11.5 Lager auswählen

|                                   |   |
|-----------------------------------|---|
| <b>Stakeholders and Interests</b> | Benutzer möchte das Lager auswählen, für welches die Simulation durchgeführt werden soll. |
|-----------------------------------|---|



|                          |  |
|--------------------------|--|
| <b>Preconditions</b>     | A.N.N.A ist gestartet.   |
| <b>Postconditions</b>    | Ein Lager wurde geladen.   |
| <b>Basic flow</b>        | 1. Benutzer wählt das Lager, in welchem die Simulation durchgeführt werden soll. |
| <b>Alternative flows</b> | 1a) Der Benutzer lädt ein Bild in die Liste der vorhandenen Lager.               |
| <b>Freq. of Occur.</b>   | Gelegentlich   |

## 11.6 Punkt setzen

|                                   |  |
|-----------------------------------|--|
| <b>Stakeholders and Interests</b> | Benutzer möchte einen bestimmten Punkt im Lager setzen.  |
| <b>Preconditions</b>              | Ein Lager ist ausgewählt.  |
| <b>Postconditions</b>             | Der Punkt ist im Lager gesetzt.  |
| <b>Basic flow</b>                 | 1. Benutzer wählt die gewünschte Punktart aus.<br>2. Der Benutzer setzt den Punkt im Lager.<br>3. Der Punkt wird im Lager angezeigt. |
| <b>Freq. of Occur.</b>            | Gelegentlich   |

## 11.7 Startpunkt setzen

|                                   |  |
|-----------------------------------|--|
| <b>Stakeholders and Interests</b> | Benutzer möchte einen Startpunkt setzen.   |
| <b>Preconditions</b>              | Ein Lager ist ausgewählt.  |
| <b>Postconditions</b>             | Der Startpunkt ist gesetzt.  |
| <b>Basic flow</b>                 | 1. Benutzer führt den Use Case 11.6 «Punkt setzen» durch und wählt dabei den Startpunkt als zu bestimmenden Punkt. |
| <b>Freq. of Occur.</b>            | Gelegentlich   |



## 11.8 Endpunkt setzen

|                                   |  |
|-----------------------------------|--|
| <b>Stakeholders and Interests</b> | Benutzer möchte einen Endpunkt setzen.   |
| <b>Preconditions</b>              | Ein Lager ist ausgewählt.  |
| <b>Postconditions</b>             | Der Endpunkt ist gesetzt.  |
| <b>Basic flow</b>                 | 1. Benutzer führt den Use Case 11.6 «Punkt setzen» durch und wählt dabei den Endpunkt als zu bestimmenden Punkt. |
| <b>Freq. of Occur.</b>            | Gelegentlich   |

## 11.9 KNN trainieren

|                                   |   |
|-----------------------------------|---|
| <b>Stakeholders and Interests</b> | Ein KNN soll für die Wegfindung trainiert werden.   |
| <b>Preconditions</b>              | Ein KNN wurde geladen und die Parameter eingestellt.  |
| <b>Postconditions</b>             | Das KNN hat sich so konfiguriert, dass es die Aufgabe lösen kann.   |
| <b>Basic flow</b>                 | <ol style="list-style-type: none"><li>1. Benutzer legt die Anzahl der Trainingsschritte fest.</li><li>2. Der Benutzer startet das Training.</li><li>3. System zeigt Lernkurve an.</li><li>4. System meldet, dass das Training durchgeführt wurde.</li></ol>   |
| <b>Alternative flows</b>          | <ol style="list-style-type: none"><li>3a) Es wurde ein Trainingsstand erreicht, welcher eine gewisse Abweichung unterschreitet.<ol style="list-style-type: none"><li>i. System fragt den Benutzer, ob das Training beendet werden soll.</li><li>ii. Benutzer kann das Training beenden oder weiterlaufen lassen.</li></ol></li><li>3b) Das Netz konvergiert nach einer festgelegten Zeit nicht.<ol style="list-style-type: none"><li>i. System fragt den Benutzer, ob er den aktuellen Trainingsstand beibehalten möchte oder den Ausgangszustand laden will.</li><li>ii. Benutzer wählt die gewünschte Option.</li></ol></li></ol> |
| <b>Freq. of Occur.</b>            | Gelegentlich  |



## 11.10 Simulation ausführen

|                                   |   |
|-----------------------------------|---|
| <b>Stakeholders and Interests</b> | Benutzer will die Simulation ausführen.   |
| <b>Preconditions</b>              | KNN und Lager wurden geladen, Start- und Endpunkt gesetzt. KNN ist konfiguriert und trainiert.  |
| <b>Postconditions</b>             | Die Simulation wurde durchgeführt, der gefundene Weg ist im Lager eingezeichnet und die Statistik wird angezeigt.   |
| <b>Basic flow</b>                 | <ol style="list-style-type: none"><li>1. Benutzer startet die Simulation.</li><li>2. System zeigt den gefundenen Weg an.</li></ol>  |
| <b>Alternative flows</b>          | <ol style="list-style-type: none"><li>2a) Benutzer unterbricht die Simulation zwischenzeitlich.<ol style="list-style-type: none"><li>i. Der Benutzer pausiert die Simulation.</li><li>ii. Benutzer setzt die Simulation fort.</li></ol></li></ol> |
| <b>Freq. of Occur.</b>            | Gelegentlich  |

## 11.11 Programm konfigurieren

|                                   |  |
|-----------------------------------|--|
| <b>Stakeholders and Interests</b> | Benutzer möchte die Applikation an seine Bedürfnisse anpassen.   |
| <b>Preconditions</b>              | A.N.N.A wurde gestartet.   |
| <b>Postconditions</b>             | Die Konfiguration von A.N.N.A wurde geändert.  |
| <b>Basic flow</b>                 | <ol style="list-style-type: none"><li>1. Benutzer wählt <i>Optionen</i>.</li><li>2. System zeigt Konfigurationsmöglichkeiten an.</li><li>3. Benutzer ändert Konfigurationspunkte und bestätigt seine Änderungen.</li></ol> |
| <b>Freq. of Occur.</b>            | Selten   |

## 11.12 KNN-Details anzeigen

|                                   |   |
|-----------------------------------|---|
| <b>Stakeholders and Interests</b> | Benutzer möchte die genauen Parameter des Netzes anzeigen – zum Beispiel die Verbindungsgewichte. |
| <b>Preconditions</b>              | KNN wurde geladen, ev. trainiert.   |



**Postconditions** Keine

**Basic flow**

1. Benutzer wählt *KNN-Details*.
2. System zeigt die Informationen an.

**Freq. of Occur.** Selten

## 11.13 Weg speichern

**Stakeholders and Interests** Benutzer möchte den gefundenen Weg speichern.

**Preconditions** Eine Simulation wurde erfolgreich durchgeführt.

**Postconditions** Weg ist gesichert.

**Basic flow**

1. Benutzer speichert den gefundenen Weg.

**Freq. of Occur.** Selten

## 11.14 KNN speichern

**Stakeholders and Interests** Benutzer möchte Parameter des KNN's sichern.

**Preconditions** KNN wurde geladen und konfiguriert, ev. trainiert.

**Postconditions** KNN ist gesichert.

**Basic flow**

1. Benutzer speichert das KNN.

**Freq. of Occur.** Selten





# III

## Analyse

Verantwortlich: Roland Fehlmann

| <b>Datum</b> | <b>Version</b> | <b>Änderung</b> |
|--------------|----------------|-----------------|
| 08.05.2006   | 0.1            | Erstellt        |
| 09.05.2006   | 0.2            | Domain Model    |
| 09.05.2006   | 0.3            | SSD             |
| 11.05.2006   | 0.4            | Allgemein       |
| 16.05.2006   | 0.5            | Position        |
| 06.07.2006   | 1.0            | Release         |

Dieser Dokumentteil behandelt den objektorientierten Anteil der Analyse und umfasst das Domain Model und Systemsequenz-Diagramme. Die Problem-domain wird einer methodischen Analyse unterzogen. Eine genauere Auflistung der Änderungen ist im Anhang A.3 zu finden.







## Gelb

**Applikation** Die Applikation als Ganzes, welches alles verwaltet.

## Rot

**Hopfield-Netz** Ein spezifisches neuronales Netz.

**KNN** Repräsentiert die Grundlage eines künstlichen neuronalen Netzes.

**Lernkurve** Die Lernkurve des neuronalen Netzes.

**Neuron** Stellt ein einzelnes Neuron innerhalb eines KNN dar.

**Plugin** Ein Plugin für die Applikation, welches ein KNN enthält.

**Verbindung** Stellt eine Verbindung zwischen zwei Neuronen dar.

## Hellblau

**Fahrzeug** Stellt das Lagerfahrzeug dar, welches den Weg suchen muss.

**Position** Die aktuelle Position des Fahrzeuges.

**Sensor** Ein einzelner Sensor des Fahrzeuges.

## Orange

**Punkt** Ein Punkt in einem Lager in einer Simulation.

**Startpunkt** Der Startpunkt des Fahrzeuges in der Simulation.

**Zielpunkt** Der zu erreichende Zielpunkt des Fahrzeuges.



## Hellgrün

- Simulation** Die Simulation der Wegsuche des Fahrzeuges durch das Lager zum Zielpunkt.
- Zeiterfassung** Misst die Zeit, welche das Fahrzeug benötigt, bis es das Ziel erreicht hat.

## Dunkelgrün

- Istweg** Der wirklich gefahrene Weg des Fahrzeuges durch das Lager.
- Sollweg** Der berechnete Sollweg des Fahrzeuges durch das Lager.
- Weg** Ein Weg durch ein Lager.

## Dunkelblau

- Hindernis** Ein Hindernis im Lager.
- Lager** Das eigentliche Lager in der Simulation.
- Statisches Hindernis** Ein unbewegliches Hindernis im Lager.

## Violett

- Bewegliches Hindernis** Ein sich bewegendes Hindernis im Lager (wird nicht implementiert).





## 13 System Sequenz Diagramme

In den folgenden Abschnitten werden die wichtigsten System Sequenz Diagramme dargestellt. Sie zeigen die vom Benutzer durchgeführten Systemaufrufe und die entsprechenden Reaktionen des Systems.

### 13.1 Programm konfigurieren

Im Sequenzdiagramm 13.1 konfiguriert der Benutzer die Applikation.

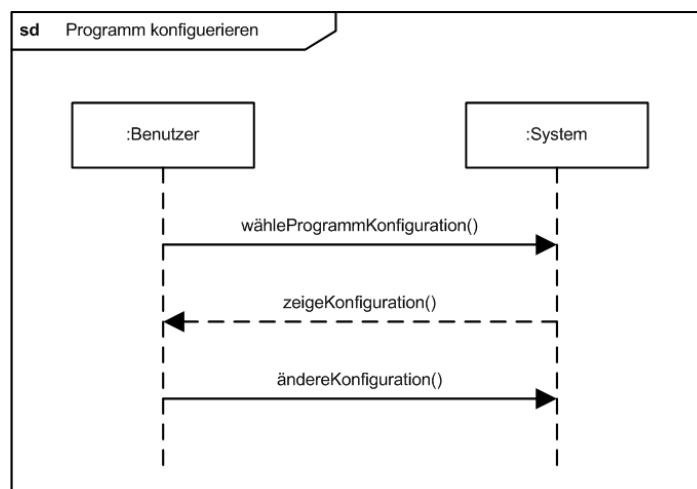


Abbildung 13.1: SSD-Programm konfigurieren

### 13.2 Simulation durchführen

Abbildung 13.2 zeigt den Ablauf der Durchführung einer Simulation.

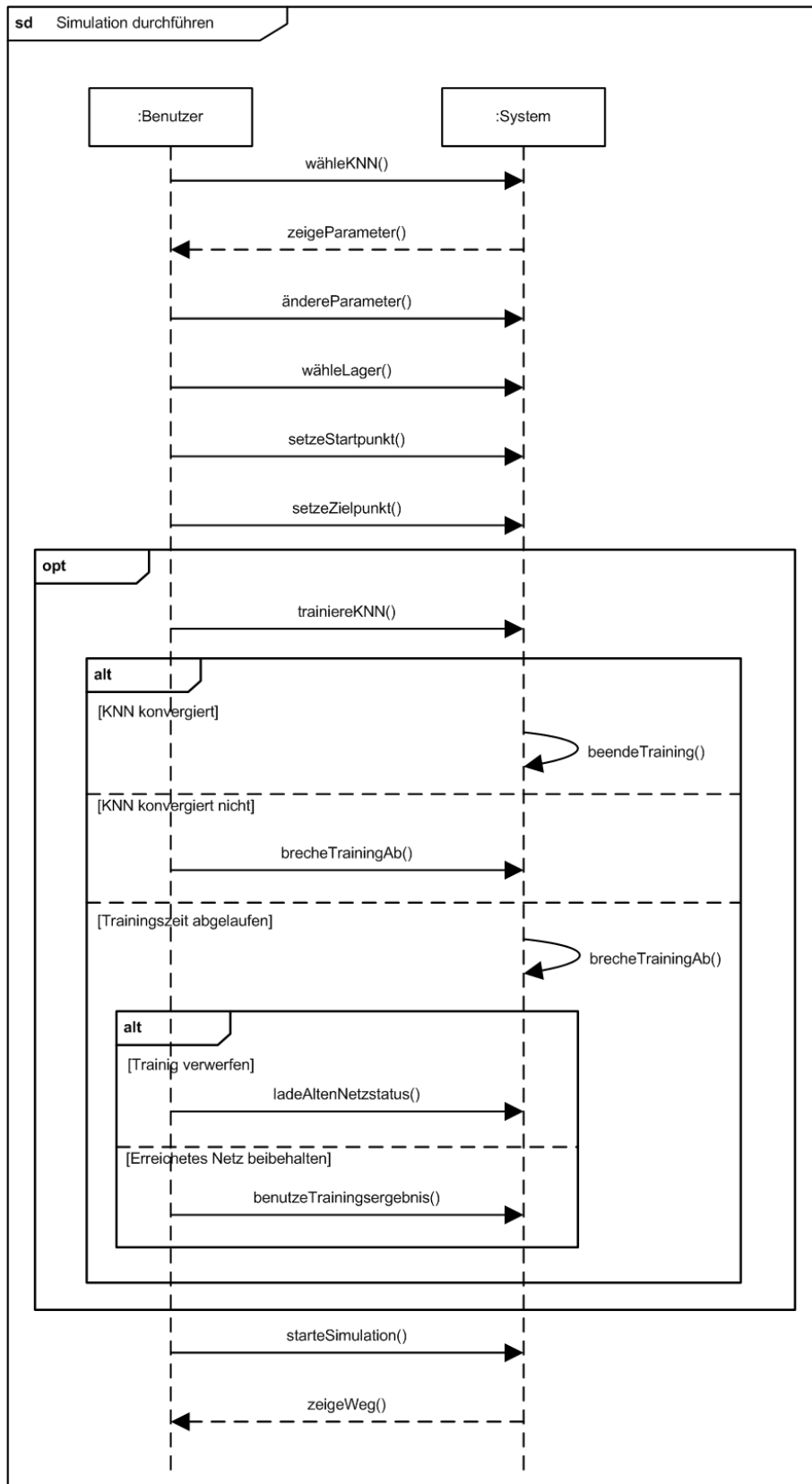


Abbildung 13.2: SSD-Simulation durchführen





## 13.3 KNN-Details anzeigen

In Abbildung 13.3 wird die Aktion angezeigt, welche durchgeführt wird, wenn der Benutzer die Details des KNN anzeigen möchte.

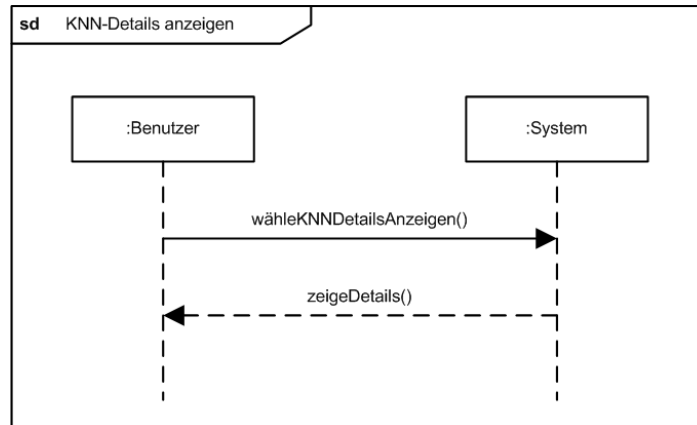


Abbildung 13.3: SSD-KNN-Details anzeigen

## 13.4 Weg speichern

Diagramm 13.4 zeigt, wie der Benutzer den gefundenen Weg abspeichert.

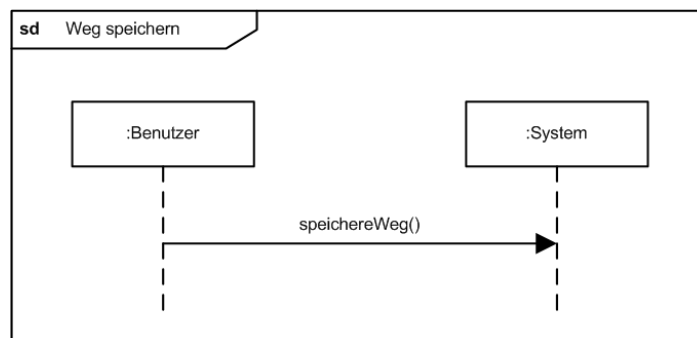


Abbildung 13.4: SSD-Weg speichern



## 13.5 KNN speichern

Wie der Benutzer das KNN speichert, sieht man in Abbildung 13.5.

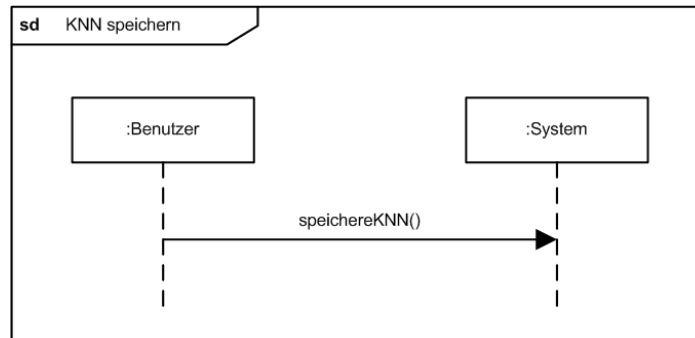


Abbildung 13.5: SSD-KNN speichern

# IV

## Architektur & Design

Verantwortlich: Rico Steffen

| <b>Datum</b> | <b>Version</b> | <b>Änderung</b> |
|--------------|----------------|-----------------|
| 14.05.2006   | 0.1            | Erstellt        |
| 15.05.2006   | 0.2            | Architektur     |
| 30.05.2006   | 0.3            | Erstes Design   |
| 05.06.2006   | 0.4            | Design UI       |
| 15.06.2006   | 0.5            | Design          |
| 23.06.2006   | 0.6            | Design          |
| 30.06.2006   | 0.7            | Fahrzeug        |
| 01.07.2006   | 0.8            | Threads         |
| 03.07.2006   | 0.9            | Entscheide      |
| 06.07.2006   | 1.0            | Release         |

Dieser Dokumentteil beschreibt die Architektur, das Detaildesign und externe Design der Applikation. Es legt fest, wie die Applikation zu implementieren ist und erklärt komplexere Zusammenhänge. Eine genauere Auflistung der Änderungen ist im Anhang A.4 zu finden.





# 14 Architektur

Abbildung 14.1 zeigt eine Übersicht der Architektur. Anschliessend werden die einzelnen Pakete erläutert.

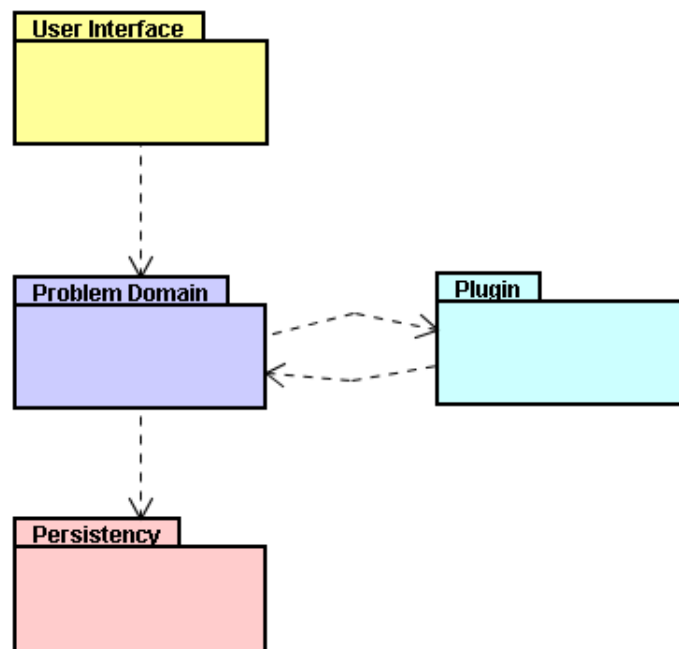


Abbildung 14.1: Architektur

## User Interface

Das User Interface bildet die Schnittstelle zum Benutzer. Alle vom User durchführbaren Aktionen werden über das User Interface gestartet und zur Verarbeitung an die Problem Domain weitergegeben. Die erhaltenen Resultate werden dann wieder im User Interface dargestellt.



## Problem Domain

Die Problem Domain beinhaltet die eigentliche Programmfunktionalität. Ihre Hauptaufgabe ist das managen der Simulation. Sie muss das Fahrzeug, das Lager, Start- und Zielpunkt sowie die gefundenen Wege verwalten. Das eigentliche Berechnen des Weges wird an das jeweilige KNN-Plugin delegiert.

## Plugin

Ein Plugin repräsentiert ein neuronales Netz. Das neuronale Netz kommt bei der Berechnung des optimalen Weges durch das Lager zum Einsatz.

## Persistency

Die Datenhaltung dient zum Laden von KNN-Plugins und Warenhäusern. Ebenfalls können über die Persistency Bilder von Warenhäusern gespeichert werden.



# 15 Design

## 15.1 Persistency

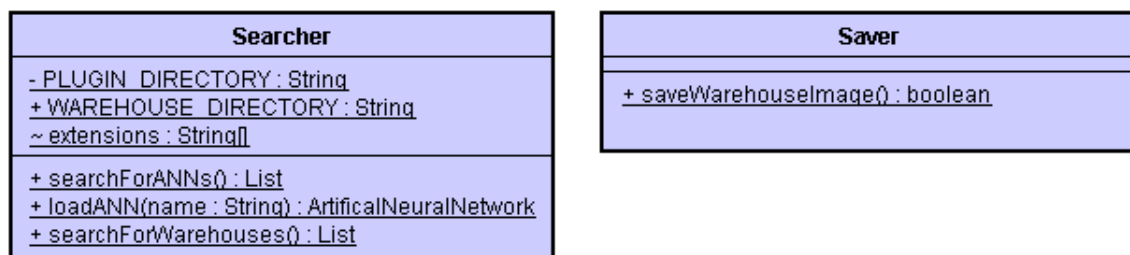


Abbildung 15.1: Persistency

Die Datenhaltung besteht, wie in der Abbildung 15.1 zusehen ist, nur aus zwei Klassen. Der Searcher stellt statische Methoden zur Verfügung, um Warehäuser und KNN-Plugins vom Dateisystem zu laden. Dies geschieht während des Startens der Applikation. Der Saver wird verwendet, um Bilder des Warenhauses inkl. Fahrzeug, Weg und detektierten Hindernissen zu speichern.

