

# Autonome Fehlererkennung bei Robotern mithilfe eines künstlichen Immunsystems

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Software Engineering & Internet Computing**

eingereicht von

**Martin Wieser**

Matrikelnummer 0727131

an der

Fakultät für Informatik der Technischen Universität Wien

Betreuung

Betreuung: Univ.Prof. Dipl.-Inf. Dr.rer.nat. Jens Knoop

Mitwirkung: Projektass. Dipl.-Ing.(FH) Dr.techn. Dietmar Schreiner

Wien, 3. September 2012 \_\_\_\_\_

(Unterschrift Verfasser)

\_\_\_\_\_  
(Unterschrift Betreuung)



# Erklärung zur Verfassung der Arbeit

Martin Wieser

Hohlweggasse 28, 1030 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

---

(Ort, Datum)

---

(Unterschrift Verfasser)



# Danksagung

Diese Arbeit bietet mir nicht nur die Möglichkeit, meine Arbeit des letzten Jahres zu dokumentieren, sondern eröffnet mir vielmehr die Gelegenheit, den Menschen zu danken, die zum Erfolg dieser Arbeit beigetragen haben.

Ich möchte meine Diplomarbeit meinen Eltern Maria und Helmut Wieser widmen, die mir das Studium ermöglichten und stets großes Interesse für meine Arbeit zeigten.

Ein besonderer Dank gilt meinen Betreuern Univ.Prof. Dr. Jens Knoop und Dr. Dietmar Schreiner, die mit guten Ideen und sehr viel Engagement meine Diplomarbeit betreut haben.

Schließlich gilt mein Dank noch Evelyn Gamper und meiner Freundin Katharina Seeber, die sich als Lektoren zur Verfügung stellten, und allen Freunden und Bekannten für ihre moralische Unterstützung.



# Abstract

This thesis investigates whether errors can be detected in a robot when active, using an artificial immune system. The use of robots on the moon, in radioactively contaminated areas, or even on the field of the RoboCup often doesn't allow manual maintenance of the system by a human. The independently operating system must detect faults autonomously in many situations to react whilst the robot is active. The error handling of the most probable errors by exceptions is often not sufficient. Due to this approaches of the discipline of machine learning are used to allow the robot to detect anomalies in an autonomous way. The artificial immune system represents one of these approaches, which allows to distinguish error-free and dangerous state patterns. The biological immune system serves as a strong source of inspiration for the algorithms and their components.

The intention of this thesis is to develop a robotic system using an artificial immune system, which identifies errors of defined classes during the robots activity and clearly distinguishes them from error-free states. The response time of the immune system and the number of mispriced states are the chosen criteria of evaluation. Throughout this work it is shown that an artificial immune system integrated into a soccer-playing robot NAO, is able to diagnose the failure of visual and motor skills.

The implemented immune system is integrated in the infrastructure of the "Austrian Kangaroos" for the purpose of evaluation. Some evaluation experiments are performed on the simulator in order to limit the wear on the hardware. Finally, the infrastructure is tested on the robot itself in the course of a soccer game. The robot is capable of detecting 99,5% of all errors within a few seconds. In addition, the immune system estimates more than 99% of the error-free states correctly. During the Robocup 2012 was the implemented system already successfully presented.



# Kurzfassung

Diese Arbeit untersucht, ob fehlerhaftes Verhalten eines Roboters zur Laufzeit mithilfe eines künstlichen Immunsystems erkannt werden kann. Der Einsatz von Robotern auf dem Mond, in radioaktiv verseuchten Gebieten oder auch auf dem Spielfeld des Robocup erlaubt häufig kein Eingreifen eines Menschen. Das selbstständig agierende System muss in vielen Situationen autonom Fehlerfälle erfassen, um darauf im laufenden Betrieb reagieren zu können. Die Abdeckung deterministisch auftretender Fehler durch Ausnahmebedingungen ist häufig nicht ausreichend. Aufgrund dessen werden Ansätze aus der Disziplin des maschinellen Lernens eingesetzt, um den Roboter die Anomalien autark erkennen zu lassen. Künstliche Immunsysteme stellen einen dieser Ansätze dar, die erlauben, fehlerfreie und gefährliche Zustandsmuster zu unterscheiden. Biologische Immunsysteme sind dabei eine Inspirationsquelle für die Algorithmen und deren Bestandteile.

Ziel dieser Diplomarbeit ist es, ein Robotersystem mittels künstlichen Immunsystems in die Lage zu versetzen, sowohl deterministisch als auch indeterministisch auftretende Fehler mit unbekanntem Quellen zur Laufzeit zu registrieren und indessen deutlich von fehlerfreien Zuständen zu unterscheiden. Dabei dienen die Reaktionszeit des Immunsystems sowie die Anzahl der falsch eingeschätzten Zustände als Evaluierungskriterien. In der Arbeit wird gezeigt, dass ein künstliches Immunsystem integriert in einen Fußball spielenden NAO-Roboter die Fähigkeit besitzt, den Ausfall von visuellen sowie motorischen Fähigkeiten zu diagnostizieren.

Das implementierte Immunsystem wird zum Zweck der Evaluierung in die Infrastruktur der "Austrian Kangaroos" integriert. Ein Großteil der Evaluierung erfolgt am Simulator, um den Verschleiß der Hardware einzuschränken. Die erstellte Infrastruktur wird abschließend am Roboter selbst im Spielbetrieb getestet. Der Roboter ist in der Lage, sowohl am Simulator als auch in der Realität 99,5% aller Fehler innerhalb von wenigen Sekunden zu erkennen. Zudem schätzt das künstliche Immunsystem mehr als 99% der fehlerfreien Zustände korrekt ein. Im Rahmen des Robocup 2012 konnte das implementierte System bereits erfolgreich vorgestellt werden.



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>xi</b>
<b>Tabellenverzeichnis</b>	<b>xii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Beiträge . . . . .	3
1.3 Gliederung . . . . .	4
<b>2 Implementierungsumgebung</b>	<b>5</b>
2.1 NAO . . . . .	6
2.2 Framework . . . . .	7
2.3 Fußballspieler . . . . .	9
<b>3 Biologisches Immunsystem</b>	<b>11</b>
3.1 Grundlagen des Immunsystems . . . . .	11
3.2 Das angeborene Immunsystem . . . . .	14
3.3 Das adaptive Immunsystem . . . . .	15
3.4 Die Arbeitsweise des Immunsystems . . . . .	16
3.5 Gefahrentheorie . . . . .	18
3.6 Dendritische Zelle . . . . .	20
3.7 Zusammenfassung . . . . .	21
<b>4 Künstliche Immunsysteme</b>	<b>23</b>
4.1 Modelle der Immunologie . . . . .	23
4.2 Konzept der negativen Selektion . . . . .	26
4.3 Konzept der klonalen Selektion . . . . .	27

4.4	Konzept idiotopischer Netzwerke . . . . .	28
4.5	Konzept der Gefahrentheorie . . . . .	29
4.6	Zukünftige Herausforderungen . . . . .	30
<b>5</b>	<b>Dendritischer Zell-Algorithmus</b>	<b>33</b>
5.1	Idee . . . . .	34
5.2	Abstraktion . . . . .	35
5.3	Algorithmus . . . . .	40
<b>6</b>	<b>Vorbereitung und Umsetzung</b>	<b>45</b>
6.1	Fehlerklassen . . . . .	46
6.2	Daten aus dem Robotersystem . . . . .	48
6.3	Migration des Algorithmus . . . . .	53
6.4	Antigene und Signale . . . . .	57
<b>7</b>	<b>Evaluierung</b>	<b>63</b>
7.1	Simulationsexperimente . . . . .	65
7.2	Einsatztest . . . . .	87
7.3	Fazit . . . . .	90
<b>8</b>	<b>Schlussfolgerung und Ausblick</b>	<b>91</b>
8.1	Schlussfolgerung . . . . .	91
	<b>Literaturverzeichnis</b>	<b>99</b>

# Abbildungsverzeichnis

2.1	Roboter NAO von Aldebaran . . . . .	6
2.2	Aufbau des Frameworks [29] . . . . .	8
2.3	Simulator [29] . . . . .	9
2.4	Verhalten des Roboters . . . . .	10
3.1	Rezeptoren und Antigene <sup>1</sup> . . . . .	12
3.2	Einteilung der Bestandteile des Immunsystems <sup>2</sup> . . . . .	13
3.3	Aktivierung von T-Zelle und B-Zellen durch Antigen-präsentierende Immunzellen .	17
3.4	Gefahrenmodell (danger model) . . . . .	19
3.5	Unreife dendritische Zellen, halb-reife dendritische Zellen und reife dendritische Zellen [12] . . . . .	21
4.1	Geschichte der künstlichen Immunsysteme [16] . . . . .	24
4.2	Prinzip der negativen Selektion [11] . . . . .	27
4.3	Mutation eines Datensatzes . . . . .	28
5.1	Verlauf der Abstraktion in [13] . . . . .	35
5.2	Modell der Transformation von dendritischen Zellen und der Einfluss entsprechender Signale [13] . . . . .	37
5.3	Aktivitätsdiagramm der Entwicklung einer dendritischen Zelle [13] . . . . .	38
6.1	Codeauszug aus Xabslbeispiel . . . . .	50
6.2	Struktur einer aktiven Option . . . . .	51
6.3	Struktur der Symbole . . . . .	52
6.4	Aufbau der Algorithmus-Bibliothek . . . . .	54
6.5	Schnittstellen für Antigene und Signale aus Nao . . . . .	55
6.6	Verbindung zwischen dem Roboter und dem künstlichem Immunsystem . . . . .	57

7.1	Experimente im Simulator . . . . .	64
7.2	Verlauf mit Positionsänderungsfehler (12 Sekunden) . . . . .	71
7.3	Vergleich ausgewählter Antigene für die Positionsänderungsfehlerklasse . . . . .	72
7.4	Verlauf mit Fehler bei visueller Suche (12 Sekunden) . . . . .	78
7.5	Vergleich ausgewählter Antigene für die Fehlerklasse der visuellen Suche . . . . .	79
7.6	Verlauf ohne Fehler . . . . .	82
7.7	Verlauf mit beiden Fehlerklassen . . . . .	84
7.8	Roboter mit Behinderung . . . . .	89
7.9	Visuelle Unterstützung der Präsentation - grün: Anzahl der gesunden Zellen, rot: Anzahl der gefährlichen Zellen, blau: Summe der gereiften Zellen . . . . .	89

## Tabellenverzeichnis

5.1	Abstraktion der Signale . . . . .	39
7.1	Ergebnisse ohne Fehler bei Positionsänderung . . . . .	66
7.2	Ergebnisse mit Fehler bei Positionsänderung (60 Sekunden) . . . . .	67
7.3	Ergebnisse mit Fehler bei Positionsänderung (6 Sekunden) . . . . .	70
7.4	Ergebnisse ohne Fehler bei visueller Suche . . . . .	74
7.5	Ergebnisse mit Fehler bei visueller Suche (60 Sekunden) . . . . .	75
7.6	Ergebnisse mit Fehler bei visueller Suche (6 Sekunden) . . . . .	77
7.7	Ergebnisse mit Fehler bei visueller Suche und verringerter Toleranzzeit . . . . .	80

# Einleitung

## 1.1 Motivation

Software wird immer komplexer und umfangreicher, dadurch wird die Fehlerbehandlung stetig aufwändiger und teurer. Bereits die Fehlererkennung stellt in vielen Systemen ein enormes Problem dar. In einigen Disziplinen, wie beispielsweise in der Robotik, ist es der Fall, dass Subsysteme aus verschiedensten Bereichen aufeinandertreffen, wodurch der Aufwand um ein Vielfaches größer wird. Dies gilt auch beim NAO Roboter der Standard Plattform League im Robocup. Fähigkeiten zur Planung, Bildverarbeitung, motorischen Bewegung, künstlichen Intelligenz sowie Zusammenarbeit werden benötigt, um ein Team zum Erfolg zu führen. Durch die enge Kollaboration der einzelnen Systeme untereinander ist es nur sehr schwer möglich, einzelne Defekte oder Anomalien zu erfassen oder einzugrenzen. Insbesondere kann es zu Fehlverhalten kommen, obwohl Hard- und Software fehlerfrei sind. Ein Beispiel dafür ist ein Lichtausfall am Spielfeld. Erschwerend kommt hinzu, dass Roboter häufig außerhalb der Erreichbarkeit des Menschen liegen.