## 15.2 Problem Domain

Die Problem Domain ist in der Abbildung 15.2 dargestellt. Sie lässt sich in folgende Gruppen unterteilen:

### Blau

**Point** Die Klasse Point repräsentiert einen einfachen Bildpunkt. Neben Methoden zum Abfragen der Koordinaten verfügt ein Punkt über die Möglichkeit, die Euklidische Distanz zu einem anderen Punkt zu berechnen. Des weiteren implementiert er, zur einfachen Sortierbarkeit, das Comparable-Interface.

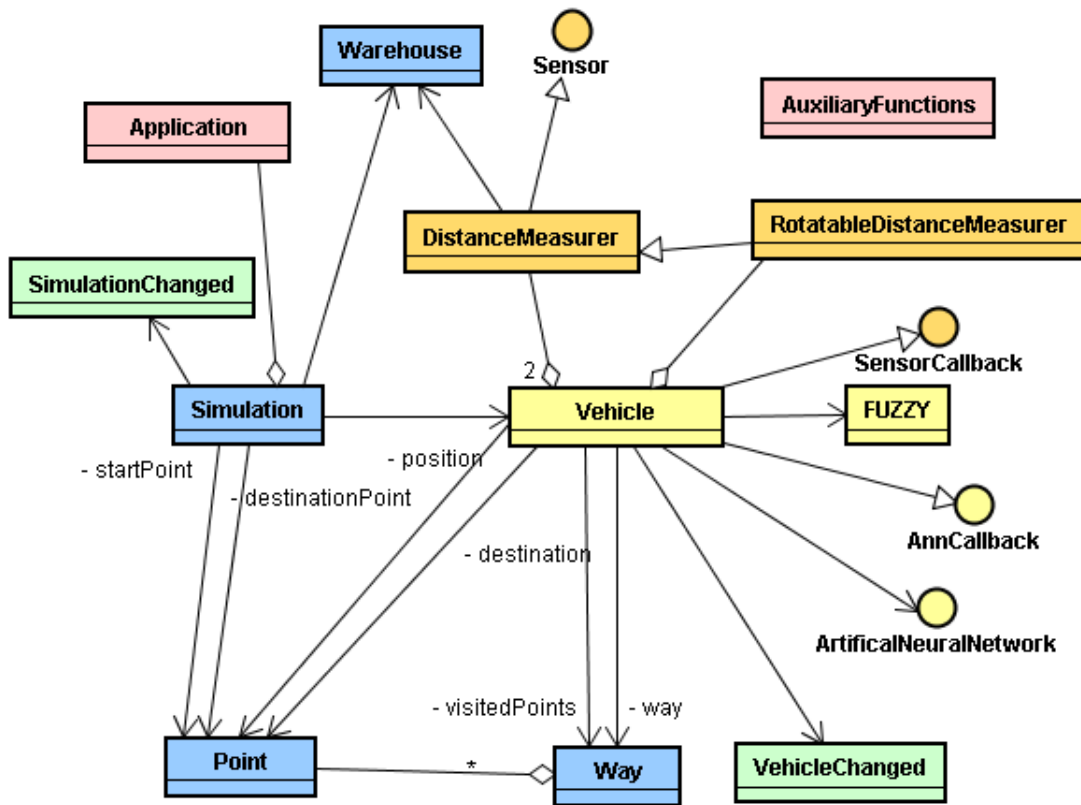


Abbildung 15.2: Problem Domain

**Simulation** Simulation ist der zentrale Punkt der Problem Domain. Sie verwaltet alle wichtigen Elemente; so das Fahrzeug, das Warenhaus, Start- und Zielkoordinaten sowie den zurückgelegten Weg des Fahrzeuges. Die Simulation stellt Methoden zum Simulationsstart, zum Pausieren, Wiederaufnehmen und Beenden der Simulation zur Verfügung. Auch alle Einstellungen zur Simulation können über Methoden der Simulation vorgenommen werden. Als letzte wichtige Funktion liefert die Simulation dem User Interface die Möglichkeit, alle wichtigen Informationen abzufragen. Auch wird das UI benachrichtigt, falls Änderungen stattgefunden haben.

**Way** Way ist eine Liste aus Punkten.

**Warehouse** Das Warehouse ist verantwortlich für das Laden eines Bildes aus einer Datei. Neben dem Laden des Bildes analysiert es dieses und nimmt eine Unterteilung in Hindernisse und Nicht-Hindernisse vor. Alle Bildpunkte, welche nicht als Hindernis taxiert werden, können von dem Fahrzeug befahren werden.





## Hellgelb

- AnnCallback** Das Interface `AnnCallback` muss implementiert werden, wenn dem künstlichen neuronalen Netz der Auftrag zur Berechnung gegeben werden soll. Wenn das KNN fertig gerechnet hat, wird die im Interface definierte Methode aufgerufen.
- Artificial-Neural-Network** Das Interface `ArtificialNeuralNetwork` definiert die Methoden, welche dem Fahrzeug vom KNN zur Verfügung gestellt werden. Die wichtigsten Methoden sind diejenigen zum Hinzufügen neu entdeckter Hindernisse, zur Neuberechnung des Netzes und zur Berechnung des Weges, welcher zum Ziel führt.
- Vehicle** `Vehicle` repräsentiert das Fahrzeug. Neben der `Simulation` bildet das `Vehicle` den zweiten grossen Teil der Problem Domain. Das Fahrzeug kann als der aktive Teil der Simulation angeschaut werden. Es beauftragt das künstliche neuronale Netz, ihm den Weg zu berechnen und manövriert dann mit Hilfe der Fuzzy-Logik dem Weg entlang um die Hindernisse. Die verschiedensten Parameter, wie zum Beispiel die maximale Beschleunigung oder der maximal mögliche Steuerungswinkel sollen ein möglichst reales physikalisches Verhalten bewirken.
- FUZZY** Die Klasse `FUZZY` wird von der Software `fuzzyTECH`<sup>1</sup> generiert und stellt eine Umsetzung der Fuzzy-Regeln in Java dar. Sie verfügt über Methoden zum Setzen der aktuellen Fahrzeugdaten vom `Vehicle` und ermöglicht es, Änderungen bezüglich der Geschwindigkeit und der Steuerung abzufragen. Mehr zur Fuzzy-Logik ist in Kapitel 15.8 und im Anhang B zu finden.

## Dunkelgelb

- DistanceMeasurer** `DistanceMeasurer` repräsentiert einen simplen Distanzmesser. Er führt eine Messung in eine Richtung durch und liefert den gefundenen Hindernis-Punkt zurück. Daneben kann er auch einfach die Euklidische Distanz zum detektierten Hindernis berechnen.
- RotatableDistanceMeasurer** `RotatableDistanceMeasurer` stellt einen Distanzmesser dar, welcher sich in einem bestimmten Aktionswinkel dreht und in einem festgelegten Intervall Messungen durchführt. Die dabei detektierten Hindernis-Punkte werden zurückgegeben.

<sup>1</sup>fuzzyTECH: <http://www.fuzzytech.de/>



**Sensor** Das Interface `Sensor` legt die Methode fest, welche ein Sensor implementiert haben muss. Es ist die Methode zur Durchführung einer Messung; zurückgegeben wird eine Sammlung der detektierten Hindernisse.

**SensorCallback** Das `SensorCallback` muss implementiert werden, wenn man eine asynchrone Anwendung von `RotatableDistanceMeasurer` realisieren möchte. Nach der Durchführung einer Messung wird dann die im Interface festgelegte Methode aufgerufen.

## Rot

**Application** Die `Application` repräsentiert die eigentliche Applikation. Sie wird zur Erzeugung neuer Simulationen verwendet, lädt die Liste der Warenhäuser und die künstlichen neuronalen Netze.

**AuxiliaryFunctions** `AuxiliaryFunctions` ist eine Sammlung von Hilfsfunktionen, welche in der gesamten Problem Domain eingesetzt werden können; so werden Funktionen zur Bestimmung der Richtung bei gegebenem Winkel, zur Normalisierung eines Winkels, welcher negativ oder grösser als  $360^\circ$  ist und zur Detektierung des nächsten Hindernis-Punktes in einer gegebenen Richtung, angeboten.

## Grün

**SimulationChanged** `SimulationChanged` wird verwendet, um den Observern, welche `Simulation` überwachen, mitzuteilen, was sich geändert hat.

**VehicleChanged** `VehicleChanged` wird verwendet, um Observern von `Vehicle` mitzuteilen, welcher Wert sich geändert hat.

## 15.3 User Interface

Das User Interface beinhaltet die in der Abbildung 15.3 dargestellten Klassen. Diese können grob in vier Gruppen unterteilt werden.

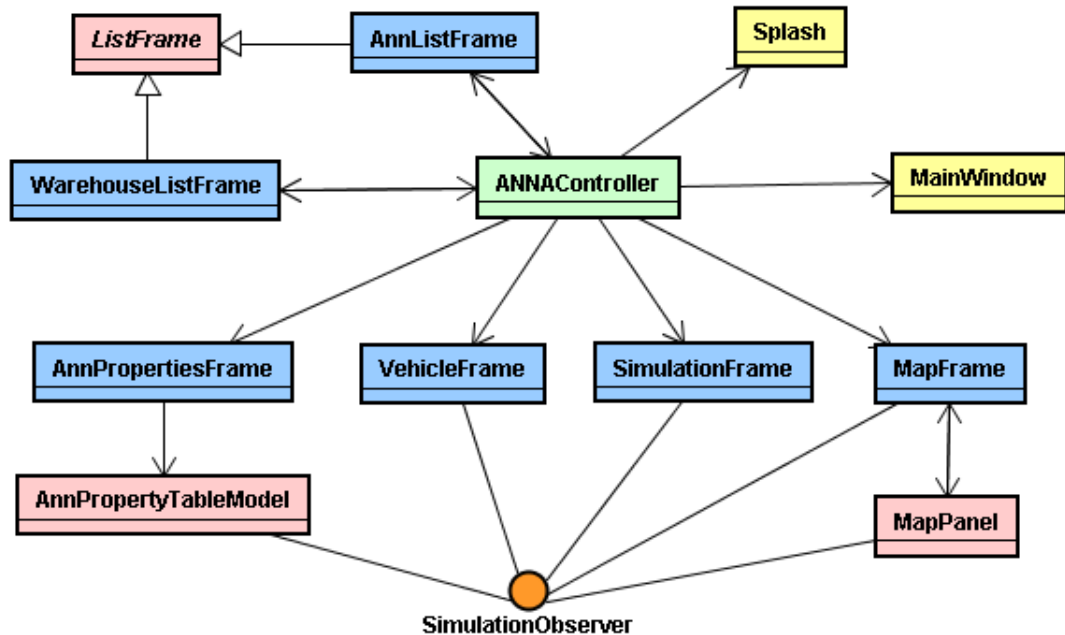


Abbildung 15.3: User Interface

## Grün

**ANNAController** Der ANNAController verwaltet die Kommunikation zwischen der Problem Domain und dem User Interface. Die Application der Problem Domain wird von ihm observiert und somit jedes mal benachrichtigt, falls eine neue Simulation erstellt wird. Die neue Simulation wird dann vom ANNAController an die einzelnen GUI-Komponenten verteilt.

## Gelb

**MainWindow** Das MainWindow ist, wie der Name schon sagt, das Hauptfenster der ganzen Applikation. Es enthält verschiedene interne Fenster, sowie Knöpfe und Menus zur Bedienung.

**Splash** Das Splash Fenster dient zur Angabe der Version und als Startfenster der Applikation.



## Blau

Alle Frames representieren je ein internes Fenster des MainWindow.

**AnnListFrame** Listet alle verfügbaren KNN-Plugins auf. Es kann jeweils nur ein Plugin ausgewählt werden.

**AnnProperties-Frame** Stellt alle Properties des ausgewählten KNN-Plugins dar.

**MapFrame** Diese Frame zeigt die Warenhäuser und Informationen zur laufenden Simulation, wie Geschwindigkeit, Grösse des Lagerhauses, zurückgelegte Distanz und aktuelle Position an.

**SimulationFrame** Hiermit kann eine neue Simulation erstellt und die aktuelle Simulation gestartet, pausiert und gestoppt werden.

**VehicleFrame** In diesem Frame können alle Einstellungen vorgenommen werden, die das Fahrzeug betreffen.

**WarehouseList-Frame** Hier werden alle verfügbaren Lagerhäuser aufgelistet.

## Rot

**AnnProperty-TabelModel** Das AnnPropertyTabelModel wird benötigt um die verschiedenen Properties von KNN-Plugins darzustellen und zu ändern.

**ListFrame** Das ListFrame stellt eine Eltern-Klasse für die einfache Darstellung einer Liste in einem internen Fenster bereit, in welcher nur ein Eintrag markiert werden kann.

**MapPanel** Dieses Panel zeichnet das Warenhaus, das Fahrzeug und allfällige Wege in das MapFrame. Es wird für jede Bewegung neu gezeichnet.

## 15.4 Fahrzeug

### Konfiguration

Das Fahrzeug bietet verschiedene Konfigurations-Möglichkeiten. Es kann die maximale Geschwindigkeit, die Beschleunigung und die Verzögerung eingestellt



werden. Dazu muss auch definiert werden, wie viele Pixel ein Meter beinhaltet. Für die Darstellung kann zusätzlich auch noch die Breite und Länge des Fahrzeuges angegeben werden.

## Sensoren

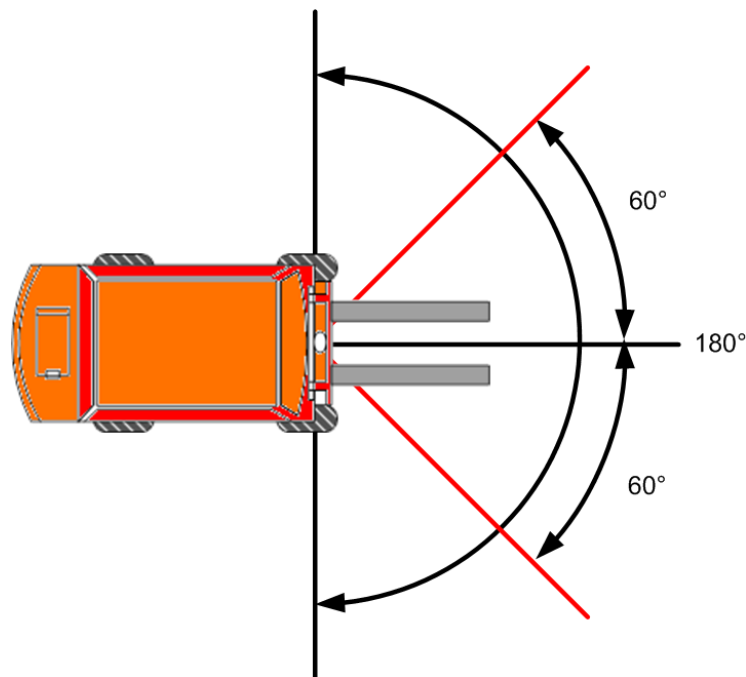


Abbildung 15.4: Fahrzeug von Oben

Zur Positionsbestimmung und für das Ausweichen von Hindernissen ist das Fahrzeug mit drei Distanzsensoren ausgerüstet. Zwei Sensoren sind in Fahrtrichtung, leicht nach aussen, gerichtet. Diese sind in der Abbildung 15.4 als rote Striche eingezeichnet. Der Winkel (hier  $60^\circ$ ) kann beliebig verändert werden. Der dritte Sensor kann sich in alle Richtungen drehen und sucht so die Umgebung nach Hindernissen ab. Auch hier kann der Winkel (hier  $180^\circ$ ) und zusätzlich noch der Abtastintervall definiert werden.

Aus Performancegründen haben die Sensoren nur eine bestimmte Reichweite. Dies bewirkt, dass das Fahrzeug Hindernisse später erkennt als mit einem echten Distanzsensor. Die Reichweite kann jedoch als Lookahead in den Fahrzeuginstellungen verändert werden. Je grösser die Reichweite, desto länger dauert die Berechnung der Distanz.



## Steuerung

Für das Fahrzeug kann ein maximaler Steuerungswinkel angegeben werden. Je schneller das Fahrzeug fährt, desto weniger kann es steuern. Die Geschwindigkeit wird umgerechnet auf Pixel pro Meter. Bei jeder Positionsverschiebung wird geschaut, wie viel Zeit verstrichen ist und dann berechnet, um wie viele Pixel das Fahrzeug verschoben werden muss und in welche Richtung.

Das Fahrzeug erhält vom KNN den optimal berechneten Weg, welcher zum Ziel führt. Der Weg besteht aus verschiedenen Punkten, diesen Punkten folgt das Fahrzeug. Da es nicht immer sinnvoll ist, gerade auf den nächsten Wegpunkt zuzusteuern, kann ein Lookahead bestimmt werden, auf welchen Punkt das Fahrzeug zusteuern soll. Zusätzlich kann noch die Länge des erhaltenen Weges angegeben werden, da es nicht immer nötig ist, den ganzen Weg zu berechnen. Es könnten zum Beispiel noch weitere Hindernisse auftauchen. Für die Steuerung wird eine Fuzzy-Logik verwendet; diese ist im Abschnitt 15.8 genauer erläutert.

## 15.5 Plugins

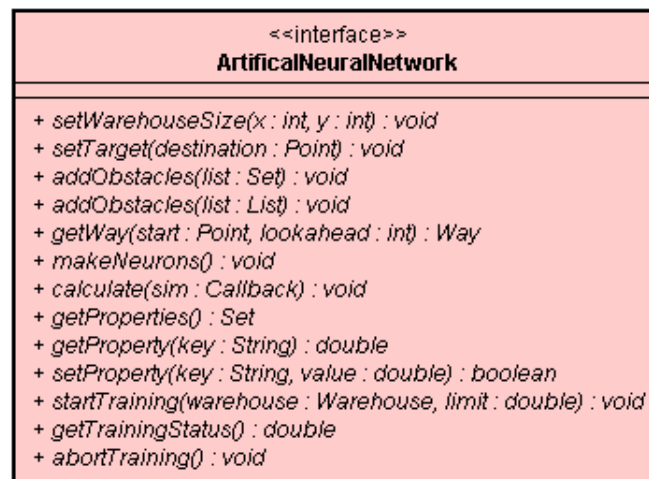


Abbildung 15.5: Interface für Plugins

Alle künstliche neuronale Netze, welche in A.N.N.A als Plugin eingesetzt werden sollen, müssen das Interface `ArtificialNeuralNetwork` implementieren. Dieses Interface definiert alle Methoden, welche von der Applikation aus aufgerufen werden können. Damit die Applikation die Plugins laden kann, müssen sie als Package im Verzeichnis `plugins/bin` gespeichert sein. Die Klasse, welche `ArtificialNeuralNetwork` implementiert, muss denselben Namen haben wie das Package.



Damit jedes Netz seine eigenen Konfigurations-Parameter haben kann, werden diese Parameter alle in einer Map abgelegt, welche dann vom User Interface geholt und in einer Tabelle dargestellt wird. So ist es möglich, mit einer dynamischen Anzahl von Parametern arbeiten zu können.

Die Methode `calculate(Callback)` benötigt als Parameter ein Objekt vom Typ `ch.hsr.anna.pd.ann.Callback`. Das, weil der Methodenaufruf asynchron geschehen soll, so dass die Berechnung des Netzes in einen eigenen Thread ausgelagert werden kann. Nach Beendigung des Berechnens wird für den Callback die Methode `finishCalculating()` aufgerufen, um zu signalisieren, dass die Berechnung abgeschlossen ist.

## 15.6 Hopfield Netz

In diesem Kapitel soll der Aufbau unseres Plugins für das Hopfield-Netz erläutert werden.

### Allgemeines zu Hopfield-Netzen

Ein Hopfield-Netz stellt ein Modell eines künstlichen neuronalen Netzes dar, in welchem alle Neuronen miteinander verbunden sind. Die Verbindungen der Neuronen untereinander sind gewichtet, im normalfall symmetrisch. Das heisst, angenommen man bezeichnet zwei Neuronen mit  $i$  und  $j$ , die Verbindung zwischen  $i$  und  $j$  mit  $w_{ij}$ , diejenige zwischen  $j$  und  $i$  mit  $w_{ji}$ , so sind  $w_{ij}$  und  $w_{ji}$  gleich.

Jedes Neuron erhält als Input den Output aller anderen Neuronen. Es gewichtet alle Inputs, summiert die gewichteten Inputs und hat als Output diese Summe, abzüglich eines eventuellen Schwellwertes  $\theta$ . Der Output wird zusätzlich durch eine Aktivierungsfunktion angepasst. Der Output  $u_i$  eines Neurons  $i$  zur Zeit  $t$  kann also wie folgt beschrieben werden:

$$u_i(t) = \sum_{j=1}^n w_{ij}v_j(t) + \theta_i$$

Als Aktivierungsfunktion kann beispielsweise die Fallunterscheidung verwendet werden, dass der Output 1 ist, falls  $u_i$  grösser als 0 ist, sonst 0.



Die eigentliche Berechnung des Hopfield-Netzes geschieht danach wie folgt: Diejenigen Neuronen, welche beim Input von der Aussenwelt über eine hohe Gewichtung verfügen, dienen als Ausgangspunkt. Ihr hoher Output pflanzt sich in denjenigen Neuronen weiter, welche eine hohe Gewichtung zu ihnen haben. Diese geben den Output ihrerseits weiter. Bei jeder neuen Berechnungsrunde verbreitet sich der Input unter den Neuronen, bis nach einer gewissen Zeit ein stabiler Zustand erreicht wird. Das Berechnen erfolgt entweder synchron, das heisst alle Neuronen berechnen ihren neuen Output zur gleichen Zeit oder asynchron, das heisst ein Neuron nach dem anderen berechnet seinen neuen Wert. Häufig wird dabei die Reihenfolge der Neuronen zufällig bestimmt.

## Unsere Variante

Das von uns implementierte Hopfield-Netz ist relativ simpel gehalten. Die Gewichtungen der einzelnen Neuronen untereinander ist so gewählt, dass jeweils die Gewichtung zu den vier Nachbar-Neuronen rechts, unten, links und oben 1 ist, zu allen anderen Neuronen 0. Dies ermöglicht eine elegante Implementierung und reduziert die Berechnungszeit. Auf die Verwendung eines Schwellwertes wurde verzichtet, da Tests keine besseren Resultate lieferten, wenn ein Schwellwert eingesetzt wurde. Die Berechnung des Outputs eines einzelnen Neurons  $i$  ist also wie folgt:

$$u_i(t) = \sum_{j=1}^n w_{ij} v_j(t)$$

Als Aktivierungsfunktion verwendeten wir einen hyperbolischen Tangens, da wir einerseits diskrete Ausgangswerte benötigten und die Funktion die besten Resultate lieferte. Um die erhaltene Kurve abzuschwächen wurde ein zusätzlicher Faktor nötig. Die verwendete Funktion sieht so aus:

$$out_i = \tanh(0.25u_i)$$

Der optimale Wert des Faktors, 0.25 wurde durch Testen ermittelt.

## Implementierung

Die Implementierung ist, wie in Abbildung 15.6 dargestellt, relativ einfach. Sie besteht aus dem eigentlichen Netz, Hopfield, welches das Interface Artificial-



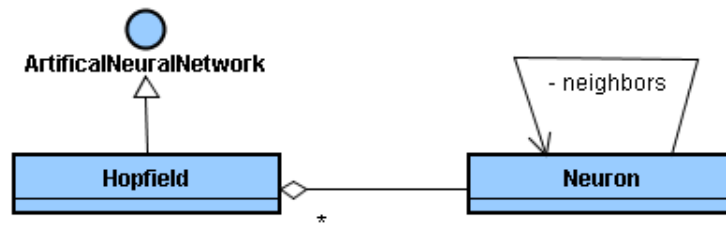


Abbildung 15.6: Hopfield

NeuralNetwork implementiert und einer, der gröse entsprechenden, Anzahl Neuronen. Jedes Neuron hat vier Nachbarn: Neuronen, bei welchen die Verbindung ein Gewicht von 1 hat.

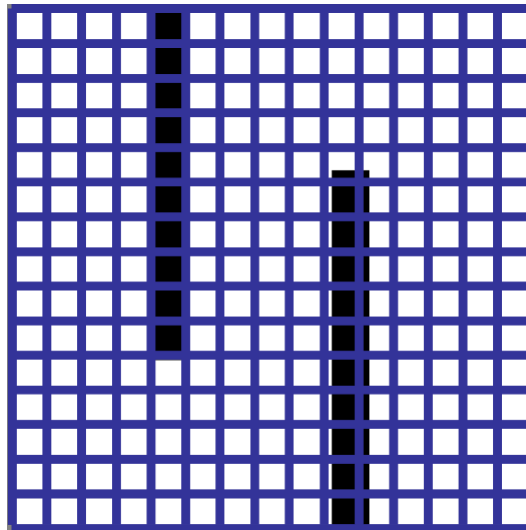


Abbildung 15.7: Hopfield Grundposition

Unsere Absicht ist es, das Netz so über das Bild des Warenhauses zu legen, dass jedes Neuron einen gewissen Bereich, eine gewisse Anzahl von Pixeln, abdeckt (Abbildung 15.7). Danach wird das Neuron, in welchem der Zielpunkt des Fahrzeuges liegt, auf den Wert 1 gesetzt. Die Berechnung des Netzes ergibt dann ein Gebilde, bei welchem der Zielpunkt die Spitze darstellt und je weiter man sich vom Zielpunkt entfernt, desto kleiner werden die Werte (Abbildung 15.8). Werden zusätzlich Hindernisse miteinbezogen (Abbildung 15.9), haben diejenigen Neuronen, welche ein Gebiet mit Hindernissen abdecken, den Wert 0. Nach der Berechnung entsteht ein Gebilde, welches noch immer die Eigenschaft hat, dass das Ziel 1 ist und die Werte der weiter entfernten Neuronen kleiner werden (Abbildung 15.10).

Das Netz wird so lange neu berechnet, bis die Änderungen der Outputs der Neuronen einen gewissen Wert unterschreiten. Falls das zu lange dauert, wird die Berechnung durch einen einstellbaren Timeout unterbrochen. So kann sicherge-

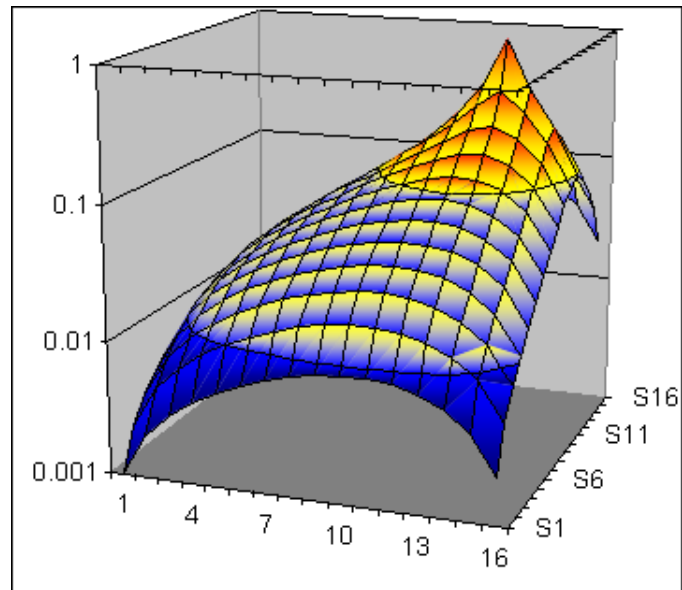


Abbildung 15.8: Hopfield nach Berechnung

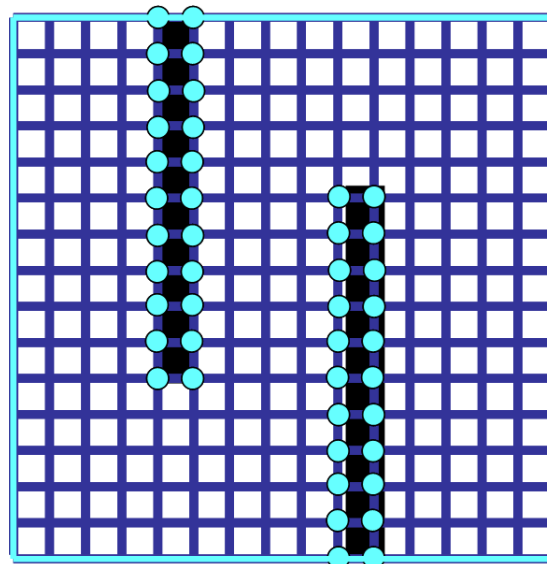


Abbildung 15.9: Hopfield Grundposition mit Hindernissen

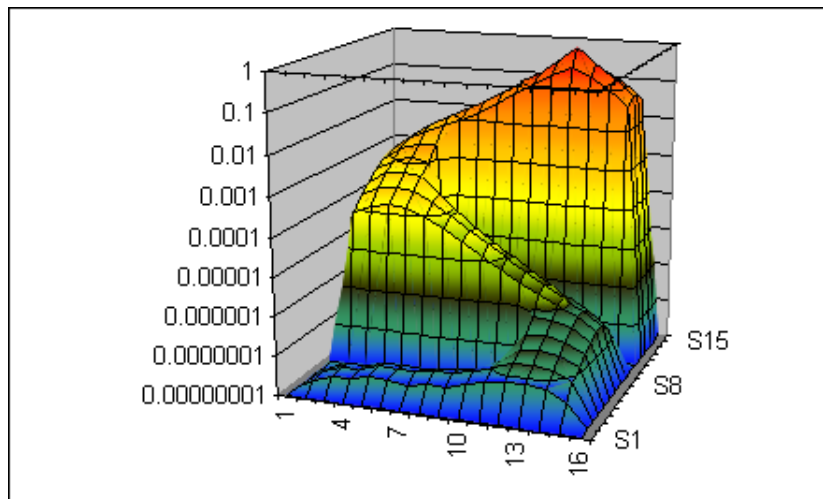


Abbildung 15.10: Hopfield nach Berechnung mit Hindernissen

stellt werden, dass das Fahrzeug innert vernünftiger Zeit einen neuen Weg erhält, auch wenn die Berechnung dann nicht über eine optimale Genauigkeit verfügt. Die Bestimmung des Weges ist danach relativ einfach. Von der aktuellen Position aus folgt man immer dem Neuron mit dem grössten Wert. Nach einer endlichen Anzahl Schritte sollte man das Ziel finden. Das Problem, das jedoch auftreten kann, ist, wenn man auf einen lokalen Maximalwert stösst: Von dort aus führt kein Weg weiter, und das Fahrzeug erreicht das Ziel nicht.

## Weitere Möglichkeiten

Als Variante haben wir ein Netz entwickelt, bei welchem die Gewichtung nicht nur bei vier Nachbar-Neuronen 1 ist, sondern bei allen acht. Diese Implementierung lieferte jedoch keine besseren Resultate; aufgrund der viel grösseren Berechnungszeit war sie für unsere Zwecke unbrauchbar.

Daneben gäbe es natürlich noch weitere Varianten, wie man ein Hopfield-Netz implementieren könnte. Neben diversen Anpassungen an Gewichten, Schwellenwert und der Aktivierungsfunktion wäre vor allem eine Lösung für das Problem der lokalen Minima interessant.

## 15.7 Threads

Im wesentlichen laufen während einer Simulation drei Threads parallel ab. Wie in der Abbildung 15.11 dargestellt, sind dies die Berechnung des KNN, das Fahrzeug und der rotierende Distanzsensor.

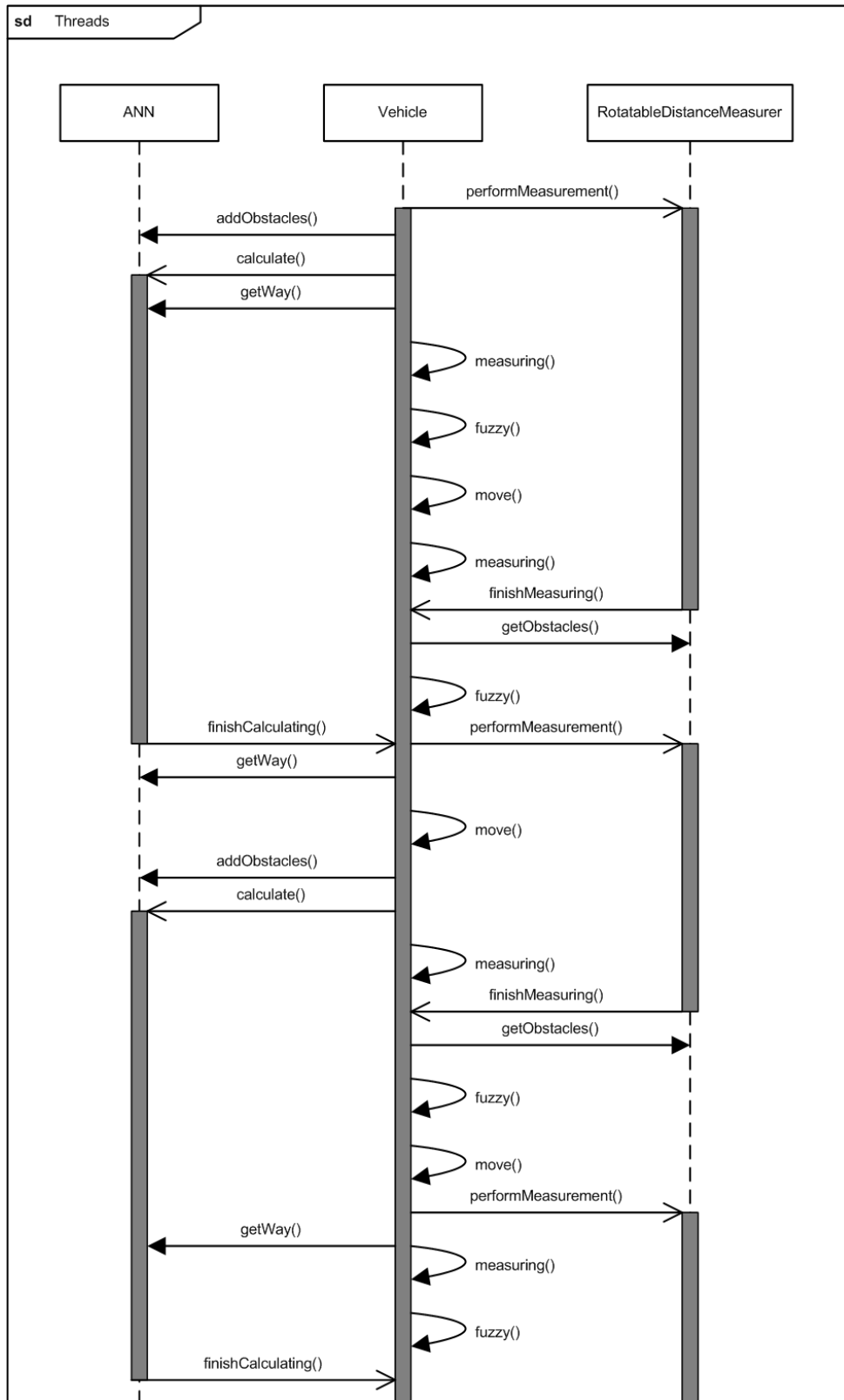


Abbildung 15.11: Threads



## KNN-Thread

Der KNN-Thread hat nur eine Aufgabe: Das Hopfield-Netz den immer wieder neu gefundenen Hindernissen anzupassen. Um dies zu berechnen, erhält das KNN vom Fahrzeug zuerst die gefundenen Hindernisse und anschliessend den Befehl die Berechnung zu starten. Wird die Methode `calculate()` aufgerufen, startet der Thread. Wenn alles korrekt abläuft und das Netz den neuen Bedingungen angepasst ist, wird dies dem Fahrzeug per Callback (`finishCalculating()`) gemeldet und der Thread beendet sich. Falls die Berechnung zu dauert, gibt es einen Timeout und die Berechnung wird abgebrochen.

## Fahrzeug-Thread

Dies ist der Hauptthread der Simulation. Der Ablauf ist immer derselbe. Anfangs werden die separaten KNN- und Distanzsensor-Threads gestartet. Dann startet der immer wiederkehrende Ablauf:

1. Der linke und rechte Distanzsensor misst die Entfernung.
2. Der Fuzzy-Logik werden die Eingangsvariablen geliefert und der anschließende Ausgang wird verarbeitet.
3. Das Fahrzeug bewegt sich in die berechnete Richtung.

Um das Diagramm 15.11 einfach zu gestalten, wurden kleinere Abläufe weggelassen und nur rechenlastige Methoden aufgeführt.

Zwischendurch wird ein Callback des KNN- oder Distanzsensor-Threads aufgerufen. Sobald dies geschieht, wird der neue Weg bzw. die neuen Hindernisse abgeholt und weitergerechnet.

## Distanzsensor-Thread

Die Aufgabe dieser Threads ist es, Hindernisse in der Umgebung des Fahrzeuges zu finden. Bei jedem Aufruf von `performMeasurement()` startet ein Thread. Dabei vollführt der rotierende Sensor eine volle Drehung, wie im Abschnitt 15.4 genauer erläutert ist. Die gefundenen Hindernisse werden zwischengespeichert bis das Fahrzeug diese abholt.

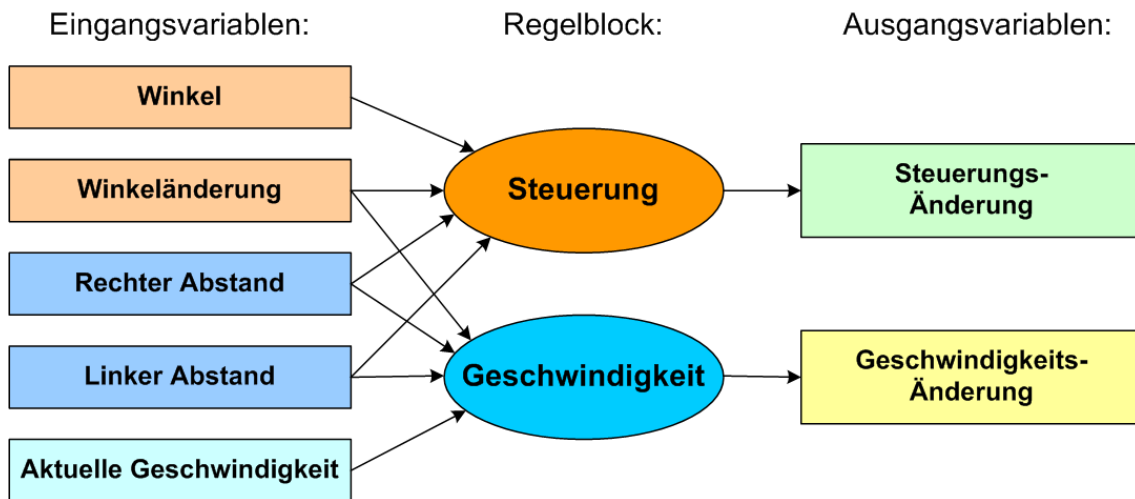


Abbildung 15.12: Fuzzy-Logik

## 15.8 Fuzzy-Logik

Das Fuzzy-System besteht aus fünf Eingangs-, zwei Ausgangsvariablen und zwei Regelblöcken. In der Abbildung 15.12 ist die Systemstruktur aufgezeichnet. Die genauen Werte und Regeln der Fuzzy-Sets sind im Anhang B auf Seite 123 zu finden.

### Eingangsvariablen

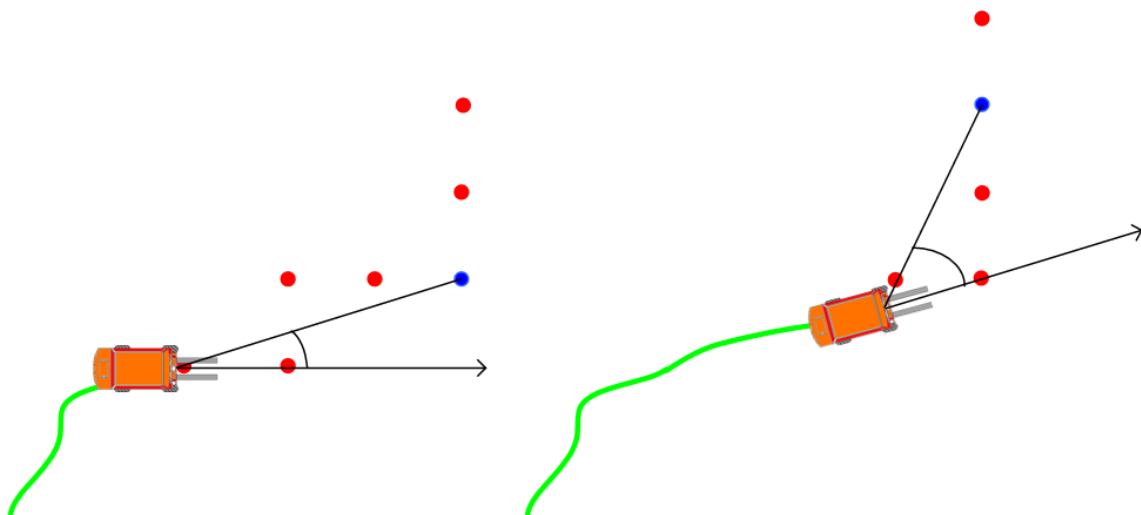


Abbildung 15.13: Lookahead

**Winkel** Hier wird der Winkel in Grad zwischen der Fahrtrichtung des



Fahrzeuges und dem angesteuerten Punkt des optimalen Weges angegeben, wie es in der Abbildung 15.13 dargestellt ist.

|                                     |  |
|-------------------------------------|--|
| <b>Winkeländerung</b>               | Dies ist die Änderung zwischen dem letzten und dem jetzigen Winkel. (siehe ebenfalls Abbildung 15.13)    |
| <b>Rechter &amp; Linker Abstand</b> | Das ist jeweils der vom rechten bzw. linken Distanzsensor gemessene Wert.                                |
| <b>Aktuelle Geschwindigkeit</b>     | Dieser Input wird mit der momentanen Geschwindigkeit in Prozent zur maximalen Geschwindigkeit angegeben. |

## Regelblöcke

Die beiden Regelblöcke, Steuerung und Geschwindigkeit, werten anhand von vordefinierten Regeln die Eingangsvariablen aus und liefern dies den Ausgangsvariablen. Die Regeln sind im Anhang B.3 zu finden.

## Ausgangsvariablen

|                                 |   |
|---------------------------------|---|
| <b>Steuerungsänderung</b>       | Dieser Ausgang gibt die Änderung des Steuerungswinkels des Fahrzeuges zurück.   |
| <b>Geschwindigkeitsänderung</b> | Hiermit wird bestimmt, wie viel das Fahrzeug abbremsen bzw. beschleunigen muss. |

## 15.9 Koordinatensystem

Das Fahrzeug und die Sensoren arbeiten nach dem kartesischen Koordinatensystem, dieses ist in der Abbildung 15.14 aufgezeichnet. Der Nullpunkt eines Warenhauses ist immer oben links. Die X-Achse vergrößert sich von links nach rechts und die Y-Achse von oben nach unten. Bei Winkel können sowohl positive als auch negative Werte auftreten.

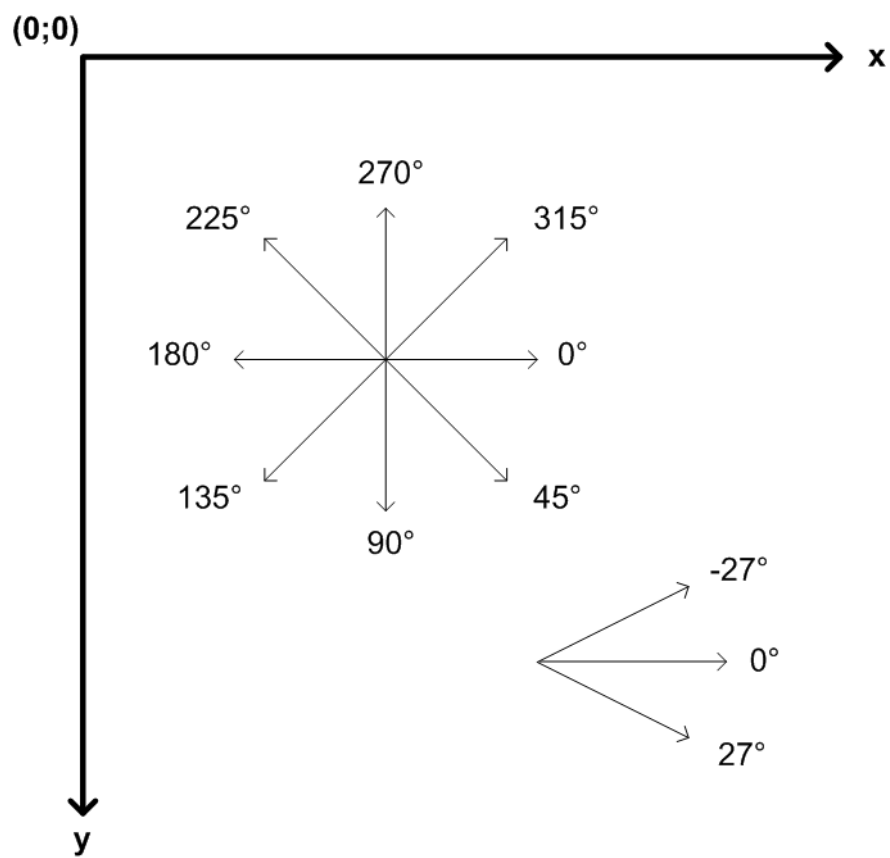


Abbildung 15.14: Koordinatensystem





## 16 Entscheide

In diesem Kapitel werden unsere Entscheidungen während der Projektdauer aufgeführt.