Im März 2002 schoss die Europäische Raumfahrtorganisation Esa den größten europäischen Erdbeobachtungssatelliten "Envisat" in die Umlaufbahn, welcher daraufhin die Erde 52.000 Mal umrundet hat, bis es zu einem Totalausfall der Kommunikationssysteme gekommen ist und alle Rettungsversuche scheiterten. Die Ursache für diesen Fehlerfall ist unklar und wird auch nie geklärt werden können, da sich der Satellit weiter als Weltraumschrott in der Umlaufbahn be-

wegt.<sup>1</sup> Vergleichbare Probleme können auch bei Robotern zur Bombenräumung, in radioaktiv verseuchten Gebieten oder auch beim NAO-Roboter, welcher gerade ein Fußballspiel bestreitet, auftreten. Da hierbei kein menschliches Eingreifen möglich ist, muss das System selbstständig den Fehlerfall erkennen und entsprechend darauf reagieren.

Mit der Erkennung von Fehlern beschäftigt sich die Informatik bereits seit Jahrzehnten, trotzdem gibt es nur ansatzweise systematische Verfahren zur Fehlerdetektion. In vielen Fällen werden deterministisch auftretende Fehler durch Ausnahmebedingungen behandelt, was bei größeren Systemen sehr schnell komplex wird und keine Garantie dafür bietet, dass alle Eventualitäten bedacht werden. Dabei liegt ein geeigneter Ansatz zur Lösung in jedem von uns: das Immunsystem. Durch die Erforschung der biologischen Grundlagen und deren Übertragung in die virtuelle Welt erhalten Softwaresysteme eine ähnliche Verteidigung gegen Anomalien. Versuche im Bereich der Netzwerküberwachung haben erfolgsversprechende Ergebnisse gebracht, woraus die Idee entsteht, diese Erkenntnisse auf andere Bereiche zu übertragen.

Künstliche Immunsysteme stellen heute ein breit gefächertes Studiengebiet im Bereich der *bio-inspired computation* dar. Daraus resultiert eine Vielzahl an unterschiedlichsten Algorithmen, welche ähnlich dem biologischen Vorbild diverse Probleme lösen. Die Verwendung und Weiterentwicklung künstlicher Immunsysteme konzentriert sich zurzeit auf zwei Generationen. Die Erste ist dabei stark vom theoretischen Immunologie-Modell beeinflusst, wobei häufig ebenso Parallelen zum Maschinernen gezogen werden können. Die Ähnlichkeit mit anderen maschinellen Lernverfahren ist auch bei den zwei wichtigsten Techniken - klonale und negative Selektion - zu erkennen. Im Gegensatz dazu wird bei der zweiten Generation der künstlichen Immunsysteme auf eine interdisziplinäre Methodik gesetzt. Diese erst kürzlich veröffentlichten Ansätze, wie beispielsweise der dendritische Zell-Algorithmus [12], bergen großes Potenzial in sich [4] [26].

Aus diesen Ansätzen ergibt sich die Fragestellung dieser Arbeit:

- Ist eine angepasste Form des dendritischen Zell-Algorithmus in der Lage eine Echtzeit-Erkennung von Fehlern in einem Robotersystem durchzuführen?

Sollte diese Frage entsprechend der Annahme positiv beantwortet werden, muss das entwickelte System auf seine Praktikabilität hin untersucht werden. Dabei ergeben sich folgende weiterge-

---

<sup>1</sup><http://www.zeit.de/wissen/2012-05/envisat-satellit-ende> - zuletzt besucht: 29.05.2012

hende Fragen:

- Ist das System in der Lage mehrere Fehlerklassen parallel zu behandeln?
- Kann das System in praktischer Sicht Fehler in akzeptabler Zeit und Genauigkeit erkennen?

Die Durchführung sowie die Analyse finden auf der Software des Robocupteams Austrian Kangaroos statt. Dabei können Daten einerseits anhand vorhandener Simulationen und andererseits direkt durch Versuche am Roboter erfasst werden. Zunächst ist eine detaillierte Untersuchung des vorhandenen Frameworks durchzuführen. Aufbauend darauf kann eine Strategie zur Anbindung des künstlichen Immunsystems erarbeitet werden. Die gewonnenen Erkenntnisse ermöglichen darauffolgend eine Anpassung des ausgewählten Algorithmus und dessen Implementierung. Nach den ersten Ergebnissen durch die Experimente im Simulator wird das Immunsystem zur weiteren Evaluierung in den Roboter integriert und getestet.

## 1.2 Beiträge

Aufbauend auf Forschungsergebnissen im Bereich der Fehlererkennung in Robotersystemen wird in dieser Arbeit mit einem künstlichen Immunsystem ein neuartiger Ansatz für dieses Problem vorgestellt, der auf dem dendritischen Zell-Algorithmus von Greensmith beruht [12]. Die vorrangigen Beiträge dabei sind:

- Anpassung des dendritischen Zell-Algorithmus an die Fehlererkennung
- Implementierung der Schnittstellen zwischen dem Robotersystem und dem Fehlererkennungsalgorithmus
- Optimierung der Algorithmusparameter