### 16.1 Design-Entscheide

#### Use Cases

Einige Use Cases wurden nicht umgesetzt. Der Grund dafür waren zum einen, dass sie nach unseren Überlegungen nicht mehr nötig waren, aber auch aus zeitlichen Gründen. Die unten aufgeführten Use Cases sind im Kapitel 10 genau beschrieben.

|                                 |   |
|---------------------------------|---|
| <b>Programm starten/beenden</b> | Der alternative Ablauf wurde nicht umgesetzt, die Konfiguration des neuronalen Netzes kann nicht gespeichert werden.  |
| <b>KNN auswählen</b>            | Die Konfiguration eines neuronalen Netzes kann nicht geladen werden, da dies nicht implementiert wurde. Ansonsten wurde der UC umgesetzt.   |
| <b>KNN-Parameter einstellen</b> | Die Speicherung der Parameter eines KNN wurde ebenfalls nicht umgesetzt, der Rest des Use Cases ist aber vorhanden.   |
| <b>Lager auswählen</b>          | Warenhäuser können nicht während der Laufzeit geladen werden. Sie müssen sich bei Programmstart im vorgegeben Ordner befinden und werden automatisch geladen. Danach können sie, wie im UC beschrieben, ausgewählt werden.  |
| <b>KNN trainieren</b>           | In der Studienarbeit wurde nur das Hopfield-Netz umgesetzt, welches kein Training benötigt. Deshalb wurde dann keine Lernkurve implementiert, da diese nicht ausgiebig hätte getestet werden können. Es ist aber möglich, ein Netz zu trainieren und allenfalls den Test abzuberechnen. |



|                               |  |
|-------------------------------|--|
| <b>Programm konfigurieren</b> | Alle Einstellungen können in den vorhandenen Fenstern gemacht werden.  |
| <b>KNN Details anzeigen</b>   | Details eines KNN können nicht angezeigt werden. Ein Plugin-System mit dynamischer Anzeige der jeweiligen Details wäre sehr zeitaufwändig gewesen und wurde deshalb nicht umgesetzt. Zusätzlich würde es die Programmierung der Plugins erschweren.          |
| <b>KNN speichern</b>          | Ein künstliche neuronales Netz, egal ob trainiert oder nicht, kann nicht abgespeichert werden. Da die Werte der einzelnen Neuronen während der Simulation immer wieder dynamisch neu berechnet werden, machte es keinen Sinn ein Netz abspeichern zu können. |
| <b>Restliche UC's</b>         | Alle nicht aufgeführten Use Cases wurden so umgesetzt, wie in Kapitel 10.  |

## KNN

**Hopfield** In der Einarbeitungs-Zeit fällten wir den Entscheid ein Hopfield-Netz zu implementieren. Mit einem Test-Programm kontrollierten wird zuerst, ob die Umsetzung möglich ist. Mit dem erfolgreichen Test fühlten wir uns sicher, das Hopfield-Netz umzusetzen zu können.

Während der Einarbeitung ins Theman KNN kam noch ein weiteres Netz in Frage, das Kohonen-Netz. Für die Umsetzung des autonomen Fahrzeuges mit unseren Bedingungen, fanden wir jedoch, das Hopfield-Netz sei die bessere Lösung.

**Wegpunkte** Der gefundene Weg des Hopfield-Netzes besteht aus einer Menge von Punkten. Der Abstand zwischen den einzelnen Punkten variiert und ist von der Dichte/Grösse des Netzes abhängig.

Eine Alternative wäre gewesen, dass das Netz einen vollständigen Weg zurück gibt, das heisst, einen Weg der alle Punkte beinhaltet auf dem das Fahrzeug optimal zum Ziel kommt. Nach unserer Überlegung wäre ein durchgezogener Weg viel rechenaufwändiger als ein Weg der nur aus wenigen Punkten besteht. Für die Steuerung wird es keinen grossen Einfluss haben, sofern ein guter Lookahead-Wert des Fahrzeuges gewählt wurde. Dies ist normalerweise ein Wert zwischen 2 und 4.



## 16.2 Software-Entscheide

**Java** Als Programmiersprache wurde Java gewählt. Dies, weil wir in unserem Studium schon viel mit Java gearbeitet haben und deshalb schon einige Erfahrung sammeln konnten.

Zur Auswahl standen nach unserer Ansicht noch C++ und C#. Da wir mit diesen Sprachen sehr wenig Erfahrung haben, besonders mit graphischen Darstellungen, fiel unsere Entscheidung schnell auf Java.

**fuzzyTECH** Für die Erstellung der Fuzzy-Logik war schnell klar, dass diese mit dem Programm fuzzyTECH erstellt werden würde. Neben einem neuronalen Netz noch ein Fuzzy-System zu implementieren, hätte noch mehr Zeit in Anspruch genommen. Zusätzlich ist der generierte Java-Code von fuzzyTECH einfach in eine Applikation einzubinden.

**L<sup>A</sup>T<sub>E</sub>X** Im Verlaufe unserer Informatik-Ausbildung haben wir schon viele Arbeiten mit dem Microsoft Word geschrieben. Mit L<sup>A</sup>T<sub>E</sub>X haben wir eine Alternativ gewählt, welche uns weniger Nerven kostete und erst noch einen Lerninhalt hatte. Zwar hatten wir schon die letzte Studienarbeit mit L<sup>A</sup>T<sub>E</sub>X geschrieben, aber man lernt immer wieder etwas Neues dabei.

L<sup>A</sup>T<sub>E</sub>X bieten schon von begin weg eine schönere Darstellung der Dokumente. Zusätzlich hat es den Vorteil, dass alle L<sup>A</sup>T<sub>E</sub>X-Dateien normale Text-Dateien sind und somit gleichzeitig am selben Dokument gearbeitet werden kann und die Versionskontrolle die verschiedenen Versionen selbständig zusammenfügt.



# V

## Resultate & Erkenntnisse

Verantwortlich: Rico Steffen

| <b>Datum</b> | <b>Version</b> | <b>Änderung</b> |
|--------------|----------------|-----------------|
| 01.07.2006   | 0.1            | Erstellt        |
| 03.07.2006   | 0.2            | Fuzzy           |
| 04.07.2006   | 0.3            | Entscheide      |
| 06.07.2006   | 1.0            | Release         |

In diesem Dokumentteil werden die Resultate und Erkenntnisse der Studienarbeit aufgeführt. Eine genauere Auflistung der Änderungen ist im Anhang A.5 zu finden.





## 17 Fuzzy-Logik

In diesem Abschnitt werden die Resultate und Erkenntnisse der Fuzzy-Logik beschrieben.

### 17.1 Aufbau

Die Fuzzy-Logik wurde im Verlauf der Studienarbeit mehrmals umgebaut und verändert. Wie der schlussendliche Aufbau aussieht ist im Abschnitt 15.8 beschrieben. Anfangs jedoch hatte das Fahrzeug nur den rotierenden Sensor, welcher nicht in die Fuzzy-Logik einfluss. Dies hatte zur Folge, dass das Fahrzeug nur nach dem vom neuronalen Netz gelieferten Weg lenkte. Dies wiederum bewirkte, dass das Fahrzeug teilweise Kurven sehr eng fuhr und darum in die Hindernisse hinein steuerte.

So kamen die seitlichen Sensoren hinzu, welche dem Fahrzeug ermöglichten zu erkennen, ob eine Kurve gefahrenlos gemacht werden kann. Dies führte jedoch auch zum Problem des Übersteuerns, welches im Abschnitt 17.2 erläutert ist.

### 17.2 Übersteuern

Das Fahrzeug fährt, wie in der Abbildung 17.1 zu sehen ist, eine Schlangenlinie. Dieses Problem des Übersteuerns ergibt sich aus der Fuzzy-Logik und ist somit natürlich. Das Fahrzeug schaut immer ein Stück voraus und zielt auf einen Punkt auf dem optimalen Weg. Durch die Unschärfe der Fuzzy-Logik, welche gewollt ist, ergibt sich daraus die Schlangenlinie.

Der Übersteuerung könnte ein wenig entgegen gewirkt werden, wenn man verschiedene Einstellungen der Fuzzy-Sets ausprobieren würde. Dabei besteht aber die Gefahr, dass das Fahrzeug zu wenig scharfe Kurven fährt und so in ein Hindernis hinein fahren würde.

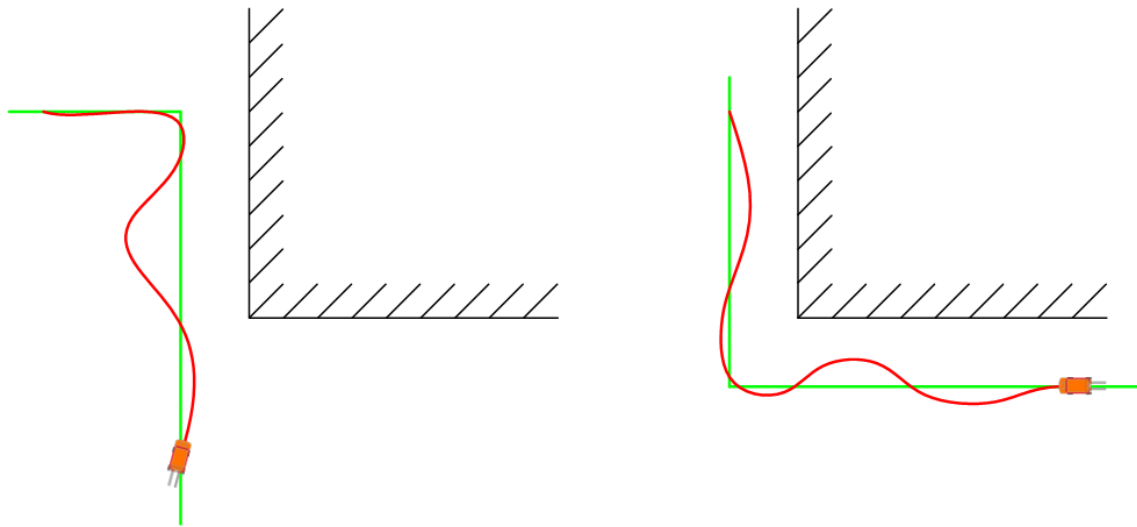


Abbildung 17.1: Fuzzy-Fahrzeugsteuerung

## 17.3 Konfiguration

Die beiden Fuzzy-Sets, welche die Steuerung der Lenkung und der Geschwindigkeit übernehmen, haben sehr viele Eingangsvariablen. Diese Variablen haben wiederum drei bis fünf linguistische Variablen. Daraus ergeben sich sehr viele Regeln für den Ausgang (siehe Anhang B). Um ein Optimum aus der Fuzzy-Logik heraus zu holen, müssten viele verschiedene Konfigurationen durchprobiert werden. Da es sehr zeitaufwändig ist, immer mit dem fuzzyTECH-Tool die Variablen zu ändern und den Code zu generieren, konnten keine optimalen Einstellungen gefunden werden.





## 18 Hopfield

In diesem Kapitel werden die Resultate und Erkenntnisse über das Hopfield-Netz, welche während der Projektzeit gemacht wurden, beschrieben.

### 18.1 Double

Schon in den Testphasen des Hopfield-Netzes wurde bald ersichtlich, dass die Genauigkeit des Doubles ein Problem darstellen könnte oder zumindest die Präzision der Berechnungen. Besonders bei sehr grossen Netzen wurden die Werte der Neuronen, je weiter sie vom Zielneuron entfernt waren, extrem klein.

### 18.2 Lokaler Maximumwert

Bei labyrinth-artigen Warenhäusern kommt es zum Problem von lokalen Minimum- bzw. bei uns Maximum-Werten. Dieses Problem wird noch verstärkt, weil die Berechnung des Hopfield-Netzes sehr lange dauert und so der Timeout für die Berechnung abläuft. So konvergiert das Netz nicht und es entstehen die lokalen Maximumwerte.

Gegen dieses Problem kann nicht viel gemacht werden. Ist der Timeout zu klein, kommen viele solche Maximumwerte vor, dafür geht die Berechnung schnell. Ist jedoch der Timeout zu gross, rechnet das Netzwerk zu lange und das Fahrzeug steuert während dieser Zeit vielleicht in ein Hindernis.

Um dem entgegen zu wirken haben wir in die Wegsuche des Netzes eingegriffen. Während das Netz den Weg sucht, von einem Neuron zum nächst höheren Neuron, kann das Letztere nicht wieder dem nächsten Neuron in der Kette folgen. Dies hilft jedoch nur begrenzt und bei grösseren lokalen Maximumwerten nützt auch dies nichts.



## 18.3 Performance

Die Berechnung des Hopfield-Netzes ist sehr rechenintensiv. Nur schon bei kleinen Netzen geht die Berechnung bis das Netz konvergiert relativ lange. Da wäre wohl noch ein wenig an Geschwindigkeit heraus zu holen.

Unsere Versuche beschränkten sich grösstenteils nur auf das «normale» Hopfield-Netz, bei welchem ein Neuron nur vier Nachbarn besitzt. Beim «diagonalen» Hopfield (mit acht Nachbarn), stieg der Rechenaufwand so extrem an, dass das Netz viel zu früh den Timeout erreichte oder das Fahrzeug in ein Hindernis krachte, da es noch keinen optimalen Weg zur Verfügung hatte.

Ebenfalls gehen Double-Berechnungen um einiges länger als solche mit Integer- bzw. Long-Werten. So könnte vielleicht mit geschickt platzierten Typumwandlungen noch mehr Geschwindigkeit heraus geholt werden.

## 18.4 Netzdichte

Wie schon erwähnt steigt der Rechenaufwand des Netzes extrem stark, je grösser das Netz gewählt wird. Es ist zusätzlich auch nicht immer gegeben, dass in Bezug auf die Wegsuche ein besseres Resultat erzielt wird. Je enger die Neuronen beieinander liegen, desto näher führt der gefundene Weg an Hindernisse heran. Dadurch kann zu Problemen mit der Steuerung kommen, da das Fahrzeug von Hindernissen wegsteuert. Jedoch kann mit einem dichteren Netz druch engere Durchgänge gefahren werden. Bei kleiner Netzdichte werden diese teilweise übersehen.



## 19 A.N.N.A

In diesem Kapitel werden noch allgemeine Resultate und Erkenntnisse zu A.N.N.A aufgeführt.

### 19.1 Performance

Wer A.N.N.A testet merkt sofort, dass die Performance sehr schlecht ist. Das Fahrzeug fährt langsam und springt. Auch sind die Reaktionszeiten des User Interfaces nicht optimal. Hier wäre bestimmt noch etwas an Performance raus zu holen.

Messungen haben gezeigt, dass das Skalieren des gepufferten Bildes, welches das Warehouse, das Fahrzeug und alle Wege enthält, extrem rechenaufwändig ist. Bei Messungen lag die Zeit zwischen 100 und 200 Millisekunden. Da dies jedoch praktisch bei jedem repaint des Bildes notwendig ist, war es schwer zu optimieren.

Zudem laufen der KNN- und Distanzsensor-Thread (siehe Kapitel 15.7) parallel. Diese beiden Threads lasten den Prozessor praktisch vollkommen aus, dadurch hat es ein Repainter-Thread schwer performant das User Interface zu zeichnen. Das heraufsetzen der Thread-Priorität des Repainters könnte dazu führen, dass Hindernisse zu spät erkannt werden. Deshalb wurde auf diese Massnahme verzichtet.

### 19.2 Implementierung

Die Implementierung des Hopfield-Netzes nahm nur sehr wenig Zeit in Anspruch, da durch die Einarbeitungsphase schon relativ klar war, was genau berechnet werden muss. So war die Wegsuche schnell implementiert.

Das Planen und Umsetzen des User Interfaces und der Problem Domain brauchte dann jedoch sehr viel Zeit. Nicht zu vergessen waren dabei die Winkel-Berech-



nungen. Diese kamen an vielen Orten in der Problem Domain vor. Diese Berechnungen waren auch mit Unit-Test schwer zu kontrollieren, da die ständigen Rad <-> Grad Umrechnungen die Genauigkeit beeinträchtigen und dadurch teilweise von Hand gerechnet werden mussten. Dies breitete uns anfangs viel Mühe und es schlichen sich viele Fehler ein.

### 19.3 Lookahead des Fahrzeuges

Einen optimalen Lookahead zu bestimmen, das heisst auf welchen Punkt des berechneten Weges das Fahrzeug zu steuern soll, ist durch Testen zu finden. Der Wert hängt stark von der Grösse des neuronalen Netzes, der Geschwindigkeit des Fahrzeuges und der Fuzzy-Logik ab. Schaut das Fahrzeug zu weit nach vorn, kommt es den Hindernissen bei Kurven viel näher und fährt teilweise in sie hinein. Schaut das Fahrzeug aber zu wenig weit nach vorn, kann es vorkommen, dass es umkehrt und auf den näheren Punkt zufährt.

# VI

## Qualitätssicherung

Verantwortlich: Roland Fehlmann

| <b>Datum</b> | <b>Version</b> | <b>Änderung</b> |
|--------------|----------------|-----------------|
| 01.07.2006   | 0.1            | Erstellt        |
| 02.07.2006   | 0.1            | Management      |
| 06.07.2006   | 1.0            | Release         |

Hier werden die getroffenen Massnahmen, um die Qualität der Applikation sicher zu stellen, beschrieben. Eine genauere Auflistung der Änderungen ist im Anhang A.6 zu finden.





# 20 Methoden zur Qualitätssicherung

## 20.1 Protokolle

Von jedem Meeting mit dem Betreuer wird ein Protokoll erstellt. Hiermit können alle besprochenen Punkte, wie Verbesserungen, Beschlüsse oder Diskutiertes, jederzeit nachgeschlagen werden. All Protokolle befinden sich im Anhang C.

## 20.2 Pair Programming

Die meisten Programmier-Aufgaben werden unter den Teammitgliedern aufgeteilt, damit ein gutes Vorankommen garantiert ist. Bei besonders schweren, komplizierten oder fehleranfälligen Methoden und Programmteilen wird zusammen gearbeitet. So können Fehler vermieden werden und die Qualität des Codes wird verbessert.

Bei unvorhergesehenen Schwierigkeiten, zum Beispiel bei heiklen Codestellen, wird das Teammitglied kurz hinzu gezogen und man schaut sich den Code zusammen an.

## 20.3 Coderichtlinien

Am Projektanfang wurde ein Code Template für Eclipse erstellt. Damit wird sichergestellt, dass beide Teammitglieder sauber formatierten Code schreiben. Zudem kann damit der Code durch Eclipse automatisch formatiert werden.



## 20.4 Eclipse

Das Eclipse ist eine gute Hilfe damit kein Code vergessen geht. Es bietet die Möglichkeit Warnhinweise für unschönen Code, vergessene Felder und Methoden oder Anderes, einzuschalten. Dies ermöglicht es relativ einfach den Code zu überprüfen und auch undokumentierte Stellen zu finden.





# 21 Getestete Ebenen

## 21.1 Unit Tests

Während der Entwicklung werden teilweise Unit-Tests geschrieben. Für viele Methoden ist der Aufwand gegenüber dem Nutzen meist jedoch zu gross. Das Ziel ist es, besonders kritische Methoden, wie z.B. Winkelberechnungen, mit Unit-Test genau zu prüfen.

## 21.2 Ad-hoc Tests

Ad-hoc Test sind informelle Tests welche während der Entwicklung durch den Entwickler selbst durchgeführt werden. Solche Tests werden während der Projektdauer ständig durchgeführt um das Programmierte sofort zu testen und zu kontrollieren.





## 22 Management

### 22.1 Software-Verwaltung

Für die Sourcecode- und Dokumentenverwaltung wird Subversion auf einem eigenen Server eingesetzt. Dadurch ist sichergestellt, dass jede frühere Codeversion jederzeit wiederhergestellt werden kann.

### 22.2 ToDo's & Bug Tracking

Im Wiki wird eine Seite gepflegt auf welcher alle Arbeiten, die noch zu erledigen sind und allfällige Fehler erfasst werden. Diese Liste wird laufend aktualisiert und ergänzt. Diese Erfassung ist zwar sehr rudimentär, bringt aber auch keinen Overhead und ist dadurch sehr effizient.

### 22.3 Änderungsmanagement

Anstehende Änderungen werden im Team besprochen und dann ins Wiki auf die ToDo Liste gesetzt. Ist die Änderung durchgeführt, wird der Eintrag unter «Erledigt» einsortiert.

### 22.4 Arbeitszeit

Die Arbeitszeiten werden laufend erfasst und jede Woche in den Detailzeitplan (siehe Seite 7) übertragen. Für das Projekt sind keine genauen Arbeitszeiten festgelegt, es sollten jedoch etwas 14 Stunden pro Woche gearbeitet werden. Wann und wo jedes Teammitglied arbeitet ist ihm freigestellt. Jedoch wird der Stand der Arbeit laufend besprochen und einander mitgeteilt. Damit das Projekt nicht in



Rückstand gerät, wird immer wieder der grobe Zeitplan konsultiert und mit dem aktuellen Stand verglichen.

# VII

## Persönliche Erfahrungen

Verantwortlich: Roland Fehlmann & Rico Steffen

| <b>Datum</b> | <b>Version</b> | <b>Änderung</b> |
|--------------|----------------|-----------------|
| 01.07.2006   | 0.1            | Erstellt        |
| 04.07.2006   | 0.2            | Rico Steffen    |
| 05.07.2006   | 0.3            | Roland Fehlmann |
| 06.07.2006   | 1.0            | Release         |

Die persönlichen Berichte zeigen unsere Erfahrungen und Erkenntnisse während der Studienarbeit. Eine genauere Auflistung der Änderungen ist im Anhang A.7 zu finden.





## 23 Roland Fehlmann

Seit ich mich intensiver mit der Informatik auseinandersetze, hat mich das Gebiet der künstlichen Intelligenz fasziniert. Da wir jetzt, im sechsten Semester, endlich etwas über KI in der Schule lernen (im Rahmen des Moduls Wissensbasierte Systeme), war der Zeitpunkt ideal, auch die Studienarbeit in diesem Gebiet zu schreiben – um mich auch wirklich in das Thema vertiefen zu können. Als dann die Arbeit Autonomes Fahrzeug ausgeschrieben war, hatte ich kein Problem mehr, mich zu entscheiden, welche Studienarbeit ich durchführen wollte. Hinzu kam das Glück, sie dann auch wirklich durchführen zu können.

Die Einarbeitungszeit zu Beginn der Arbeit war sehr lehrreich. Da ich mich bereits für meine letzte Studienarbeit in ein Gebiet einarbeiten musste (Bildsegmentierung), wusste ich, was mich erwartet. Der schwierigste Punkt beim Einarbeiten war es, das ganze Gebiet überschauen zu müssen, um festlegen zu können, welche Techniken in der Arbeit zur Anwendung kommen sollten. Hinzu kam, dass es manchmal schwierig war, sich die konkrete Anwendung der, in den Büchern oft sehr abstrakt beschriebenen, Methoden vorzustellen. Auch wenn ein Beispiel beschrieben war, konnten die Erkenntnisse nicht immer einfach auf unsere Problemstellung übertragen werden. Trotzdem war die Zeit aufschlussreich und ich denke, wir haben schlussendlich mit dem Hopfield-Netz und der Fuzzy-Logik keine schlechte Wahl getroffen.

Der erste Wendepunkt unserer Arbeit war, als wir eine erste Hopfield-Implementation erstellen konnten. Von diesem Moment an wussten wir wirklich, dass und wie unser Problem zu lösen war. In der Folge kamen wir sehr gut voran, die Problem-Domain, das GUI, die Steuerung und auch das Hopfield-Plugin für unsere Anwendung konnten zügig erstellt werden. Schwieriger wurde es, das Ganze anschliessend zu optimieren: Die Steuerung zu verbessern, die Winkelberechnungen überprüfen, zusätzliche kleine Features hinzufügen – all das benötigte sehr viel Zeit, welche wegen der Einarbeitungsphase nicht wirklich vorhanden war.

Dank der guten Zusammenarbeit im Team und der hohen Motivation, das Fahrzeug wirklich zum Fahren zu bringen, konnte am Schluss doch eine gut funktionierende Applikation erstellt werden – mit dem erreichten Resultat bin ich sehr zufrieden. Auch in Zukunft werde ich mich mit künstlicher Intelligenz auseinandersetzen und versuchen, mich weiter zu vertiefen. Schön wäre es, auch die Diplomarbeit in diesem Themengebiet schreiben zu können.







## 24 Rico Steffen

Ich war motiviert in das Semester gestartet. Freute mich auf die Studienarbeit über das Thema der künstlichen neuronalen Netze. Auch jetzt am Ende ist meine Stimmung, in Bezug auf die Arbeit, noch immer sehr gut.

Anfangs hiess es für uns, Einarbeiten in das neue Thema. Da ich mich noch nicht sehr stark mit künstlichen Intelligenzen auseinandergesetzt hatte, waren die neuronalen Netze für mich völlig neu. Für diese Projektphase wurde sehr viel Zeit eingeplant. Im nachhinein muss ich jedoch sagen, dass diese Zeit vielleicht etwas hätte verkürzt werden können. Das viele Lesen über die Netze, welche in den Büchern meist etwa gleich erklärt waren, drückte dann doch ein wenig auf die Arbeits- bzw. Lesemoral.

Nachdem wir uns auf das Hopfield-Netz festgelegt hatten, ging es dann endlich an die Entwicklung von A.N.N.A. Wir erledigten dies nach dem gelernten Software-Entwicklungs-Prozess. Vielleicht hätte eine agilere Methode besser gepasst, da es teilweise doch sehr viel zu schreiben und definieren gab.

Als wir dann endlich ans Programmieren kamen, ging es doch sehr flott voran. Leider gab es wie immer einige unvorhergesehene Probleme. So kosteten mich die Winkelberechnungen einiges an Nerven. Ein Weg wie wir diese Berechnungen einfacher hätten testen können, fiel mir jedoch nicht ein. So schrieben wir gute Unit-Test und bekamen damit auch noch die letzten fehlerhaften Winkel heraus. Ein Punkt der mich stört ist, dass wir die Performance nicht verbessern konnten. Dazu hatten wir zu wenig Zeit und Java war dafür wohl auch die falsche Programmiersprache. Das Endergebnis von A.N.N.A gefällt mir dann doch sehr.

Während der gesamten Projektdauer konnten Roland und ich sehr gut zusammenarbeiten. Wir ergänzen uns sehr gut und bei den wenigen Unstimmigkeiten war schnell eine Lösung gefunden und wir konnten weiterarbeiten. Wir liessen uns auch durch den immer näher kommenden Abgabetermin nicht aus der Ruhe bringen. Dazu beigetragen hat bestimmt auch unsere gute Planung, bei der wir immer gut sahen, wie viel Zeit uns noch blieb und wir dort immer gut in der Zeit lagen.

Zum Schluss bin ich sehr zufrieden mit unserer Studienarbeit und würde jederzeit gerne wieder eine Arbeit auf diesem Gebiet schreiben.



# VIII

## Benutzerhandbuch

Verantwortlich: Rico Steffen

| <b>Datum</b> | <b>Version</b> | <b>Änderung</b> |
|--------------|----------------|-----------------|
| 01.07.2006   | 0.1            | Erstellt        |
| 05.07.2006   | 0.2            | Installation    |
| 05.07.2006   | 0.3            | Bedienung       |
| 06.07.2006   | 1.0            | Release         |

In diesem Teil befindet sich die Installations- und Bedienungsanleitung. Eine genauere Auflistung der Änderungen ist im Anhang A.8 zu finden.





## 25 Installationsanleitung

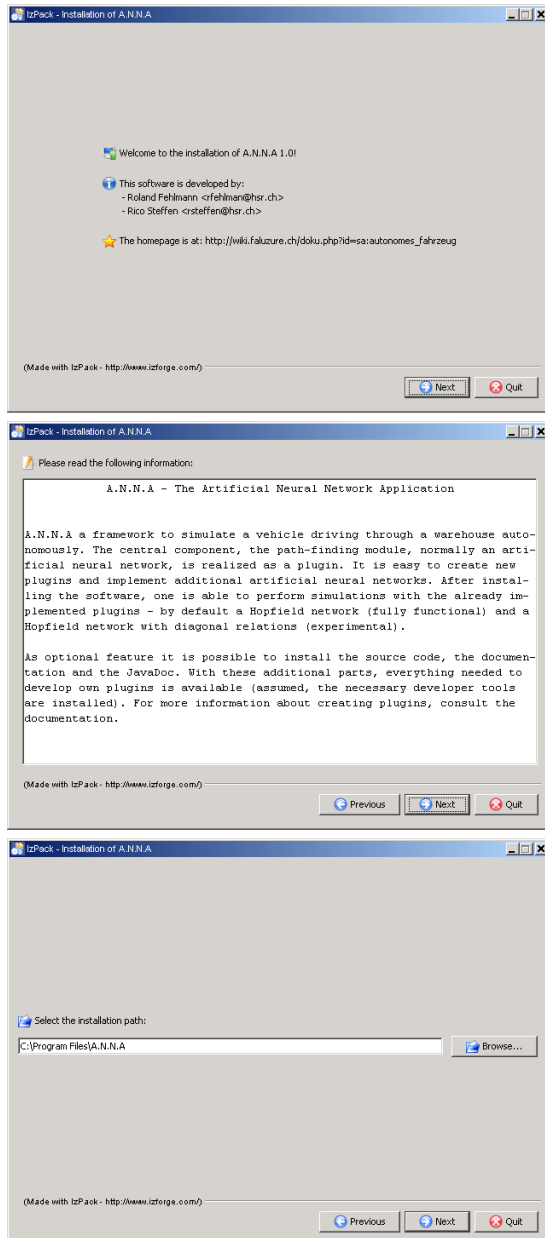


Abbildung 25.1: Installation Schritte 1-3

Die Installation gestaltet sich sehr einfach. Nachdem man das Installations-JAR-File ausgeführt hat – je nach Installation der Java-Umgebung<sup>1</sup> per Doppelklick oder aus der Kommandozeile mit `java -jar A.N.N.A.install.jar` erscheint der IzPack-Installer. Klicken auf «Next» bewirkt die Fortsetzung der Installation. Ein Klick auf «Quit» beendet die Installation und belässt das System unverändert.

Beim nächsten Fenster lesen Sie die Readme-Datei. Danach können Sie mit «Next» die Installation fortsetzen. Auch hier haben Sie die Möglichkeit, die Installation abzubrechen. Mit einem Klick auf «Previous» kommen Sie zurück zum Willkommens-Bildschirm.

Im dritten Installationsfenster können Sie spezifizieren, wohin A.N.N.A installiert werden soll. Als Standardpfad wird das Standardverzeichnis für Programme vorgeschlagen. Sie können natürlich einen beliebigen anderen Pfad angeben.

<sup>1</sup>Java Runtime 5 muss installiert sein. Download: [http://java.com/en/download/windows\\_xpi.jsp](http://java.com/en/download/windows_xpi.jsp)



Das vierte Fenster erlaubt es einem, die gewünschten Komponenten zu installieren. Base ist nötig, die anderen 3 Punkte sind freiwillig. Für Anwender wird Base und Docs empfohlen, Entwickler müssen mindestens die Sources dazustellen. JavaDoc wird für Entwickler zusätzlich empfohlen.

Als Nächstes folgt das Kopieren der Dateien. Bitte warten Sie, bis alle Komponenten kopiert wurden (Statusbalken beachten). Danach können Sie mit «Next» mit der Installation fortfahren.

Als letzter wirklicher Schritt folgt das Erstellen von Verknüpfungen. Bitte geben Sie an, wo und welche Verknüpfungen Sie haben möchten – wenn Sie nicht ganz sicher sind, dürften die Voreinstellungen keine schlechte Wahl sein.

Danach folgt nur noch der Bildschirm mit der Bestätigung, dass die Installation erfolgreich war. Klicken Sie auf «Done» um den Installer zu schliessen.

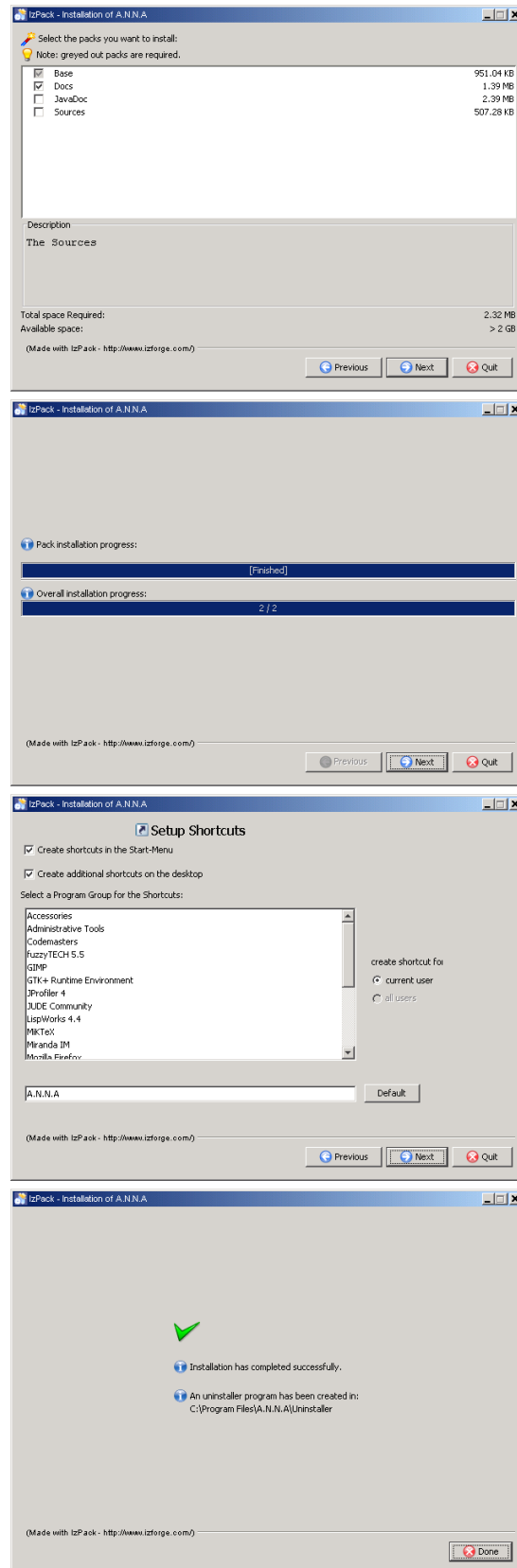


Abbildung 25.2: Installation Schritte 4-7



## 26 Bedienungsanleitung

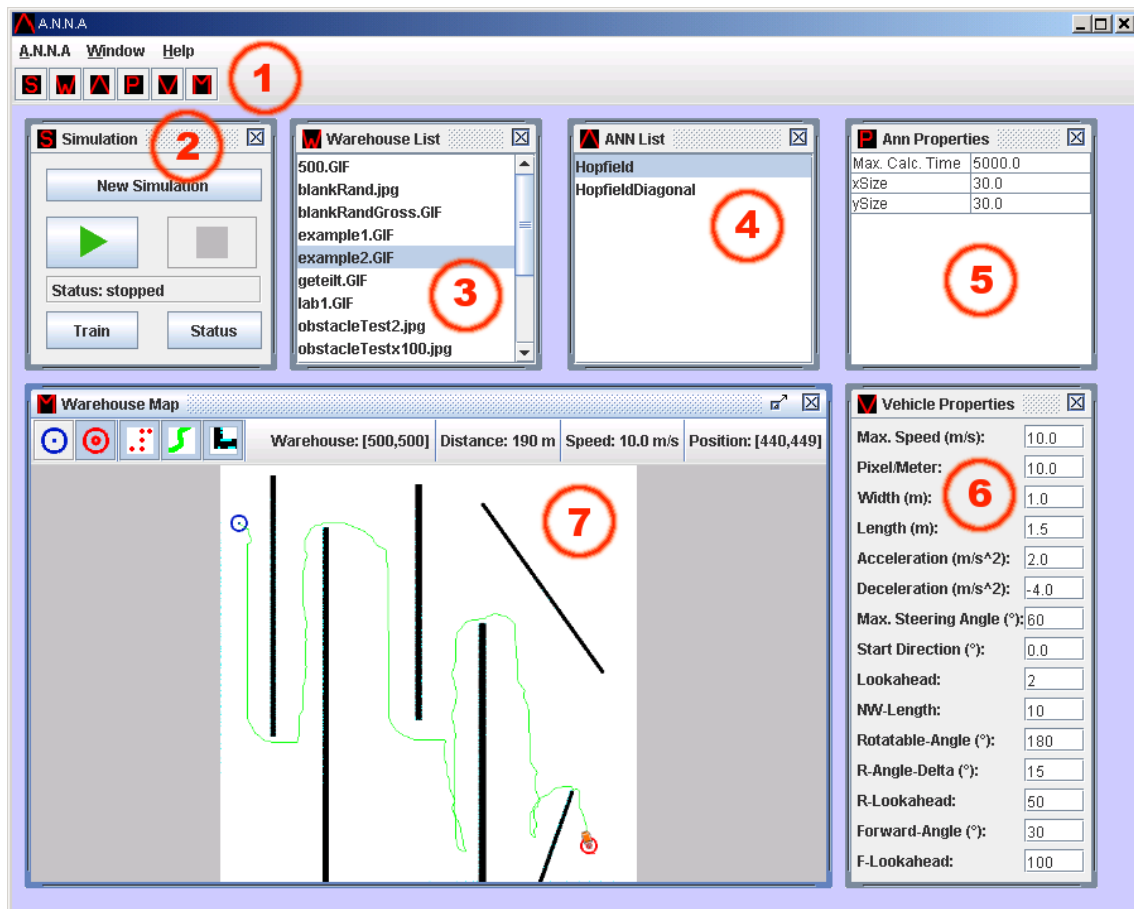


Abbildung 26.1: Screenshot der Applikation

Abbildung 26.1 zeigt die Benutzeroberfläche von A.N.N.A. In den folgenden Abschnitten werden die einzelnen Komponenten (im Bild eingezeichnet: Punkte 1-7) detailliert beschrieben.

### Punkt 1: Menu und Toolbar

**Menu A.N.N.A.** Im Menu «A.N.N.A.» befindet sich ein Eintrag zum Speichern des aktuell geladenen Warenhauses als Bild («Save Map»). Dabei



werden auch das Fahrzeug, der Weg und alle erkannten Hindernisse im Bild eingezeichnet. Der zweite Eintrag im Menu, «Exit», beendet die Applikation.

- Menu Window** Die Einträge im Menu «Window» dienen dazu, die einzelnen Fenster einzublenden. Ausgewählte Fenster erhalten den Fokus und treten in den Vordergrund.
- Menu Help** Im Menu «Help» befindet sich der Menu-Eintrag «About A.N.N.A», welcher den Splash-Screen einblendet. Durch einen Klick auf diesen verschwindet er wieder.
- Buttons der Toolbar** Die sechs Buttons der Toolbar dienen zum ein- und ausblenden der einzelnen Fenster.

## Punkt 2: Fenster Simulation

- Button New Simulation** Der Button «New Simulation» bereitet die Applikation vor, damit eine neue Simulation gestartet werden kann. Konkret bedeutet das, dass alle Einstellungen zurückgesetzt werden.
- Button Play/Pause** Der Button «Play» dient zum Starten der Simulation. Bevor eine Simulation gestartet werden kann, muss jedoch mindestens ein Lagerhaus und ein künstliches neuronales Netz selektiert sein. Zusätzlich muss der Start- und der Zielpunkt festgelegt werden. Ist eine Simulation am laufen, ändert der Button sein Icon und dient der Unterbrechung der Simulation. Mit demselben Button kann auch eine unterbrochene Simulation wiederaufgenommen werden.
- Button Stop** Der «Stopp»-Button bricht eine laufende Simulation ab. Durch anschliessendes Betätigen des Buttons «Play» kann die Simulation mit denselben Einstellungen neu gestartet werden. Erst ein Klick auf «New Simulation» setzt die Konfiguration zurück.
- Status-Anzeige** Die «Status»-Anzeige informiert den Benutzer, wie der aktuelle Status der Applikation ist. Zu Beginn ist der Status immer «new Simulation», wenn eine Simulation läuft, ändert der Status auf «running». Zwischenzeitliches pausieren der Simulation wird durch die Anzeige von «paused» signalisiert. «Stopped» bedeutet, dass die Simulation gestoppt wurde, die Konfiguration aber noch vorhanden ist. Erst ein Klick auf den Button «New Simulation» bringt die Anzeige wieder in die Ausgangsposition.





**Button Train** «Train» ruft die Trainingsfunktion des künstlichen neuronalen Netzes auf. Wird vom selektierten Netz keine solche benötigt oder wenn keine Trainingsfunktion implementiert ist, erscheint eine entsprechende Meldung.

**Button Status** «Status» zeigt den aktuellen Trainingsstand des künstlichen neuronalen Netzes an. Dieser Button hat nur eine Funktion, wenn die entsprechende Abfrage Sinn macht und implementiert wurde.

### **Punkt 3: Fenster Warehouse List**

**Liste** In der Warenhaus-Liste werden alle Bilder als Warenhäuser geladen, welche sich im Verzeichnis `warehouses` befinden. Um eine Simulation durchführen zu können, muss das gewünschte Warenhaus aus der Liste ausgewählt werden.

### **Punkt 4: Fenster ANN List**

**Liste** In dieser Liste werden die verfügbaren Plugins der künstlichen neuronalen Netze angezeigt. Entwickler, welche selber ein Plugin erstellen möchten finden die Informationen dazu im Kapitel 27.1.

### **Punkt 5: Fenster ANN Properties**

**Tabelle mit Eigenschaften** In diesem Fenster werden die konfigurierbaren Parameter des ausgewählten Plugins angezeigt. Diese können je nach Plugin variieren. Beim Hopfield-Plugin sind es zum Beispiel der Berechnungstimeout – eine Zeit in Milisekunden, nach welcher eine Neuberechnung des neuronalen Netzes abgebrochen und mit den bis dahin errechneten Werten gearbeitet werden soll, die Anzahl Neuronen des Netzes in x-Richtung sowie die Anzahl Neuronen in y-Richtung.

### **Punkt 6: Fenster Vehicle Properties**

**Max. Speed** In diesem Feld kann man die maximale Geschwindigkeit des Fahrzeuges festlegen. Die Angabe erfolgt in Metern pro Sekunde.



|                            |  |
|----------------------------|--|
| <b>Pixel/Meter</b>         | Diese Angabe legt den Koeffizienten Anzahl Pixel pro Meter fest. Durch die richtige Angabe kann man die reale Grösse des Warenhauses simulieren.   |
| <b>Width</b>               | Dieser Parameter legt die Breite des Fahrzeuges fest. Die Angabe erfolgt in Metern.  |
| <b>Length</b>              | Hier kann die Länge des Fahrzeuges angegeben werden (in Metern).   |
| <b>Acceleration</b>        | Dieses Feld definiert die Beschleunigung des Fahrzeuges in Metern pro Sekunde im Quadrat.  |
| <b>Deceleration</b>        | Hier wird die Verzögerung in Metern pro Sekunde im Quadrat angegeben.  |
| <b>Max. Steering Angle</b> | Durch diese Angabe legt man den maximalen Steuerungswinkel, welchen das Fahrzeug einschlagen kann, fest. Die Angabe erfolgt in Grad.   |
| <b>Start Direction</b>     | Mit dieser Einstellung definiert man die Fahrzeugrichtung beim Start der Simulation. Die Angabe erfolgt in Grad, $0^\circ$ bedeutet nach rechts, $90^\circ$ nach unten, $180^\circ$ nach links und $270^\circ$ nach oben. Natürlich können auch Zwischenwerte gewählt werden.  |
| <b>Lookahead</b>           | Dieser Parameter definiert, auf welches Neuron des berechneten Weges das Fahrzeug zusteuert. Der Wert 2 bedeutet zum Beispiel, dass, wenn sich das Fahrzeug bei Neuron 3 des Weges befindet, es auf Neuron 5 zusteuert.  |
| <b>NW-Length</b>           | Mit dem Wert bestimmt man, wie viele Neuronen des Weges vorausberechnet werden sollen. Der Wert sollte grösser sein als der Lookahead.   |
| <b>Rotatable-Angle</b>     | Dieser Wert (in Grad) legt fest, in welchem Winkel der rotierende Distanzmesser operieren soll. Ein Winkel von $180^\circ$ bedeutet beispielsweise, dass der ganze Bereich vor dem Fahrzeug ausgeleuchtet werden soll.   |
| <b>R-Angle-Delta</b>       | Dieser Wert wird ebenfalls in Grad angegeben und bestimmt, in welchem Intervall der rotierende Distanzmesser Messungen durchführt. Wird zum Beispiel im obigen Feld ein Winkel von $180^\circ$ angegeben und das Delta auf $15^\circ$ eingestellt, werden pro Umdrehung 13 Messungen durchgeführt. Achtung, die Wahl eines kleinen Intervalls kann die Performance der Applikation erheblich beeinflussen. |



- R-Lookahead** Mit diesem Wert kann angegeben werden, wie viele Pixel weit der rotierende Distanzmesser messen kann. Bei einem Wert von 50 werden Hindernisse zum Beispiel erst erkannt, wenn sich das Fahrzeug ihnen bis auf 50 Pixel genähert hat. Die Auswahl eines grossen Wertes kann die Performance der Applikation beeinflussen.
- Forward-Angle** Hier kann der Winkel (in Grad) angegeben werden, in welchem die beiden vorne angebrachten Sensoren fixiert werden sollen. Ideal ist, wenn die Sensoren ein bisschen seitlich eingestellt werden: 45°-60° dürfte nicht schlecht sein.
- F-Lookahead** Dieser Lookahead bestimmt die Entfernung in Pixeln, in welcher die beiden Frontsensoren noch Hindernisse erkennen können.

## Punkt 7: Fenster Warehouse-Map

- Button Startpunkt** Nach Auswahl dieses Buttons (blauer Kreis) kann im geladenen Warenhaus der Startpunkt festgelegt werden.
- Button Zielpunkt** Nach Anklicken dieses Buttons (roter Kreis) kann der Zielpunkt festgelegt werden.
- Button Netz anzeigen** Dieser Button (rote Punkte) aktiviert die Anzeige der vorausberechneten Wegpunkte des neuronalen Netzes. Er kann auch ein- und ausgeschaltet werden, während die Simulation läuft.
- Button Weg anzeigen** Hier (Button mit grüner Linie) kann man die Anzeige des Weges, welchen das Fahrzeug zurückgelegt hat, ein- und ausschalten. Auch das kann geschehen, während die Simulation ausgeführt wird.
- Button Hindernisse anzeigen** Mit diesem Button (Schwarzes Symbol mit türkis Punkten) kann man ein- und ausschalten, ob die Hindernisse, welche von den Sensoren erkannt wurden, angezeigt werden sollen. Erkannte Punkte werden im Bild türkis eingefärbt. Bei hochauflösenden Bildern, welche in einem kleinen Fenster dargestellt werden, kann die Anzeige der Hindernisse sehr klein sein.
- Statistik-Anzeigen** Die vier Statistik-Anzeigen liefern Informationen zur aktuellen Simulation. In der linken wird angezeigt, wie gross das geladene Warenhaus ist (in Pixeln). Die Distance-Anzeige gibt an, wieviele Meter das Fahrzeug bereits zurückgelegt hat, Speed informiert



über die aktuelle Fahrzeuggeschwindigkeit (in Metern pro Sekunde) und Position gibt an, an welcher Position im Warenhaus sich das Fahrzeug aktuell befindet.