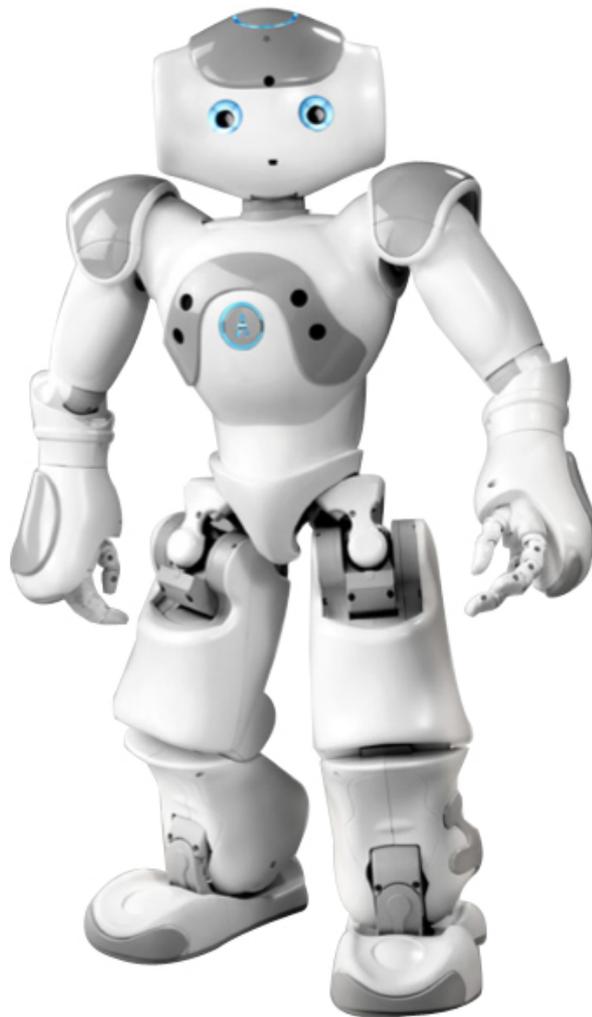
## 1.3 Gliederung

- **Kapitel 1** gibt einen Überblick über die Arbeit und führt mit wenigen Worten zum Thema hin. Außerdem werden die Motivation und einige grundlegende Punkte des Werkes beschrieben.
- **Kapitel 2** stellt den Roboter NAO und das verwendete Framework vor.
- **Kapitel 3** befasst sich mit den biologischen Ansätzen dieser Arbeit. Anfangs werden Grundlagen der Immunologie angeführt, welche zu konkreten Theorien und Bestandteilen des verwendeten Algorithmus hinführen.
- **Kapitel 4** zeigt die historische Entwicklung der künstlichen Immunsysteme. Dabei werden konkrete Konzepte der negativen sowie klonalen Selektion behandelt. Ein besonderes Augenmerk liegt jedoch auf den Konzepten der Gefahrentheorie und den damit verbundenen zukünftigen Herausforderungen.
- **Kapitel 5** beschreibt detailliert den dendritischen Zell-Algorithmus. Die grundlegenden Hypothesen sowie Annahmen, welche im Algorithmus immer wieder auftauchen, werden aufgezeigt. Außerdem wird im Abschnitt Abstraktion die Verbindung zur Biologie erläutert, worauf eine ausführliche Beschreibung des Algorithmus folgt.
- **Kapitel 6** behandelt die Herangehensweise an die Problemstellung. Die Festlegung der zu untersuchenden Fehlerklassen bildet den Anfang des Kapitels. Die Lokalisierung einer geeigneten Datenschnittstelle im Roboter-Framework sowie die Beschreibung der Architektur und der Komponenten der erstellten Software bilden die Grundlage für die folgende Evaluierung. Des Weiteren werden die Integration des Immunsystems und die dafür benötigten Bestandteile behandelt.
- **Kapitel 7** umfasst die Ergebnisse der Versuche, eine ausführliche Analyse und daraus resultierende Erkenntnisse. Die ersten beiden Teile beinhalten Experimente zu je einer der beiden Fehlerklassen. Im dritten Abschnitt werden Erkenntnisse aus der Kombination der Fehlerklassen angeführt, worauf eine Analyse der Versuche am echten Roboter folgt.
- **Kapitel 8** versucht den Kreis zur Biologie zu schließen, indem einige Bestandteile des künstlichen Immunsystems mit ihren Namensgebern aus der Biologie verglichen werden. Eine Zusammenfassung und ein Ausblick auf weiterführende Schritte schließen die Arbeit ab.

## Implementierungsumgebung

Eine der wichtigsten Erfindungen unserer Zeit ist der Computer, dessen Erfolgsgeschichte bis heute beispiellos und von immer neuen Trends geprägt ist. Neben dem Streben nach dem allgegenwärtigen Computer geht heute die Entwicklung in Richtung der Vernetzung. Computer sind nicht länger einzelne Rechner, sondern ein Teil von größeren verteilten Systemen. Außerdem sind die Menschen bemüht, immer mehr Aufgaben an den Computer abzugeben. Zunächst sind dies vor allem repetitive Tätigkeiten gewesen, welche mit der Zeit durch immer komplexere abgelöst worden sind und mittlerweile einen gewissen Grad an Intelligenz voraussetzen.

Die Speerspitze in diesen Trends bilden oft die Roboter, welche die Möglichkeit zur Nachbildung von menschlichen Zügen darstellen. Besonders humanoide Roboter, wie sie in der Standard Platform League des Robocups verwendet werden, zeigen viele Merkmale von Menschen. In den folgenden Abschnitten wird anfangs der verwendete Roboter “NAO” vorgestellt. In einem zweiten Schritt fällt das Augenmerk auf das Innenleben des Roboters, die Software. Hierbei wird hauptsächlich der Bereich der künstlichen Intelligenz im Detail erläutert.



**Abbildung 2.1:** Roboter NAO von Aldebaran<sup>1</sup>

## 2.1 NAO

Im Jahr 2004 startete das französische Start-up-Unternehmen Aldebaran das Projekt NAO. Den großen Durchbruch schaffte der Roboter 2007, als NAO den bis dahin verwendeten Aibo von Sony in der Standard Platform League (SPL) des Robocups ersetzte. 2010 gingen erneut Bilder des humanoiden Roboters um die Welt, da am Rande der Shanghai Expo in China eine Grup-

---

<sup>1</sup><http://www.robotshop.com> zuletzt besucht 15.02.2012

pe von NAOs eine Reihe von Choreografien in perfekter Synchronität vorführten. Ende 2011 präsentierte Aldebaran schließlich die neueste Version dieser Roboterreihe.<sup>2</sup>

NAO ist ein 58 Zentimeter großer, programmierbarer autonomer Roboter. Die Maschine besitzt mehrere menschliche Merkmale, wie zwei Arme, zwei Beine, einen Rumpf und einen Kopf, was ihn zu einem sogenannten humanoiden Roboter macht. Das gesamte System besitzt ein Gewicht von etwas über vier Kilogramm, verfügt über 25 Freiheitsgrade in seinen Gelenken und beinhaltet unterschiedlichste Sensoren, wobei die zwei CMOS 640x480 Kameras die Augen des NAO darstellen. Auf einer x86 AMD Geode CPU mit 500 Megahertz läuft das linux-basierte Betriebssystem. Der Roboter kann über Wifi oder Ethernet gesteuert und programmiert werden. Dies erfolgt in den meisten Fällen in den Sprachen C, C++, Python, Urbi oder .Net.

Dieser hohe Grad an Kontrolle durch die volle Programmierbarkeit und das vergleichsweise einfache Arbeiten mit dem System macht den NAO vielseitig einsetzbar. Neben den Verwendungsmöglichkeiten in einem breiten Spektrum der Wissenschaft, wird dieser Roboter in der Bildung und zu Präsentationszwecken bei Veranstaltungen eingesetzt. Im Bereich der Wissenschaft wird das System zur Erforschung der künstlichen Intelligenz, Lokalisierung, Navigation, Struktur- und Bewegungsanalyse sowie in einigen weiteren Gebieten benutzt [3].

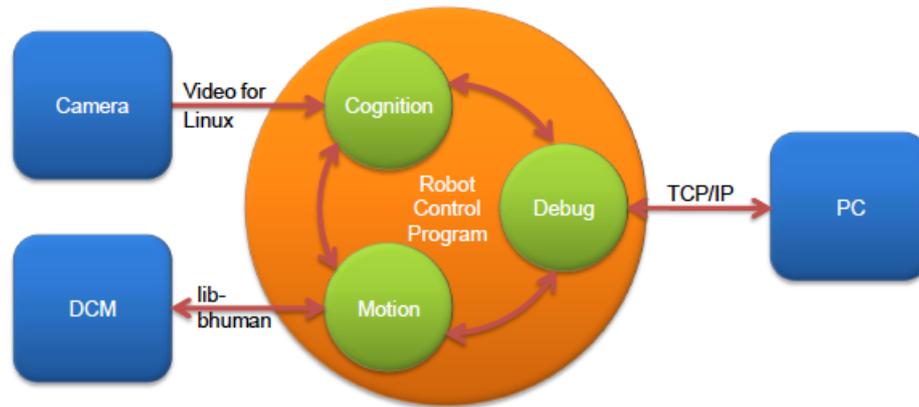
## 2.2 Framework

Die Programmierung und letztendlich die Steuerung des Roboters beruht auf einer sehr umfangreichen und äußerst komplexen Middleware. Wir bauen auf dem Framework vom Robocup Team BHuman auf, welches 2006 als Zusammenschluss der Universität Bremen und dem deutschen Forschungszentrum für künstliche Intelligenz (DFKI) gegründet wurde. Als Grundlage für das veröffentlichte System diente das Framework des ehemaligen GermanTeams 2007, das an die Anforderungen des NAO angepasst wurde. Neben der Vereinfachung der Programmierung der Roboter beeindruckt das System durch eine enorm mächtige Toolunterstützung, die die Arbeit der Entwickler vereinfacht. In den folgenden Abschnitten werden grob die fundamentalen Bausteine dieser Middleware erläutert. Eine detaillierte Beschreibung ist in [29] gegeben.

Die Grundlage für das Framework bildet die Anbindung an die Hardware, welche von BHuman teilweise neu implementiert wurde. Im Normalfall erfolgt die Kommunikation zur Hardware

---

<sup>2</sup>Die Beschreibung beziehen sich auf eine ältere Version, da diese für die weitere Arbeit verwendet wurde.



**Abbildung 2.2:** Aufbau des Frameworks [29]

mittels der NaoQi SDK-Bibliothek, welche vom Hersteller zur Verfügung gestellt wird. Einige Module dieses Codes werden durch das Framework deaktiviert und durch die neuen *libbhuman*-Komponenten ersetzt.

Aufbauend auf dieses Fundament wird der Roboter von zwei Prozessen gesteuert: dem Bewegungsprozess (Motion) und dem Wahrnehmungsprozess (Cognition). Vervollständigt wird das Framework durch den Debug-Prozess, welcher allerdings nicht direkt für die Handlungen des NAOs sorgt. In Abbildung 2.2 werden die Verbindungen der Abläufe untereinander sowie zu weiteren Komponenten dargestellt. Der Wahrnehmungsprozess verarbeitet neben den Daten der Kamera auch Informationen des Bewegungsprozesses. Dieser Ablauf ist über die *libbhuman*-Bibliothek mit den restlichen Sensoren und Motoren des NAOs verbunden. Der Debug-Prozess stellt die Daten der anderen Prozesse dem Benutzer am PC zur Verfügung und gibt dessen Eingaben an die Bewegung und Wahrnehmung weiter.

Die Interprozess-Kommunikation wird in der bereitgestellten Middleware anhand von Blackboards realisiert. Jeder Prozess besitzt einen dieser zentralen Speicher für Informationen, welche in der Dokumentation des Frameworks [29] als *Representation* bezeichnet werden. Die Prozesse stellen dabei eine Reihe von Referenzen auf *Representations* bereit, welche von anderen Prozessen bei Bedarf verwendet werden können.

Die bereits erwähnte Vereinfachung der Entwicklungsarbeit wird vor allem durch den SimRobot-Simulator (Abbildung 2.3) enorm gesteigert. Dieser bietet von den einzelnen Werten verschiedenster Sensoren über die Zustände der künstlichen Intelligenz bis hin zur grafischen Darstellung