## 27 Entwickler-Dokumentation

### 27.1 Plugin-Anleitung

Um ein eigenes Plugin zu entwickeln, müssen folgende Schritte durchgeführt werden:

#### 1. Package erstellen

Das Package wird am besten im Ordner `plugins/src` angelegt. Dazu kann einfach ein neuer Ordner angelegt werden, analog zu den Ordnern `Hopfield` und `HopfieldDiagonal`. Dabei muss der Ordner den gleichen Namen haben wie die Klasse, welche das Interface `ArtificialNeuralNetwork` implementiert.

#### 2. Klasse anlegen, welche das Interface `ArtificialNeuralNetwork` implementiert

Hat man das Package erstellt, kann man die Java-Klasse anlegen, welche das Interface implementiert. Dabei ist zu beachten, dass die Klasse über die richtige Packagezugehörigkeit verfügt (Befehl `package Packagename`; – wobei `Packagename` der Name des vorhin erstellten Ordners ist) und dass sie selbst denselben Namen hat wie das Package. Damit das Interface `ArtificialNeuralNetwork` gefunden wird, muss man die entsprechende Datei importieren: `import ch.hsr.anna.pd.ann.ArtificialNeuralNetwork;`  
Als Beispiel schaut man sich am besten das vorhandene Plugin `Hopfield` an.

#### 3. Das Plugin implementieren

Der nächste Schritt ist die Implementierung der gewünschten Funktionalität. Dabei muss beachtet werden, dass die im Interface definierten Methoden implementiert werden, da sie die Schnittstelle zur Applikation bilden. Ansonsten ist man frei, was die Entwicklung weiterer Methoden oder zusätzlicher Klassen anbelangt.

#### 4. Das Class-File generieren



Der letzte Schritt ist die Erzeugung der .class-Datei mit dem Java-Compiler. Das generierte File muss im Ordner `plugins/bin/Packagename` abgelegt werden.

## 27.2 Building-Anleitung

### Benötigte Bibliotheken

- `Ftrun.jar`
- `junit.jar`
- `swing-layout-1.0.jar`

### Mit Eclipse

1. Eclipse starten.
2. Ein neues «Java-Projekt» erstellen.
3. Die Jar-Dateien in den «Java Build Path» einfügen.
4. A.N.N.A sollte nun kompilierbar sein.
5. Für das Builden der Plugins kann ein weiterer Source-Path im Projekt definiert werden. Bei dieser Lösungen müssen nach dem Builden die generierten Plugin-Verzeichnisse vom bin-Order in den `plugins/bin`-Ordern kopiert werden. Ansonsten kann A.N.N.A die Plugins nicht finden.

### Ohne Eclipse

1. Kommandozeile öffnen.
2. In das Verzeichnis wechseln in welchem die Source-Dateien abgelegt sind.
3. Folgenden Befehl eingeben:  

```
javac ch/hsr/anna/Anna.java -d ../bin
```



```
-cp .;../tools/Ftrun.jar;../tools/swing-layout-1.0.jar;  
../tools/junit.jar
```

4. Für das Builden der Plugins muss ins Plugin-Verzeichnis gewechselt werden (cd ../plugins/bin)
5. Übersetzen mit:  
javac Hopfield/Hopfield.java -cp .;../../bin -d ../bin
6. Nun kann in das Hauptverzeichnis (cd ..) gewechselt werden und mit folgendem Befehl A.N.N.A gestartet werden:  
java -cp bin;plugins/bin;tools/swing-layout-1.0.jar;  
tools/Ftrun.jar;tools/junit.jar ch.hsr.anna.Anna

## 27.3 Fuzzy-Erweiterungs-Anleitung

Die bestehende Fuzzy-Steuerung wurde mit der Software fuzzyTECH 5.50b Professional Edition erstellt. Die hier beschriebene Anleitung bezieht sich auf die Anpassung der Fuzzy-Logik mit dieser Software. Die korrekte Funktion mit einer anderen Version oder gar einer anderen Software kann nicht garantiert werden.

Um die Fuzzy-Logik anzupassen, startet man als erstes die Software fuzzyTECH. Dort öffnet man die Datei src/ch/hsr/anna/pd/vehicle/fuzzy/FUZZY.FTL. Dadurch wird das entsprechende Projekt geladen. Im Projekt selbst dürfen keine neuen Eingangs- oder Ausgangsvariablen definiert werden – ansonsten müsste der Source-Code des Fahrzeuges und anderer Klassen angepasst werden, was hier aber nicht beschrieben wird. Auch der Wertebereich dieser Variablen sollte nicht verändert werden, da die Funktionalität des Fahrzeuges danach nicht mehr gewährleistet werden kann.

Beim Hinzufügen, Löschen und Ändern der Terme innerhalb der Variablen ist man hingegen frei, auch die Regeln können nach Belieben angepasst werden. Informationen bezüglich der bestehenden Fuzzy-Logik findet man in Kapitel 15.8 und im Anhang B.

Hat man die Variablen und Regeln angepasst, gilt es den Java-Code zu generieren. Das geschieht via Menu Extras->Kompilieren nach->Java... Im Dialog wählt man als Zieldatei die Java-Datei src/ch/hsr/anna/pd/vehicle/fuzzy/FUZZY.java und startet die Generierung. Im Anschluss an die Codegenerierung muss man in der generierten Datei noch die Package-Deklaration hinzufügen. Dazu öffnet man die Datei src/ch/hsr/anna/pd/vehicle/fuzzy/FUZZY.java mit einem Editor und fügt ganz zu oberst die Zeile package ch.hsr.anna.pd.vehicle; hinzu.



Wenn man nicht ganz sicher ist, tut man gut daran, sich die bereits vorhandene Datei vor dem Generieren anzuschauen oder sich eine Sicherungskopie davon anzulegen.



**IX**

**Anhang**





# A Änderungen

In diesem Kapitel werden die Änderungen an den einzelnen Dokumentteilen genauer beschrieben.

## A.1 Projektplan

### **v0.1 Erstellt** 06.04.2006

- Der Dokumentteil wurde erstellt und eine erste Struktur festgelegt.

### **v0.2 Risikoanalyse** 06.04.2006

- Eine Risikoanalyse mit Risikomanagement wurde erstellt.
- Die Projektübersicht wurde geschrieben.
- Die Organisation des Projektes wurde festgelegt.

### **v0.3 Planung** 10.04.2006

- Ein grober Zeitplan wurde festgelegt.
- Der Detailzeitplan wurde in einer Excel-Tabelle aufgestellt, mit ersten Sollzeiten für die Startphase.
- Die verschiedenen Meilensteine der Studienarbeit wurden definiert und beschrieben.
- Das Projekt wurde in verschiedene Arbeitspakete unterteilt und diese jeweils genauer beschrieben.

### **v0.4 Infrastruktur** 13.04.2006

- Die verwendete Infrastruktur wurde aufgezeichnet.
- Die Software zur Dokumentation und Entwicklung wurde aufgeführt.



### **v0.5 Review** 17.04.2006

- Der Projektplan wurde nochmals überarbeitet und für das Review fertiggestellt.

### **v1.0 Release** 06.07.2006

- Der Projektplan wurde für die Abgabe fertiggestellt.

## **A.2 Anforderungsspezifikation**

### **v0.1 Erstellt** 27.04.2006

- Der Dokumentteil wurde erstellt und eine erste Struktur festgelegt.

### **v0.2 Use-Cases & UI** 01.05.2006

- Die Use Cases wurden definiert und beschrieben.
- Ein Use Case Diagramm wurde erstellt.
- Es wurden zwei Benutzerrollen definiert.
- User Interface Skizzen wurden eingefügt.

### **v0.3 Diverses** 02.05.2006

- Die Allgemeine Beschreibung der Anforderungen wurden geschrieben.
- Die Working Environment wurde definiert.

### **v0.4 Nichtfunktionale Anforderungen** 04.05.2006

- Es wurden die nichtfunktionalen Anforderungen spezifiziert.
- Die Benutzerrollen wurden beschrieben.

### **v0.5 Korrekturen** 08.05.2006

- Kleine Änderungen der Anforderungen.
- Klarer ausgedrückt, was realisiert werden soll.
- UC «KNN trainieren» angepasst.



## **v1.0 Release** 06.07.2006

- Die Anforderungsspezifikation wurde für die Abgabe fertiggestellt.

## **A.3 Analyse**

### **v0.1 Erstellt** 08.05.2006

- Der Dokumentteil wurde erstellt und eine erste Struktur festgelegt.

### **v0.2 Domain Model** 09.05.2006

- Das Domain Model wurde erstellt.
- Die einzelnen Klassen des Domain Models sind beschrieben.

### **v0.3 SSD** 09.05.2006

- System Sequenz Diagramme erstellt und eingefügt.

### **v0.4 Allgemein** 11.05.2006

- System Sequenz Diagramme fertig gestellt.
- Korrekturen vorgenommen.

### **v0.5 Position** 16.05.2006

- Position ins Domain Model hinzugefügt.

## **v1.0 Release** 06.07.2006

- Die Analyse wurde für die Abgabe fertiggestellt.



## A.4 Architektur & Design

### v0.1 Erstellt 14.05.2006

- Der Dokumentteil wurde erstellt und eine erste Struktur festgelegt.

### v0.2 Architektur 15.05.2006

- Das Diagramm wurde eingefügt.
- Die einzelnen Pakete wurden beschrieben.

### v0.3 Erstes Design 30.05.2006

- Das Design Diagramm der Problem Domain wurde eingefügt.
- Erste Beschreibungen geschrieben.

### v0.4 Design UI 05.06.2006

- User Interface Design eingefügt.
- Verbesserungen am ursprünglichen Design.
- Persistency Design fertig.

### v0.5 Design 15.06.2006

- Einzelne Diagramme erstellt.
- Hopfield Design fertig.
- Weitere Design Verbesserungen und Änderungen.

### v0.6 Design 23.06.2006

- Design fertig gestellt.
- Diagramme verbessert.
- Genauere Beschreibungen erstellt.
- Dokumentteil erweitert.

### v0.7 Fahrzeug 30.06.2006

- Fahrzeug fertig beschrieben.



- User Interface genauer beschrieben.
- Fuzzy-Logik beschrieben und im Anhang eingefügt.
- Problem Domain besser beschrieben.

#### **v0.8 Threads** 01.07.2006

- Threads beschrieben.
- Fuzzy umgeschrieben.
- Plugins beschrieben.
- Hopfield genauer beschrieben, mit Bilder erklärt.

#### **v0.9 Entscheide** 03.07.2006

- Entscheide hinzugefügt.
- Entscheide gschrieben.
- Design-Teil ansonsten fertig gestellt.
- Koordinatensystem erklärt.

#### **v1.0 Release** 06.07.2006

- Die Architektur und das Design wurden für die Abgabe fertiggestellt.

## **A.5 Resultate & Erkenntnisse**

#### **v0.1 Erstellt** 01.07.2006

- Der Dokumentteil wurde erstellt und eine erste Struktur festgelegt.

#### **v0.2 Fuzzy** 03.07.2006

- Fuzzy Resultate und Erkenntnisse geschrieben.
- Hopfield Resultate und Erkenntnisse geschrieben.
- A.N.N.A Resultate und Erkenntnisse geschrieben.

#### **v0.3 A.N.N.A** 04.07.2006

- A.N.N.A Resultate und Erkenntnisse verbessert.



- Lookahead hinzugefügt.
- Alle Punkte erweitert.

#### **v1.0 Release 06.07.2006**

- Die Resultate und Erkenntnisse wurden für die Abgabe fertiggestellt.

## **A.6 Qualitätssicherung**

#### **v0.1 Erstellt 01.07.2006**

- Der Dokumentteil wurde erstellt und eine erste Struktur festgelegt.

#### **v0.2 Management 02.07.2006**

- Methoden zur Qualitätssicherung hinzugefügt.
- Getestete Ebenen geschrieben
- Management erstellt.

#### **v1.0 Release 06.07.2006**

- Die Qualitätssicherung wurde für die Abgabe fertiggestellt.

## **A.7 Persönliche Erfahrungen**

#### **v0.1 Erstellt 01.07.2006**

- Der Dokumentteil wurde erstellt.

#### **v0.2 Rico Steffen 04.07.2006**

- Persönlicher Bericht geschrieben.





### **v0.3 Roland Fehlmann 05.07.2006**

- Persönlicher Bericht geschrieben.

### **v1.0 Release 06.07.2006**

- Die Berichte wurden für die Abgabe fertiggestellt.

## **A.8 Benutzerhandbuch**

### **v0.1 Erstellt 01.07.2006**

- Der Dokumentteil wurde erstellt und eine erste Struktur festgelegt.

### **v0.2 Installation 05.07.2006**

- Installationsanleitung geschrieben
- Entwickler-Dokumentation begonnen.

### **v0.3 Bedienung 05.07.2006**

- Bedienungsanleitung erstellt.
- Entwickler-Dokumentation fertiggestellt.

### **v1.0 Release 06.07.2006**

- Die Berichte wurden für die Abgabe fertiggestellt.





# B Fuzzy-Logik Einstellungen

## B.1 Systemstruktur

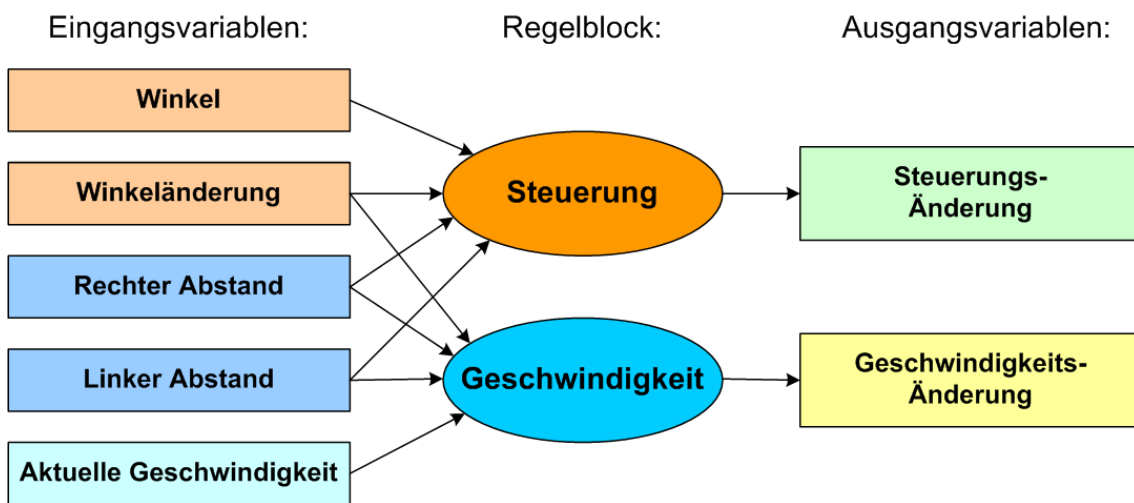


Abbildung B.1: Fuzzy Systemstruktur

Die Systemstruktur B.1 beschreibt den Datenfluss in dem Fuzzy System. Es besteht aus fünf Eingangs- und zwei Ausgangs-Variablen, sowie zwei Regelblöcken. Die genauen Einstellungen werden in den nächsten Abschnitten aufgeführt.

## B.2 Eingangs- & Ausgangs-Variablen

In der Tabelle B.1 sind die Eingangs- und in der Tabelle B.2 die Ausgangs-Variablen des Fuzzy-Systems aufgelistet.



| # | Variablenname | Einheit | Min  | Max | Default | Termnamen   |
|---|---------------|---------|------|-----|---------|---|
| 1 | Angle         | Degree  | 0    | 360 | 0       | StrongLeft<br>Left<br>Front<br>Right<br>StrongRight |
| 2 | AngleDelta    | Grad    | -180 | 180 | 0       | Left<br>None<br>StrongLeft<br>StrongRight<br>Right  |
| 3 | CurrentSpeed  | Prozent | 0    | 100 | 0       | low<br>mid<br>high                                  |
| 4 | LeftDistance  | Units   | 0    | 100 | 100     | nah<br>mittel<br>weit                               |
| 5 | RightDistance | Units   | 0    | 100 | 100     | nah<br>mittel<br>weit                               |

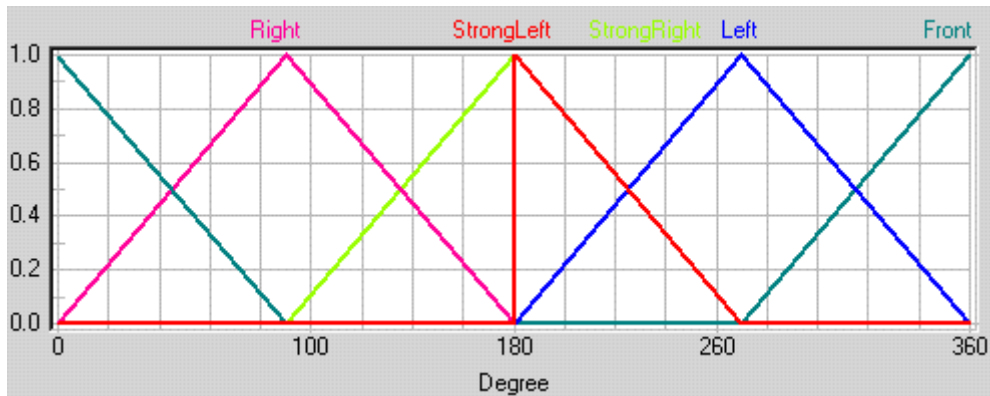
Tabelle B.1: Eingangs-Variablen

| # | Variablenname | Einheit | Min  | Max | Default | Termnamen   |
|---|---------------|---------|------|-----|---------|---|
| 6 | SpeedDelta    | Units   | -100 | 100 | 0       | Break<br>Maintain<br>Accelerate                       |
| 7 | SteeringDelta | Grad    | -90  | 90  | 0       | Left<br>StrongLeft<br>Correct<br>Right<br>StrongRight |

Tabelle B.2: Ausgangs-Variablen

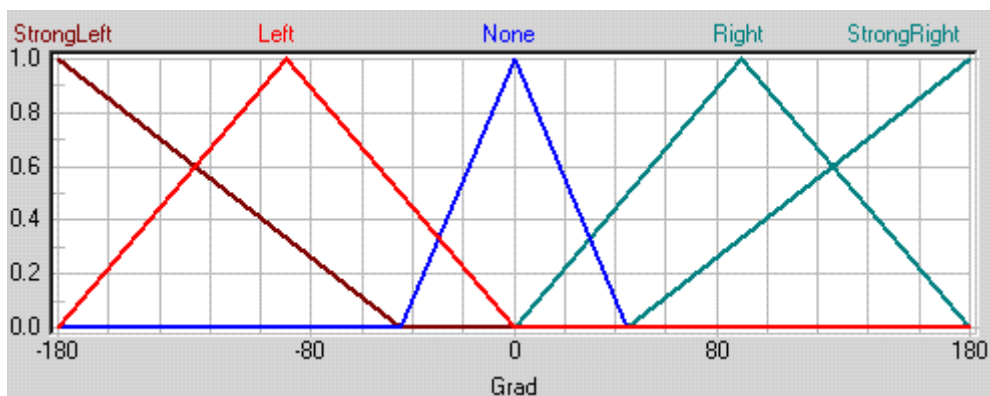


## Eingangsvariable: Angle



| Termname    | Form/Par. | Definitionspunkte (x, y)                              |
|-------------|-----------|---|
| StrongLeft  | L-Form    | (0, 0) (180, 0) (180, 1)<br>(270, 0) (360, 0)         |
| Left        | L-Form    | (0, 0) (180, 0) (270, 1)<br>(360, 0)                  |
| Front       | L-Form    | (0, 0.99152) (90, 0) (182.98, 0)<br>(270, 0) (360, 1) |
| Right       | L-Form    | (0, 0) (90, 1) (180, 0)<br>(360, 0)                   |
| StrongRight | L-Form    | (0, 0) (90, 0) (180, 1)<br>(180, 0) (360, 0)          |

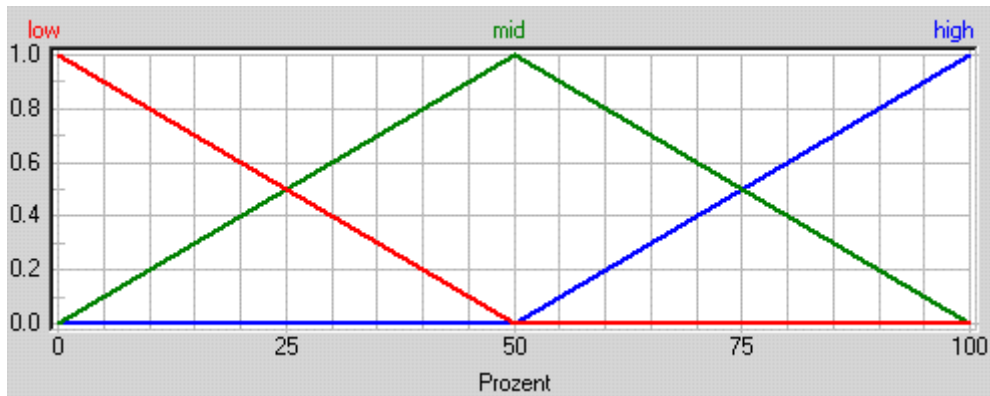
## Eingangsvariable: AngleDelta





| Termname    | Form/Par. | Definitionspunkte (x, y)                      |
|-------------|-----------|---|
| Left        | L-Form    | (-180, 0) (-90, 1) (0, 0)<br>(180, 0)         |
| None        | L-Form    | (-180, 0) (-45, 0) (0, 1)<br>(45, 0) (180, 0) |
| StrongLeft  | L-Form    | (-180, 1) (-45, 0) (180, 0)                   |
| StrongRight | L-Form    | (-180, 0) (45, 0) (180, 1)                    |
| Right       | L-Form    | (-180, 0) (0, 0) (90, 1)<br>(180, 0)          |