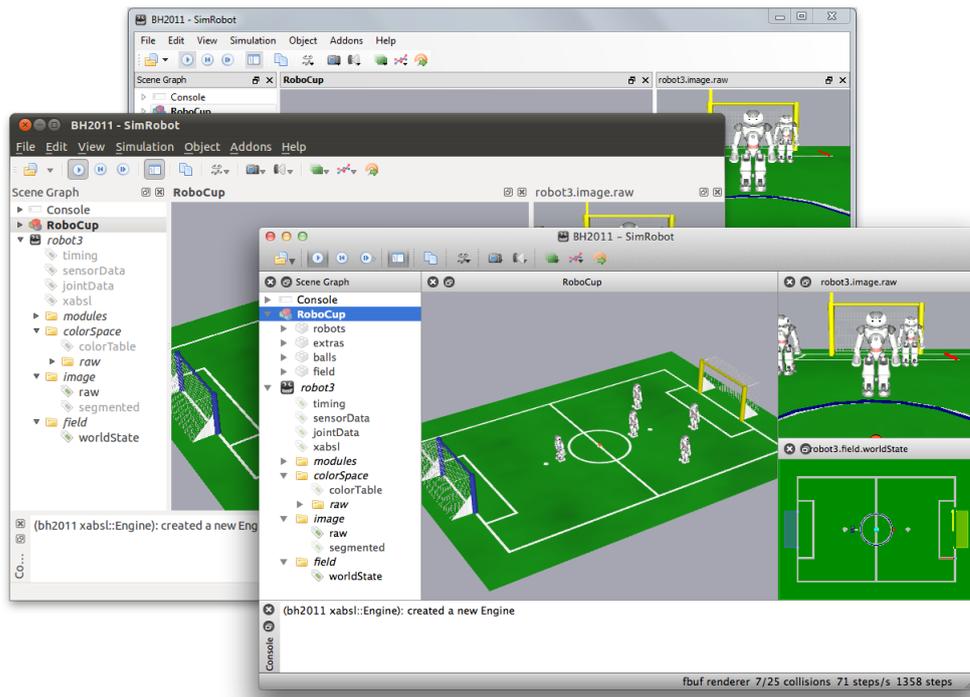


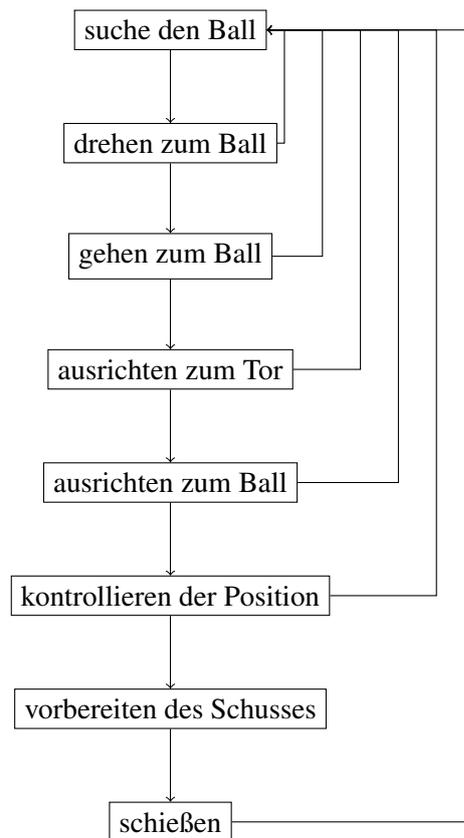
Abbildung 2.3: Simulator [29]

kompletter Spiele nahezu alle Visualisierungen, welche bei einer effizienten Entwicklung benötigt werden. Im weiteren Verlauf dieser Arbeit werden besonders die Auswertungen bezüglich der Zustände der künstlichen Intelligenz anhand des Xabsl-Viewers eine tragende Rolle spielen.

## 2.3 Fußballspieler

Die Verwendung des beschriebenen Frameworks stellt die Möglichkeit zur Erstellung eines mehrschichtigen Verhaltens zu Verfügung, wobei die einzelnen Schichten untereinander hierarchisch angeordnet werden. Im konkreten Fall wird damit ein Fußballspieler programmiert, dessen einzige Aufgabe das Erzielen von Toren ist.

Die erste Hierarchieschicht des Verhaltens bildet dabei die “Körperkontrolle”, welche es erlaubt, den Roboter in verschiedene Zustände, wie “Ready”, “Set” oder “Play” zu versetzen. In jedem dieser Zustände erfüllt der Roboter im Rahmen eines Robocupspiels unterschiedliche Aufgaben. Für die folgenden Untersuchungen sind ausschließlich die Zustände “Set”, welcher zur Orien-



**Abbildung 2.4:** Verhalten des Roboters

tierung dient, und “Play”, wobei der Roboter zu spielen beginnt, relevant. Eine weitere Aufgabe dieser Schicht ist die Erkennung von Stürzen des Roboters und deren Behebung. Das eigentliche Spielverhalten des Roboters wird nur im Zustand “Play” aufgerufen und kann auch während der Ausführung jederzeit von einer höheren Schicht in der Verhaltenshierarchie beendet werden.

Das Verhalten des Roboters während eines Spieles ist in Abbildung 2.4 schematisch dargestellt. Den Startpunkt dabei bildet ständig die Suche des Balls, welche ohne Beschränkung bis zu dessen Erkennen durchgeführt wird. Anschließend dreht der Roboter sich zum Ball und geht auf ihn zu. Sobald dieser erreicht wird, erfolgt zunächst eine exakte Ausrichtung zum Ball und zum gegnerischen Tor. Den Abschluss der Aktion bildet der Schuss selbst, welcher bestenfalls im Tor landen sollte. Anschließend beginnt das Verhaltensmuster mit der Suche des Balls von Neuem.

# Biologisches Immunsystem

Nahezu jeder Organismus besitzt ein Immunsystem, denn nur so ist es möglich, den ständigen Angriffen verschiedenster Pathogene<sup>1</sup> standzuhalten. Durch die Hinderung an ihrer Verbreitung und Vermehrung sind diese Eindringlinge nicht in der Lage dem Organismus zu schaden oder ihn zu töten. Die Pathogene besitzen eine enorme Vielfalt an hochkomplexen Methoden, um einen Organismus zu befallen.

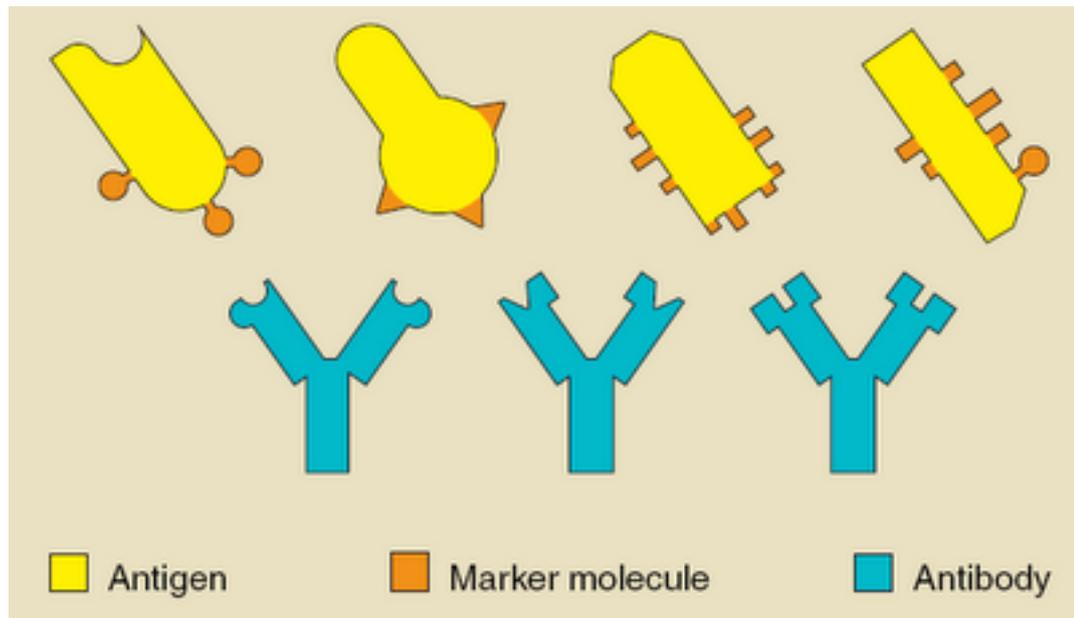
Doch ebenso ausgefeilt wie mögliche Eindringlinge arbeiten, agiert auch unser Immunsystem. Die schnelle Anpassungsfähigkeit an neue Angriffe sowie die Merkfähigkeit vergangener Angriffsstrategien machen das Immunsystem zu einem äußerst effektiven und genauso komplexen Verteidigungsapparat. Es besteht aus einer nahezu unzählbaren Menge an Zellen und gelösten Substanzen, welche über den gesamten Organismus verteilt auftreten. Im folgenden Kapitel werden auf Grundlage der Arbeit von Hannes Schlender und Jo Schilling [30] fundamentale Bestandteile des menschlichen Immunsystems angeführt sowie deren Zusammenarbeit erläutert.

## 3.1 Grundlagen des Immunsystems

Bei der Abwehr von Krankheitserregern bildet das Immunsystem den wichtigsten Bestandteil eines stark vernetzten Systems von Abwehrmechanismen. Als äußerste Barriere können Haut

---

<sup>1</sup>griechisch pathos: Leiden, Krankheit; gennan: erzeugen



**Abbildung 3.1:** Rezeptoren und Antigene<sup>2</sup>

und Schleimhäute angesehen werden, welche durch ihre Rissfestigkeit sowie einen Säuremantel einen Großteil der Angriffe abwehren können. Ähnliche Strategien werden im Bereich des Magens durch dessen Magensäure oder im Darm durch alkalische Flüssigkeiten eingesetzt. Weitere Bestandteile dieses ersten Abwehrrings stellen Speichel und Tränenflüssigkeit dar.

Nach der ersten Barriere treffen Eindringlinge auf das Immunsystem, dessen Bestandteile im gesamten Körper verteilt auftreten. Dabei wird zwischen zellulärem und humoralem Immunsystem unterschieden. Damit das Immunsystem den Angreifer als solchen erkennt, werden Muster der Angreifer verglichen. Dabei spielen die Komponenten des zellulären Immunsystems, die Immunzellen eine tragende Rolle. An der Oberfläche der Zellen befinden sich mehrere Rezeptoren. Aufgrund der Form sowie elektrischen Ladung ermöglichen diese Rezeptoren die Anbindung der Zelle an gewisse Strukturen (siehe Abbildung 3.1). Die Anbindungsstelle des Rezeptors wird als Epitop bezeichnet. Sobald ein Eindringling eine Bindung mit einem Rezeptor eingeht, wird dieser als fremd eingestuft und als Antigen bezeichnet. Die darauffolgende Reaktion der Immunzellen hängt nun vom Zelltyp ab. Einige Typen werden im folgenden Kapitel detailliert angeführt. Das Immunsystem besteht allerdings nicht nur aus Zellen, wobei alle restlichen Elemente zum humoralen Teil des Immunsystems zählen. Dieses besteht vorwiegend aus Immun-

<sup>2</sup><http://www.web-books.com/eLibrary/Medicine/Physiology/Immune/Antigen.htm>  
zuletzt besucht 11.11.2011

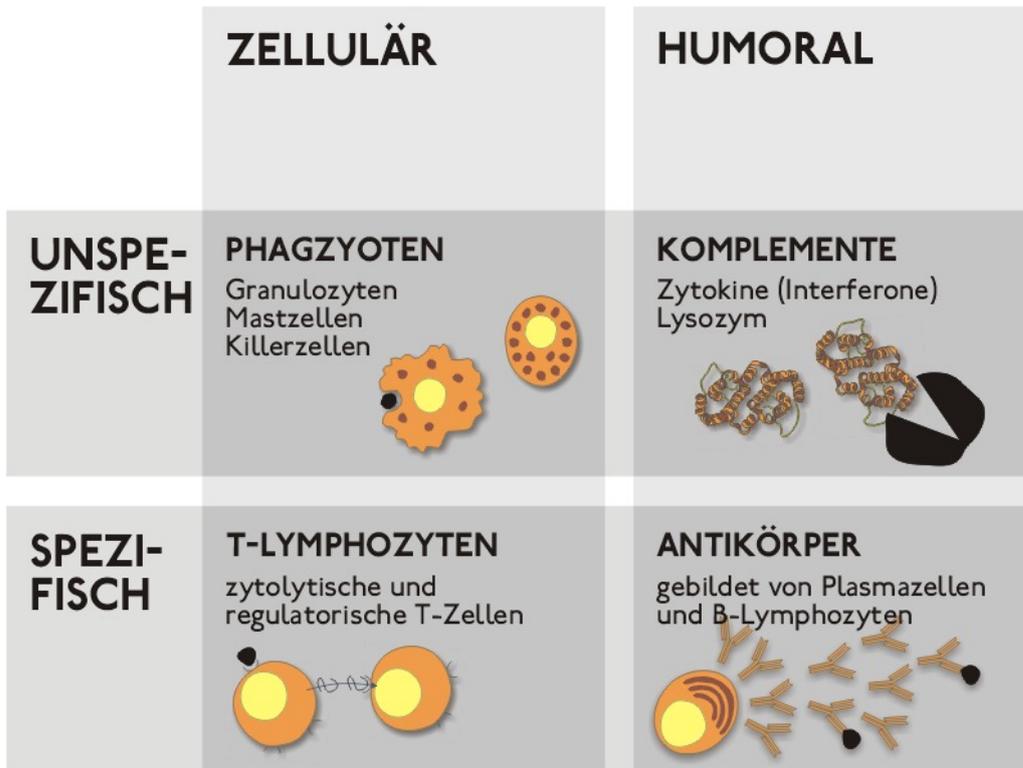


Abbildung 3.2: Einteilung der Bestandteile des Immunsystems<sup>3</sup>

botenstoffen, Antikörpern und anderen immunwirksamen Proteinen.