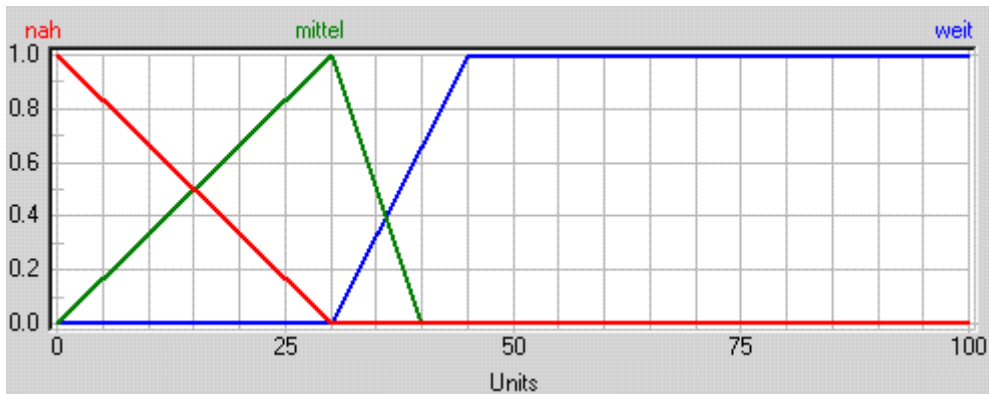
### Eingangsvariable: CurrentSpeed



| Termname | Form/Par. | Definitionspunkte (x, y) |
|----------|-----------|--------------------------|
| low      | L-Form    | (0, 1) (50, 0) (100, 0)  |
| mid      | L-Form    | (0, 0) (50, 1) (100, 0)  |
| high     | L-Form    | (0, 0) (50, 0) (100, 1)  |

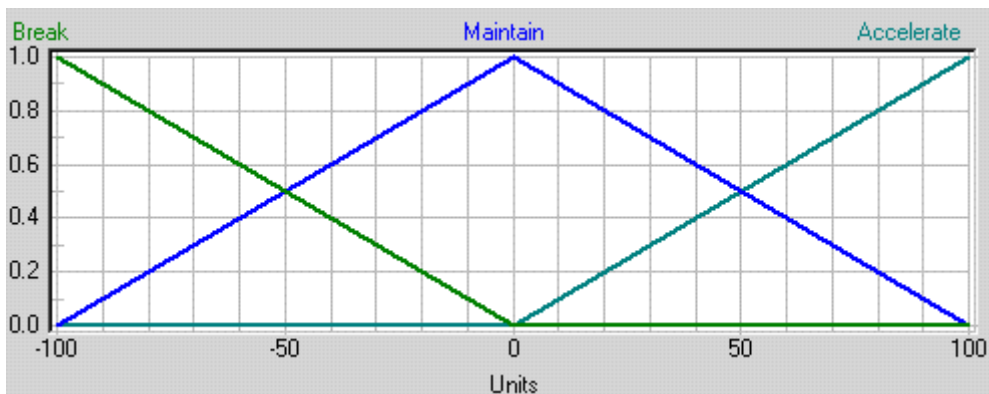


## Eingangsvariablen: LeftDistance & RightDistance



| Termname | Form/Par. | Definitionspunkte (x, y)              |
|----------|-----------|---------------------------------------|
| nah      | L-Form    | (0, 1) (30, 0) (100, 0)               |
| mittel   | L-Form    | (0, 0) (30, 1) (40, 0) (100, 0)       |
| weit     | L-Form    | (0, 0) (30, 0) (45, 0.99146) (100, 1) |

## Ausgangsvariable: SpeedDelta





| Termname   | Form/Par. | Definitionspunkte (x, y)  |
|------------|-----------|---------------------------|
| Break      | L-Form    | (-100, 1) (0, 0) (100, 0) |
| Maintain   | L-Form    | (-100, 0) (0, 1) (100, 0) |
| Accelerate | L-Form    | (-100, 0) (0, 0) (100, 1) |

## Ausgangsvariable: SteeringDelta



| Termname    | Form/Par. | Definitionspunkte (x, y)                    |
|-------------|-----------|---|
| Left        | L-Form    | (-90, 0) (-60, 1) (0, 0)<br>(90, 0)         |
| StrongLeft  | L-Form    | (-90, 1) (-45, 0) (90, 0)                   |
| Correct     | L-Form    | (-90, 0) (-45, 0) (0, 1)<br>(45, 0) (90, 0) |
| Right       | L-Form    | (-90, 0) (0, 0) (60, 1)<br>(90, 0)          |
| StrongRight | L-Form    | (-90, 0) (45, 0) (90, 1)                    |

## B.3 Regelblöcke

In den folgenden Tabellen sind die Regeln der zwei Regelblöcke Steering und Speed aufgelistet.

### Steering

Auf den nächsten Seiten sind die Regeln des Steering-Blockes abgebildet:





| WENN       |             |              |               | DANN |               |
|------------|-------------|--------------|---------------|------|---------------|
| Angle      | AngleDelta  | LeftDistance | RightDistance | DoS  | SteeringDelta |
| StrongLeft | Left        | nah          | nah           | 1    | StrongLeft    |
| StrongLeft | Left        | nah          | mittel        | 1    | StrongRight   |
| StrongLeft | Left        | nah          | weit          | 1    | StrongRight   |
| StrongLeft | Left        | mittel       | nah           | 1    | StrongLeft    |
| StrongLeft | Left        | mittel       | mittel        | 1    | StrongLeft    |
| StrongLeft | Left        | mittel       | weit          | 1    | StrongLeft    |
| StrongLeft | Left        | weit         | nah           | 1    | StrongLeft    |
| StrongLeft | Left        | weit         | mittel        | 1    | StrongLeft    |
| StrongLeft | Left        | weit         | weit          | 1    | StrongLeft    |
| StrongLeft | None        | nah          | nah           | 1    | StrongLeft    |
| StrongLeft | None        | nah          | mittel        | 1    | StrongRight   |
| StrongLeft | None        | nah          | weit          | 1    | StrongRight   |
| StrongLeft | None        | mittel       | nah           | 1    | StrongLeft    |
| StrongLeft | None        | mittel       | mittel        | 1    | Left          |
| StrongLeft | None        | mittel       | weit          | 1    | Left          |
| StrongLeft | None        | weit         | nah           | 1    | StrongLeft    |
| StrongLeft | None        | weit         | mittel        | 1    | Left          |
| StrongLeft | None        | weit         | weit          | 1    | Left          |
| StrongLeft | StrongLeft  | nah          | nah           | 1    | StrongLeft    |
| StrongLeft | StrongLeft  | nah          | mittel        | 1    | StrongRight   |
| StrongLeft | StrongLeft  | nah          | weit          | 1    | StrongRight   |
| StrongLeft | StrongLeft  | mittel       | nah           | 1    | StrongLeft    |
| StrongLeft | StrongLeft  | mittel       | mittel        | 1    | StrongLeft    |
| StrongLeft | StrongLeft  | mittel       | weit          | 1    | StrongLeft    |
| StrongLeft | StrongLeft  | weit         | nah           | 1    | StrongLeft    |
| StrongLeft | StrongLeft  | weit         | mittel        | 1    | StrongLeft    |
| StrongLeft | StrongLeft  | weit         | weit          | 1    | StrongLeft    |
| StrongLeft | StrongRight | nah          | nah           | 1    | StrongLeft    |
| StrongLeft | StrongRight | nah          | mittel        | 1    | StrongRight   |
| StrongLeft | StrongRight | nah          | weit          | 1    | StrongRight   |
| StrongLeft | StrongRight | mittel       | nah           | 1    | StrongLeft    |
| StrongLeft | StrongRight | mittel       | mittel        | 1    | Correct       |
| StrongLeft | StrongRight | mittel       | weit          | 1    | Correct       |
| StrongLeft | StrongRight | weit         | nah           | 1    | StrongLeft    |
| StrongLeft | StrongRight | weit         | mittel        | 1    | Correct       |
| StrongLeft | StrongRight | weit         | weit          | 1    | Correct       |
| StrongLeft | Right       | nah          | nah           | 1    | StrongLeft    |
| StrongLeft | Right       | nah          | mittel        | 1    | StrongRight   |
| StrongLeft | Right       | nah          | weit          | 1    | StrongRight   |
| StrongLeft | Right       | mittel       | nah           | 1    | StrongLeft    |
| StrongLeft | Right       | mittel       | mittel        | 1    | Correct       |
| StrongLeft | Right       | mittel       | weit          | 1    | Correct       |



| WENN       |             |              |               | DANN |               |
|------------|-------------|--------------|---------------|------|---------------|
| Angle      | AngleDelta  | LeftDistance | RightDistance | DoS  | SteeringDelta |
| StrongLeft | Right       | weit         | nah           | 1    | StrongLeft    |
| StrongLeft | Right       | weit         | mittel        | 1    | Correct       |
| StrongLeft | Right       | weit         | weit          | 1    | Correct       |
| Left       | Left        | nah          | nah           | 1    | StrongLeft    |
| Left       | Left        | nah          | mittel        | 1    | StrongRight   |
| Left       | Left        | nah          | weit          | 1    | StrongRight   |
| Left       | Left        | mittel       | nah           | 1    | StrongLeft    |
| Left       | Left        | mittel       | mittel        | 1    | Left          |
| Left       | Left        | mittel       | weit          | 1    | Left          |
| Left       | Left        | weit         | nah           | 1    | StrongLeft    |
| Left       | Left        | weit         | mittel        | 1    | Left          |
| Left       | Left        | weit         | weit          | 1    | Left          |
| Left       | None        | nah          | nah           | 1    | StrongLeft    |
| Left       | None        | nah          | mittel        | 1    | StrongRight   |
| Left       | None        | nah          | weit          | 1    | StrongRight   |
| Left       | None        | mittel       | nah           | 1    | StrongLeft    |
| Left       | None        | mittel       | mittel        | 1    | Left          |
| Left       | None        | mittel       | weit          | 1    | Left          |
| Left       | None        | weit         | nah           | 1    | StrongLeft    |
| Left       | None        | weit         | mittel        | 1    | Left          |
| Left       | None        | weit         | weit          | 1    | Left          |
| Left       | StrongLeft  | nah          | nah           | 1    | StrongLeft    |
| Left       | StrongLeft  | nah          | mittel        | 1    | StrongRight   |
| Left       | StrongLeft  | nah          | weit          | 1    | StrongRight   |
| Left       | StrongLeft  | mittel       | nah           | 1    | StrongLeft    |
| Left       | StrongLeft  | mittel       | mittel        | 1    | StrongLeft    |
| Left       | StrongLeft  | mittel       | weit          | 1    | StrongLeft    |
| Left       | StrongLeft  | weit         | nah           | 1    | StrongLeft    |
| Left       | StrongLeft  | weit         | mittel        | 1    | StrongLeft    |
| Left       | StrongLeft  | weit         | weit          | 1    | StrongLeft    |
| Left       | StrongRight | nah          | nah           | 1    | StrongLeft    |
| Left       | StrongRight | nah          | mittel        | 1    | StrongRight   |
| Left       | StrongRight | nah          | weit          | 1    | StrongRight   |
| Left       | StrongRight | mittel       | nah           | 1    | StrongLeft    |
| Left       | StrongRight | mittel       | mittel        | 1    | Correct       |
| Left       | StrongRight | mittel       | weit          | 1    | Correct       |
| Left       | StrongRight | weit         | nah           | 1    | StrongLeft    |
| Left       | StrongRight | weit         | mittel        | 1    | Correct       |
| Left       | StrongRight | weit         | weit          | 1    | Correct       |
| Left       | Right       | nah          | nah           | 1    | StrongLeft    |
| Left       | Right       | nah          | mittel        | 1    | StrongRight   |
| Left       | Right       | nah          | weit          | 1    | StrongRight   |



| WENN  |             |              |               | DANN |               |
|-------|-------------|--------------|---------------|------|---------------|
| Angle | AngleDelta  | LeftDistance | RightDistance | DoS  | SteeringDelta |
| Left  | Right       | mittel       | nah           | 1    | StrongLeft    |
| Left  | Right       | mittel       | mittel        | 1    | Correct       |
| Left  | Right       | mittel       | weit          | 1    | Correct       |
| Left  | Right       | weit         | nah           | 1    | StrongLeft    |
| Left  | Right       | weit         | mittel        | 1    | Correct       |
| Left  | Right       | weit         | weit          | 1    | Correct       |
| Front | Left        | nah          | nah           | 1    | StrongLeft    |
| Front | Left        | nah          | mittel        | 1    | StrongRight   |
| Front | Left        | nah          | weit          | 1    | StrongRight   |
| Front | Left        | mittel       | nah           | 1    | StrongLeft    |
| Front | Left        | mittel       | mittel        | 1    | Left          |
| Front | Left        | mittel       | weit          | 1    | Left          |
| Front | Left        | weit         | nah           | 1    | StrongLeft    |
| Front | Left        | weit         | mittel        | 1    | Left          |
| Front | Left        | weit         | weit          | 1    | Left          |
| Front | None        | nah          | nah           | 1    | StrongLeft    |
| Front | None        | nah          | mittel        | 1    | StrongRight   |
| Front | None        | nah          | weit          | 1    | StrongRight   |
| Front | None        | mittel       | nah           | 1    | StrongLeft    |
| Front | None        | mittel       | mittel        | 1    | Correct       |
| Front | None        | mittel       | weit          | 1    | Correct       |
| Front | None        | weit         | nah           | 1    | StrongLeft    |
| Front | None        | weit         | mittel        | 1    | Correct       |
| Front | None        | weit         | weit          | 1    | Correct       |
| Front | StrongLeft  | nah          | nah           | 1    | StrongLeft    |
| Front | StrongLeft  | nah          | mittel        | 1    | StrongRight   |
| Front | StrongLeft  | nah          | weit          | 1    | StrongRight   |
| Front | StrongLeft  | mittel       | nah           | 1    | StrongLeft    |
| Front | StrongLeft  | mittel       | mittel        | 1    | StrongLeft    |
| Front | StrongLeft  | mittel       | weit          | 1    | StrongLeft    |
| Front | StrongLeft  | weit         | nah           | 1    | StrongLeft    |
| Front | StrongLeft  | weit         | mittel        | 1    | StrongLeft    |
| Front | StrongLeft  | weit         | weit          | 1    | StrongLeft    |
| Front | StrongRight | nah          | nah           | 1    | StrongRight   |
| Front | StrongRight | nah          | mittel        | 1    | StrongRight   |
| Front | StrongRight | nah          | weit          | 1    | StrongRight   |
| Front | StrongRight | mittel       | nah           | 1    | StrongLeft    |
| Front | StrongRight | mittel       | mittel        | 1    | StrongRight   |
| Front | StrongRight | mittel       | weit          | 1    | StrongRight   |
| Front | StrongRight | weit         | nah           | 1    | StrongLeft    |
| Front | StrongRight | weit         | mittel        | 1    | StrongRight   |
| Front | StrongRight | weit         | weit          | 1    | StrongRight   |



| WENN  |             |              |               | DANN |               |
|-------|-------------|--------------|---------------|------|---------------|
| Angle | AngleDelta  | LeftDistance | RightDistance | DoS  | SteeringDelta |
| Front | Right       | nah          | nah           | 1    | StrongRight   |
| Front | Right       | nah          | mittel        | 1    | StrongRight   |
| Front | Right       | nah          | weit          | 1    | StrongRight   |
| Front | Right       | mittel       | nah           | 1    | StrongLeft    |
| Front | Right       | mittel       | mittel        | 1    | Right         |
| Front | Right       | mittel       | weit          | 1    | Right         |
| Front | Right       | weit         | nah           | 1    | StrongLeft    |
| Front | Right       | weit         | mittel        | 1    | Right         |
| Front | Right       | weit         | weit          | 1    | Right         |
| Right | Left        | nah          | nah           | 1    | StrongRight   |
| Right | Left        | nah          | mittel        | 1    | StrongRight   |
| Right | Left        | nah          | weit          | 1    | StrongRight   |
| Right | Left        | mittel       | nah           | 1    | StrongLeft    |
| Right | Left        | mittel       | mittel        | 1    | Correct       |
| Right | Left        | mittel       | weit          | 1    | Correct       |
| Right | Left        | weit         | nah           | 1    | StrongLeft    |
| Right | Left        | weit         | mittel        | 1    | Correct       |
| Right | Left        | weit         | weit          | 1    | Correct       |
| Right | None        | nah          | nah           | 1    | StrongRight   |
| Right | None        | nah          | mittel        | 1    | StrongRight   |
| Right | None        | nah          | weit          | 1    | StrongRight   |
| Right | None        | mittel       | nah           | 1    | StrongLeft    |
| Right | None        | mittel       | mittel        | 1    | Right         |
| Right | None        | mittel       | weit          | 1    | Right         |
| Right | None        | weit         | nah           | 1    | StrongLeft    |
| Right | None        | weit         | mittel        | 1    | Right         |
| Right | None        | weit         | weit          | 1    | Right         |
| Right | StrongLeft  | nah          | nah           | 1    | StrongRight   |
| Right | StrongLeft  | nah          | mittel        | 1    | StrongRight   |
| Right | StrongLeft  | nah          | weit          | 1    | StrongRight   |
| Right | StrongLeft  | mittel       | nah           | 1    | StrongLeft    |
| Right | StrongLeft  | mittel       | mittel        | 1    | Correct       |
| Right | StrongLeft  | mittel       | weit          | 1    | Correct       |
| Right | StrongLeft  | weit         | nah           | 1    | StrongLeft    |
| Right | StrongLeft  | weit         | mittel        | 1    | Correct       |
| Right | StrongLeft  | weit         | weit          | 1    | Correct       |
| Right | StrongRight | nah          | nah           | 1    | StrongRight   |
| Right | StrongRight | nah          | mittel        | 1    | StrongRight   |
| Right | StrongRight | nah          | weit          | 1    | StrongRight   |
| Right | StrongRight | mittel       | nah           | 1    | StrongLeft    |
| Right | StrongRight | mittel       | mittel        | 1    | StrongRight   |
| Right | StrongRight | mittel       | weit          | 1    | StrongRight   |



| WENN        |             |              |               | DANN |               |
|-------------|-------------|--------------|---------------|------|---------------|
| Angle       | AngleDelta  | LeftDistance | RightDistance | DoS  | SteeringDelta |
| Right       | StrongRight | weit         | nah           | 1    | StrongLeft    |
| Right       | StrongRight | weit         | mittel        | 1    | StrongRight   |
| Right       | StrongRight | weit         | weit          | 1    | StrongRight   |
| Right       | Right       | nah          | nah           | 1    | StrongRight   |
| Right       | Right       | nah          | mittel        | 1    | StrongRight   |
| Right       | Right       | nah          | weit          | 1    | StrongRight   |
| Right       | Right       | mittel       | nah           | 1    | StrongLeft    |
| Right       | Right       | mittel       | mittel        | 1    | Right         |
| Right       | Right       | mittel       | weit          | 1    | Right         |
| Right       | Right       | weit         | nah           | 1    | StrongLeft    |
| Right       | Right       | weit         | mittel        | 1    | Right         |
| Right       | Right       | weit         | weit          | 1    | Right         |
| StrongRight | Left        | nah          | nah           | 1    | StrongRight   |
| StrongRight | Left        | nah          | mittel        | 1    | StrongRight   |
| StrongRight | Left        | nah          | weit          | 1    | StrongRight   |
| StrongRight | Left        | mittel       | nah           | 1    | StrongLeft    |
| StrongRight | Left        | mittel       | mittel        | 1    | Correct       |
| StrongRight | Left        | mittel       | weit          | 1    | Correct       |
| StrongRight | Left        | weit         | nah           | 1    | StrongLeft    |
| StrongRight | Left        | weit         | mittel        | 1    | Correct       |
| StrongRight | Left        | weit         | weit          | 1    | Correct       |
| StrongRight | None        | nah          | nah           | 1    | StrongRight   |
| StrongRight | None        | nah          | mittel        | 1    | StrongRight   |
| StrongRight | None        | nah          | weit          | 1    | StrongRight   |
| StrongRight | None        | mittel       | nah           | 1    | StrongLeft    |
| StrongRight | None        | mittel       | mittel        | 1    | Right         |
| StrongRight | None        | mittel       | weit          | 1    | Right         |
| StrongRight | None        | weit         | nah           | 1    | StrongLeft    |
| StrongRight | None        | weit         | mittel        | 1    | Right         |
| StrongRight | None        | weit         | weit          | 1    | Right         |
| StrongRight | StrongLeft  | nah          | nah           | 1    | StrongRight   |
| StrongRight | StrongLeft  | nah          | mittel        | 1    | StrongRight   |
| StrongRight | StrongLeft  | nah          | weit          | 1    | StrongRight   |
| StrongRight | StrongLeft  | mittel       | nah           | 1    | StrongLeft    |
| StrongRight | StrongLeft  | mittel       | mittel        | 1    | Correct       |
| StrongRight | StrongLeft  | mittel       | weit          | 1    | Correct       |
| StrongRight | StrongLeft  | weit         | nah           | 1    | StrongLeft    |
| StrongRight | StrongLeft  | weit         | mittel        | 1    | Correct       |
| StrongRight | StrongLeft  | weit         | weit          | 1    | Correct       |
| StrongRight | StrongRight | nah          | nah           | 1    | StrongRight   |
| StrongRight | StrongRight | nah          | mittel        | 1    | StrongRight   |
| StrongRight | StrongRight | nah          | weit          | 1    | StrongRight   |



| WENN        |             |              |               | DANN |               |
|-------------|-------------|--------------|---------------|------|---------------|
| Angle       | AngleDelta  | LeftDistance | RightDistance | DoS  | SteeringDelta |
| StrongRight | StrongRight | mittel       | nah           | 1    | StrongLeft    |
| StrongRight | StrongRight | mittel       | mittel        | 1    | StrongRight   |
| StrongRight | StrongRight | mittel       | weit          | 1    | StrongRight   |
| StrongRight | StrongRight | weit         | nah           | 1    | StrongLeft    |
| StrongRight | StrongRight | weit         | mittel        | 1    | StrongRight   |
| StrongRight | StrongRight | weit         | weit          | 1    | StrongRight   |
| StrongRight | Right       | nah          | nah           | 1    | StrongRight   |
| StrongRight | Right       | nah          | mittel        | 1    | StrongRight   |
| StrongRight | Right       | nah          | weit          | 1    | StrongRight   |
| StrongRight | Right       | mittel       | nah           | 1    | StrongLeft    |
| StrongRight | Right       | mittel       | mittel        | 1    | StrongRight   |
| StrongRight | Right       | mittel       | weit          | 1    | StrongRight   |
| StrongRight | Right       | weit         | nah           | 1    | StrongLeft    |
| StrongRight | Right       | weit         | mittel        | 1    | StrongRight   |
| StrongRight | Right       | weit         | weit          | 1    | StrongRight   |

## Speed

Auf den folgenden Seiten sind die Regeln des Speed-Blockes abgebildet:

| WENN       |              |              |               | DANN |            |
|------------|--------------|--------------|---------------|------|------------|
| Angle      | CurrentSpeed | LeftDistance | RightDistance | DoS  | SpeedDelta |
| StrongLeft | low          | nah          | nah           | 1    | Accelerate |
| StrongLeft | low          | nah          | mittel        | 1    | Break      |
| StrongLeft | low          | nah          | weit          | 1    | Break      |
| StrongLeft | low          | mittel       | nah           | 1    | Break      |
| StrongLeft | low          | mittel       | mittel        | 1    | Maintain   |
| StrongLeft | low          | mittel       | weit          | 1    | Maintain   |
| StrongLeft | low          | weit         | nah           | 1    | Break      |
| StrongLeft | low          | weit         | mittel        | 1    | Maintain   |
| StrongLeft | low          | weit         | weit          | 1    | Maintain   |
| StrongLeft | mid          | nah          | nah           | 1    | Break      |
| StrongLeft | mid          | nah          | mittel        | 1    | Break      |
| StrongLeft | mid          | nah          | weit          | 1    | Break      |



| WENN       |              |              |               | DANN |            |
|------------|--------------|--------------|---------------|------|------------|
| Angle      | CurrentSpeed | LeftDistance | RightDistance | DoS  | SpeedDelta |
| StrongLeft | mid          | mittel       | nah           | 1    | Break      |
| StrongLeft | mid          | mittel       | mittel        | 1    | Break      |
| StrongLeft | mid          | mittel       | weit          | 1    | Break      |
| StrongLeft | mid          | weit         | nah           | 1    | Break      |
| StrongLeft | mid          | weit         | mittel        | 1    | Break      |
| StrongLeft | mid          | weit         | weit          | 1    | Maintain   |
| StrongLeft | high         | nah          | nah           | 1    | Break      |
| StrongLeft | high         | nah          | mittel        | 1    | Break      |
| StrongLeft | high         | nah          | weit          | 1    | Break      |
| StrongLeft | high         | mittel       | nah           | 1    | Break      |
| StrongLeft | high         | mittel       | mittel        | 1    | Break      |
| StrongLeft | high         | mittel       | weit          | 1    | Break      |
| StrongLeft | high         | weit         | nah           | 1    | Break      |
| StrongLeft | high         | weit         | mittel        | 1    | Break      |
| StrongLeft | high         | weit         | weit          | 1    | Break      |
| Left       | low          | nah          | nah           | 1    | Break      |
| Left       | low          | nah          | mittel        | 1    | Break      |
| Left       | low          | nah          | weit          | 1    | Break      |
| Left       | low          | mittel       | nah           | 1    | Break      |
| Left       | low          | mittel       | mittel        | 1    | Maintain   |
| Left       | low          | mittel       | weit          | 1    | Accelerate |
| Left       | low          | weit         | nah           | 1    | Break      |
| Left       | low          | weit         | mittel        | 1    | Accelerate |
| Left       | low          | weit         | weit          | 1    | Accelerate |
| Left       | mid          | nah          | nah           | 1    | Break      |
| Left       | mid          | nah          | mittel        | 1    | Break      |
| Left       | mid          | nah          | weit          | 1    | Break      |
| Left       | mid          | mittel       | nah           | 1    | Break      |
| Left       | mid          | mittel       | mittel        | 1    | Maintain   |
| Left       | mid          | mittel       | weit          | 1    | Maintain   |
| Left       | mid          | weit         | nah           | 1    | Break      |
| Left       | mid          | weit         | mittel        | 1    | Maintain   |
| Left       | mid          | weit         | weit          | 1    | Accelerate |
| Left       | high         | nah          | nah           | 1    | Break      |
| Left       | high         | nah          | mittel        | 1    | Break      |
| Left       | high         | nah          | weit          | 1    | Break      |
| Left       | high         | mittel       | nah           | 1    | Break      |
| Left       | high         | mittel       | mittel        | 1    | Break      |
| Left       | high         | mittel       | weit          | 1    | Break      |
| Left       | high         | weit         | nah           | 1    | Break      |
| Left       | high         | weit         | mittel        | 1    | Break      |
| Left       | high         | weit         | weit          | 1    | Maintain   |



| WENN  |              |              |               | DANN |            |
|-------|--------------|--------------|---------------|------|------------|
| Angle | CurrentSpeed | LeftDistance | RightDistance | DoS  | SpeedDelta |
| Front | low          | nah          | nah           | 1    | Break      |
| Front | low          | nah          | mittel        | 1    | Break      |
| Front | low          | nah          | weit          | 1    | Maintain   |
| Front | low          | mittel       | nah           | 1    | Break      |
| Front | low          | mittel       | mittel        | 1    | Accelerate |
| Front | low          | mittel       | weit          | 1    | Accelerate |
| Front | low          | weit         | nah           | 1    | Break      |
| Front | low          | weit         | mittel        | 1    | Accelerate |
| Front | low          | weit         | weit          | 1    | Accelerate |
| Front | mid          | nah          | nah           | 1    | Break      |
| Front | mid          | nah          | mittel        | 1    | Break      |
| Front | mid          | nah          | weit          | 1    | Break      |
| Front | mid          | mittel       | nah           | 1    | Break      |
| Front | mid          | mittel       | mittel        | 1    | Maintain   |
| Front | mid          | mittel       | weit          | 1    | Accelerate |
| Front | mid          | weit         | nah           | 1    | Break      |
| Front | mid          | weit         | mittel        | 1    | Accelerate |
| Front | mid          | weit         | weit          | 1    | Accelerate |
| Front | high         | nah          | nah           | 1    | Break      |
| Front | high         | nah          | mittel        | 1    | Break      |
| Front | high         | nah          | weit          | 1    | Break      |
| Front | high         | mittel       | nah           | 1    | Break      |
| Front | high         | mittel       | mittel        | 1    | Break      |
| Front | high         | mittel       | weit          | 1    | Maintain   |
| Front | high         | weit         | nah           | 1    | Break      |
| Front | high         | weit         | mittel        | 1    | Maintain   |
| Front | high         | weit         | weit          | 1    | Accelerate |
| Right | low          | nah          | nah           | 1    | Break      |
| Right | low          | nah          | mittel        | 1    | Break      |
| Right | low          | nah          | weit          | 1    | Break      |
| Right | low          | mittel       | nah           | 1    | Break      |
| Right | low          | mittel       | mittel        | 1    | Maintain   |
| Right | low          | mittel       | weit          | 1    | Accelerate |
| Right | low          | weit         | nah           | 1    | Break      |
| Right | low          | weit         | mittel        | 1    | Accelerate |
| Right | low          | weit         | weit          | 1    | Accelerate |
| Right | mid          | nah          | nah           | 1    | Break      |
| Right | mid          | nah          | mittel        | 1    | Break      |
| Right | mid          | nah          | weit          | 1    | Break      |
| Right | mid          | mittel       | nah           | 1    | Break      |
| Right | mid          | mittel       | mittel        | 1    | Maintain   |
| Right | mid          | mittel       | weit          | 1    | Maintain   |





| WENN        |              |              |               | DANN |            |
|-------------|--------------|--------------|---------------|------|------------|
| Angle       | CurrentSpeed | LeftDistance | RightDistance | DoS  | SpeedDelta |
| Right       | mid          | weit         | nah           | 1    | Break      |
| Right       | mid          | weit         | mittel        | 1    | Maintain   |
| Right       | mid          | weit         | weit          | 1    | Accelerate |
| Right       | high         | nah          | nah           | 1    | Break      |
| Right       | high         | nah          | mittel        | 1    | Break      |
| Right       | high         | nah          | weit          | 1    | Break      |
| Right       | high         | mittel       | nah           | 1    | Break      |
| Right       | high         | mittel       | mittel        | 1    | Break      |
| Right       | high         | mittel       | weit          | 1    | Break      |
| Right       | high         | weit         | nah           | 1    | Break      |
| Right       | high         | weit         | mittel        | 1    | Break      |
| Right       | high         | weit         | weit          | 1    | Maintain   |
| StrongRight | low          | nah          | nah           | 1    | Break      |
| StrongRight | low          | nah          | mittel        | 1    | Break      |
| StrongRight | low          | nah          | weit          | 1    | Break      |
| StrongRight | low          | mittel       | nah           | 1    | Break      |
| StrongRight | low          | mittel       | mittel        | 1    | Maintain   |
| StrongRight | low          | mittel       | weit          | 1    | Maintain   |
| StrongRight | low          | weit         | nah           | 1    | Break      |
| StrongRight | low          | weit         | mittel        | 1    | Maintain   |
| StrongRight | low          | weit         | weit          | 1    | Maintain   |
| StrongRight | mid          | nah          | nah           | 1    | Break      |
| StrongRight | mid          | nah          | mittel        | 1    | Break      |
| StrongRight | mid          | nah          | weit          | 1    | Break      |
| StrongRight | mid          | mittel       | nah           | 1    | Break      |
| StrongRight | mid          | mittel       | mittel        | 1    | Break      |
| StrongRight | mid          | mittel       | weit          | 1    | Break      |
| StrongRight | mid          | weit         | nah           | 1    | Break      |
| StrongRight | mid          | weit         | mittel        | 1    | Break      |
| StrongRight | mid          | weit         | weit          | 1    | Maintain   |
| StrongRight | high         | nah          | nah           | 1    | Break      |
| StrongRight | high         | nah          | mittel        | 1    | Break      |
| StrongRight | high         | nah          | weit          | 1    | Break      |
| StrongRight | high         | mittel       | nah           | 1    | Break      |
| StrongRight | high         | mittel       | mittel        | 1    | Break      |
| StrongRight | high         | mittel       | weit          | 1    | Break      |
| StrongRight | high         | weit         | nah           | 1    | Break      |
| StrongRight | high         | weit         | mittel        | 1    | Break      |
| StrongRight | high         | weit         | weit          | 1    | Break      |





# C Sitzungsprotokolle

## C.1 03.04.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### Themen

- Startmeeting
- Allgemeine Besprechung

### Diskussionen, Anträge, Beschlüsse

- Aufgabenstellung
- Anforderung und Bewertung
  - Nach HSR-Vorgabe
- ca. 5-6 Wochen Einarbeitungszeit
- Zeiterfassung
  - Mit Soll- und Ist-Zeit
- Wöchentliches Meeting vorgesehen
- Überblick verschaffen

**Nächste Besprechung** Dienstag, 11. 04. 2006, 13:00 Uhr

## C.2 11.04.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### Themen



- Zeitplan
- Allgemeine Besprechung

### **Diskussionen, Anträge, Beschlüsse**

- Das Vorgehen wird bewertet, wichtig ist nicht dass das Fahrzeug am schluss fährt sondern dass alle Requirements erfüllt sind. d.h:
  - Genaue Requirements
  - So einarbeiten das Aufwandabschätzung möglich ist
- Bücher besprochen
- wie KNN's funktionieren ist Stand der Technik («State of the art»)
  - Nur eine kurze allgemeine Beschreibung nötig
  - Die eigene Realisierung muss genau beschrieben sein
- Auf nächstes mal:
  - Projektplan fertig
  - Weiterhin vertiefen
  - Protokoll mitnehmen
- vielleicht Software anschauen.

**Nächste Besprechung** Dienstag, 18.04.2006, 13:00 Uhr

## **C.3 18.04.2006**

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### **Themen**

- Projektplan
- Einarbeiten
- Software

### **Diskussionen, Anträge, Beschlüsse**

- Die Protokolle der ersten beiden Meetings wurden (mit kleinen Änderungen) genehmigt.



- Der Projektplan wurde vorgelegt und nach kurzer Einsicht durch Herrn Kurka als vollständig («alle wichtigen Punkte sind vorhanden») bezeichnet
  - Auch der Detailzeitplan ist in Ordnung
- Es wurden eine Software zur Erstellung/Simulation neuronaler Netze und eine für Fuzzy Systeme angeschaut.
  - Von der Software für die neuronalen Netze haben wir eine Kopie (Image) bekommen
  - Von der Software für Fuzzy Systeme erhielten wir das Image sowie einen Lizenz-Dongle
- Weiter wurde auf die Homepage von Prof. Jürgen Sauer von der FH Regensburg verwiesen
  - Java Bibliothek für KNN's
  - Studien- und Diplomarbeiten
- Auf nächstes Mal:
  - Java Bibliotheken anschauen
  - Weiter einarbeiten (Kohonen/Hopfield/Fuzzy)
  - ev. GUI-Konzept
- Beim weiteren Vorgehen ist darauf zu achten, das ganze Gebiet möglichst weitgefächert unter die Lupe zu nehmen. Neben dem eigentlichen Einarbeiten soll das Ziel nicht aus den Augen verloren werden; es sollen auch bereits Gedanken zum Schlussprodukt angestellt werden.

**Nächste Besprechung** Dienstag, 25. 04. 2006, 13:00 Uhr

## C.4 25.04.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### Themen

- Einarbeiten
- Requirements
- GUI
- Diplomarbeit

### Diskussionen, Anträge, Beschlüsse

- Erste GUI Entwürfe besprochen mit Skizze



- Requirements
  - In Anforderungsspezifikation festlegen was und wie getestet wird.
- Hopfield: erstes Testprogramm
- Einarbeiten:
  - Hopfield
  - Kohonen
  - Fuzzy-Logik eher weniger, da voraussichtlich während der Studienarbeit nicht zum Einsatz kommt
- Diplomarbeit:
  - Eventuell Umbau eines ferngesteuerten Spielzeugautos
- Auf nächstes Mal:
  - Grobe Requirements
  - Use Cases
  - Weiter Einarbeiten
  - Hopfield Testimplementation

**Nächste Besprechung** Dienstag, 02. 05. 2006, 13:00 Uhr

## C.5 02.05.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### Themen

- Hopfield
- Requirements

### Diskussionen, Anträge, Beschlüsse

- Erster Entwurf der Anforderungsspezifikationen (Use Cases) besprochen
  - Soll- und Ist-Weg darstellen
  - Mathematisches Modell des Fahrzeuges
    - Gewicht
    - Beschleunigung
    - Ausmasse



- Performance
  - Hardwareanforderungen
- Allgemeine Spezifikationen
  - Betriebssystem
  - Software
  - Genaue Versionsnummern angeben
- Dokumenthistory
  - Grobänderungen auf Deckblatt
  - Genaue Änderungsdetails im Anhang
- Auf nächstes Mal:
  - Requirements fertigstellen (letzter Entwurf)
  - ev. Kohonen Testimplementation

**Nächste Besprechung** Dienstag, 09.05.2006, 13:00 Uhr

## C.6 09.05.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### Themen

- Anforderungsspezifikation
- Analyse

### Diskussionen, Anträge, Beschlüsse

- Entwurf der Anforderungsspezifikationen besprochen
  - Vom Umfang her in Ordnung, kleine Details noch korrigieren
    - KNN-Training -> Was tun, wenn KNN nicht konvergiert?
    - Möglichkeit, das Netz zurückzusetzen
    - Benutzer benachrichtigen
    - Falls Hopfield Energieminimum nicht erreicht
      - Use Cases anpassen (diesen Fall berücksichtigen)
      - Abweichungen grafisch darstellen



- Performance
- Performance nicht festlegen
  - Dafür Hard- und Softwarekonfiguration genau angeben
- Analyse nach Larman
- Auf nächstes Mal:
  - Analyse weiterarbeiten
  - Anforderungsspezifikation überarbeiten
  - Fuzzy-Einführung

**Nächste Besprechung** Dienstag, 16. 05. 2006, 13:00 Uhr

## C.7 16.05.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### Themen

- Anforderungsspezifikation
- Analysedokument
- Fuzzy Logik

### Diskussionen, Anträge, Beschlüsse

- Anforderungsspezifikation:
  - Genauer festlegen was «Ausgangskonfiguration» heisst
  - Letzte funktionierende Netzwerkkonfiguration
- Analyse
  - Domainmodel -> Limite bei KNN
- Fuzzy Logik
  - Kurze Einführung
  - Beispiel angeschaut in Fuzzy Tech
- Auf nächstes Mal:
  - Design Problem Domain

**Nächste Besprechung** Dienstag, 23. 05. 2006, 13:00 Uhr





## C.8 23.05.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### Themen

- Design
- Fuzzy Logik

### Diskussionen, Anträge, Beschlüsse

- Design kurz angeschaut
  - soweit ok.
- Fuzzy Logik
  - Neuer Fahrzeugsensor (drehender Distanzmesser)
  - Fahrzeugregelung besprochen
  - Fuzzy-Konfiguration erst machen, nachdem GUI implementiert wurde
    - Ermöglicht visuelle Überprüfung des Verhaltens des Fahrzeuges
- Auf nächstes Mal:
  - Weiterarbeiten Design
  - Implementierung fortsetzen

**Nächste Besprechung** Dienstag, 30.05.2006, 13:00 Uhr

## C.9 30.05.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### Themen

- Design
- Implementierung



### **Diskussionen, Anträge, Beschlüsse**

- Aktuellen Stand besprochen
  - Liegen gut im Zeitplan
- Auf nächstes Mal:
  - Implementierung fortsetzen

**Nächste Besprechung** Dienstag, 06. 06. 2006, 13:00 Uhr

## **C.10 06.06.2006**

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### **Themen**

- Implementierung
- Fuzzy-Sets

### **Diskussionen, Anträge, Beschlüsse**

- Aktuellen Stand besprochen
  - Liegen gut im Zeitplan
- Fuzzy-Sets angeschaut
  - Schauen, dass gesamter Wertebereich abgedeckt ist
  - Bei Winkel-Input ev. 5 Unterscheidungen machen
- Auf nächstes Mal:
  - Implementierung fortsetzen

**Nächste Besprechung** Dienstag, 13. 06. 2006, 13:00 Uhr

## **C.11 13.06.2006**

**Teilnehmer** Herr Kurka, Fehlmann, Steffen



## Themen

- Implementierung

## Diskussionen, Anträge, Beschlüsse

- Aktuellen Stand besprochen
  - Liegen gut im Zeitplan
- Applikation
  - nimmt langsam Form an
  - hat noch Fehler
    - Hopfield Netz -> Fahrzeug hängt manchmal fest
    - Sensor -> Noch zu wenig schnell
    - Performance allgemein noch nicht optimal
- Auf nächstes Mal:
  - Implementierung fortsetzen

**Nächste Besprechung** Dienstag, 20.06.2006, 13:00 Uhr

## C.12 20.06.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

## Themen

- Implementierung
- Präsentation
- Tests

## Diskussionen, Anträge, Beschlüsse

- Aktuellen Stand besprochen
- Applikation
  - Frontsensor erstellt, welcher verhindern soll, dass Fahrzeug während Netzberechnung in ein Hindernis fährt



- Sensor hat noch Fehler, Fahrzeug bleibt bei Hindernis stehen, auch wenn es bereits abgedreht hat
- Präsentation
  - Am 4. Juli, 13 Uhr
  - Am Arbeitsplatz
  - Dauer 20 Minuten
  - Wird bewertet
  - Zielpublikum: Informatik-/Elektroingenieur
- Auf nächstes Mal:
  - Implementierung/Tests fortsetzen

**Nächste Besprechung** Dienstag, 27.06.2006, 13:00 Uhr

## C.13 27.06.2006

**Teilnehmer** Herr Kurka, Fehlmann, Steffen

### Themen

- Implementierung
- Abgabe
- Demo des aktuellen Standes

### Diskussionen, Anträge, Beschlüsse

- Aktuellen Stand besprochen
  - Fahrzeug fährt, findet Ziel, Kollisionen sind selten
  - Noch kleines Problem mit Rechtskurven
- Präsentation
  - Vielleicht Raum reservieren
  - Weitere Zuschauer zugelassen
- Abgabe
  - Alles auf CD, inkl. Entwicklungsumgebung
  - Dokumentieren, wie Software weiterentwickelt werden kann
    - Zum Beispiel Anpassungen an Fuzzy-Logik
  - 2 Ausdrücke der Dokumentation



- Abgabe am Freitag Abend
- Auf nächstes Mal:
  - Arbeit abschliessen
  - Präsentation vorbereiten

**Nächste Besprechung** Dienstag, 04. 06. 2006, 13:00 Uhr





## D Verwendete Software

| <b>Name:</b>                      | <b>Version:</b> |
|-----------------------------------|-----------------|
| DokuWiki                          | 2005-09-22      |
| Eclipse                           | 3.1.2           |
| fuzzyTECH Professional            | 5.50b           |
| IzPack-Installer                  | 3.8.1           |
| Java                              | 1.5.0-07        |
| JUDE Community                    | 2.5.1           |
| Microsoft Visio                   | 2003 SP2        |
| Microsoft Windows XP Professional | SP2             |
| MiKTeX $\LaTeX$                   | 2.4             |
| SubVersion                        | 1.3.2           |
| TeXnicCenter                      | 1 Beta 7.01     |
| Tortoise Svn                      | 1.3.5           |

Tabelle D.1: Software







# Glossar

|                                 |  |
|---------------------------------|--|
| A.N.N.A                         | Artificial Neural Network Application  |
| L <sup>A</sup> T <sub>E</sub> X | Bei L <sup>A</sup> T <sub>E</sub> X handelt es sich um ein spezielles Format für T <sub>E</sub> X. Unter einem Format versteht man in diesem Zusammenhang eine bestimmte Anzahl vordefinierter Kommandos, die beim Starten von T <sub>E</sub> X zur Verfügung stehen. Der Sinn dabei ist der, T <sub>E</sub> X so flexibel wie möglich zu halten. Dazu wurden nur sehr wenige Kommandos fest eingebaut, der Rest wird als Format dazu geladen. |
| T <sub>E</sub> X                | Technisch gesehen handelt es sich bei TeX um einen Interpreter, der ca. 300 fest eingebaute Befehle kennt und einen komplexen Mechanismus zur Definition eigener Makros bereitstellt.  |
| Fuzzy-Logik                     | Fuzzy-Logik (englisch: fuzzy = ungenau, verschwommen, unscharf) ist eine Theorie, eine Verallgemeinerung der zweiwertigen Booleschen Logik, die vor allem für die Darstellung menschlichen Wissens und menschlicher Überlegung zur Verarbeitung in Computern entwickelt wurde.   |
| JUDE                            | UML Modeling Tool - [JUD04]  |
| KNN                             | <b>K</b> ünstliche <b>n</b> euronale <b>N</b> etze (engl. artificial neural networks - ANNs) sind ein Zweig der künstlichen Intelligenz und prinzipieller Forschungsgegenstand der Neuroinformatik. Man stellt KNNs den natürlichen neuronalen Netzen gegenüber, welche Nervenzellverschaltungen im Gehirn modellieren. Bücher: [Läm01], [Bra99]   |
| UML                             | Unified Modeling Language  |
| Wiki                            | Wikis, auch WikiWikis und WikiWebs genannt, sind im World Wide Web verfügbare Seitensammlungen, die von den Benutzern nicht nur gelesen, sondern auch online geändert werden. Sie ähneln damit Content Management Systemen. Der Name stammt von wiki-wiki, dem hawaiianischen Wort für 'schnell'. Wie bei Hypertexten  |



üblich, sind die einzelnen Seiten und Artikel eines Wikis durch Querverweise (Links) miteinander verbunden. Die Seiten lassen sich jedoch sofort am Bildschirm ändern.

<http://de.wikipedia.org/wiki/Wiki>



## Literaturverzeichnis

- [Bra99] BRAUSE, Rüdiger: *Neuronale Netze*. Teubner Verlag, 1999
- [Fou05] FOUNDATION, The E. *Eclipse.org*. <http://www.eclipse.org/>. 2005
- [Hel02] HELMUT, Kopka: *L<sup>A</sup>T<sub>E</sub>X Band 1: Einführung*. 3., überarbeitete Auflage. ADDISON-WESLEY, 2002
- [JUD04] JUDE - UML Modeling Tool. <http://objectclub.esm.co.jp/Jude/>. 2004
- [Koh04] KOHM, Markus. *Das KOMA-Script-Paket*. <http://www.komascript.de/>. 2004
- [Lar04] LARMAN, Craig: *Applying UML and Patterns*. Third Edition. Prentice Hall PTR, 2004
- [Läm01] LÄMMEL, Uwe: *Künstliche Intelligenz*. Fabhbuchverlag Leibzig, 2001