Die Unterscheidung zwischen angeborenem und adaptivem Immunsystem bietet eine weitere Möglichkeit zur Gruppierung verschiedener Bestandteile des Immunsystems. Diese grundlegende Einteilung der Bestandteile des Immunsystems findet sich in Abbildung 3.2 wieder. Die Stärke des angeborenen Immunsystems ist seine Schnelligkeit. Allerdings sind diese Verteidigungsmaßnahmen sehr unspezifisch, wodurch bei einigen Bakterien und Viren nur eine geringe Wirkung erzielt wird (siehe Kapitel 3.2). Besonders die Weiterentwicklung der Pathogene bringt diese Art der Verteidigung schnell an seine Grenzen, weshalb viele Lebewesen einen zweiten Abwehrmechanismus besitzen.

Das adaptive oder erworbene Immunsystem (siehe Kapitel 3.3) entwickelt bei Kontakt mit Pathogenen spezielle Immunantworten und ist in der Lage, diese Angriffsmuster abzuspeichern. Beim nächsten Angriff dieser Art kann dadurch sehr viel schneller darauf reagiert werden.

<sup>3</sup>[http://de.wikipedia.org/wiki/Datei:Uw\\_grafik\\_immunsystem.jpg](http://de.wikipedia.org/wiki/Datei:Uw_grafik_immunsystem.jpg) zuletzt besucht 12.11.2011

## 3.2 Das angeborene Immunsystem

Zum angeborenen Immunsystem zählen wiederum eine erhebliche Anzahl humoraler Faktoren und Immunzellen. Die Zellen können dabei wiederum unterteilt werden in Fresszellen und Granulozyten. Basierend auf unspezifischen Rezeptoren erzeugt dieses Teilsystem eine erste Immunantwort bei unterschiedlichsten Angriffen.

Die Fresszellen haben dabei die Aufgabe, Eindringlinge anhand der Rezeptoren zu erkennen und zu töten, indem sie diese aufnehmen und in den Vesikeln<sup>4</sup> verdauen. Zu den Fresszellen zählen Makrophagen, welche sich in allen Geweben befinden, Zytokine<sup>5</sup> produzieren und Antigene<sup>6</sup> präsentieren. Außerdem werden die Monozyten, welche als Vorläufer der Makrophagen bezeichnet werden, mitgerechnet. Natürliche Killerzellen, welche Viren und Tumore angreifen, und dendritische Zellen, welche ebenfalls Pathogene verdauen, anschließend zu den Lymphknoten wandern und dort durch die Präsentation der Antigene das adaptive Immunsystem aktivieren, vervollständigen die Klasse der Fresszellen. Die Granulozyten greifen hingegen nur spezielle Pathogene an. So zerstören neutrophile Granulozyten nur Bakterien, Viren und Pilze in den Blutbahnen, eosinophile Granulozyten ausschließlich Parasiten und basophile Granulozyten, auch Mastzellen genannt, töten Parasiten mittels Gift, welches sie mitführen.

Der humorale Teil des angeborenen Immunsystems besteht ebenfalls aus mehreren Elementen. Die Zytokine sind Stoffe, welche das Wachstum und die Vermehrung von Zellen beeinflussen. Diese Stoffe werden von vielen Immunzellen produziert, um andere Immunzellen anzulocken. Außerdem hilft das Lysozym, welches sich in der Lympheflüssigkeit sowie in vielen Körpersekreten befindet, bei der Abwehr mit, indem es die Zellwände einiger Pathogene zerstört und dadurch tötet. Das Komplementsystem bildet aus 20 verschiedenen plasmalöslichen Proteinen eine eigene Unterordnung, das die Aufgaben der Anregung von Fresszellen und die Vernichtung der Eindringlinge erfüllt.

---

<sup>4</sup>Bläschen

<sup>5</sup>unspezifische Kommunikationsmoleküle

<sup>6</sup>Spezifischer Teil des Erregers

### 3.3 Das adaptive Immunsystem

Die dendritischen Zellen aktivieren nicht nur die T-Zellen, sondern ein ganzes System an Zellen, welche in Zusammenarbeit mit den humoralen Bestandteilen des Immunsystems eine sekundäre Immunantwort durchführen. Die Immunzellen des erworbenen Immunsystems teilen sich grundsätzlich in zwei Gruppen auf, die B-Lymphozyten und T-Lymphozyten<sup>7</sup>. Der Aufbau der Rezeptoren stellt sich in beiden Gruppen sehr ähnlich dar und ihre Aktivierung erfolgt in beiden Fällen durch die Bindung an ein Antigen. Des Weiteren werden beide Zelltypen im Knochenmark gebildet, wobei sich nachher ihre Wege schnell trennen.

Die B-Zellen reagieren auf Pathogene mit einer spezifischen Immunantwort und sind außerdem in der Lage, selbst Antigene zu präsentieren. Ihre vorwiegende Aufgabe stellt die Produktion von Antikörpern, welche frei im Blut schwimmende B-Zell-Rezeptoren sind, dar. Durch die Aktivierung der B-Zelle mittels passendem Antigen, entwickelt sie sich zu einer Plasmazelle. Diese schwimmt weiterhin im Blut und entwickelt in großer Anzahl spezifische Antikörper gegen den bekannten Krankheitserreger. Ehemalige Plasmazellen können sich zu B-Gedächtniszellen weiterentwickeln und dadurch Jahrzehnte in den Lymphen erhalten bleiben. Bei einer erneuten Infektion kennt das Immunsystem dadurch das Pathogen bereits und kann schneller reagieren.

Die T-Helferzelle, welche auch als Kommunikator des Immunsystems bezeichnet wird, gehört zur Gruppe der T-Lymphozyten und besitzt die Möglichkeit, Antigene auf antigenpräsentierenden Zellen zu erfassen. Sobald ein Antigen erkannt wurde, aktivieren sie die Plasmazellen sowie die T-Killerzellen. Eine Art des Gegenspielers zur Helferzelle stellt die T-Suppressorzelle dar, welche bei Bedarf in der Lage ist, die Zytokinausschüttung zu hemmen und die B-Zellen zu deaktivieren. Die bereits erwähnte T-Killerzelle hat, wie der Name schon andeutet, die Aufgabe Eindringlinge zu zerstören. Durch das Durchlöchern der Hülle kann die Killerzelle das Pathogen befallen und anschließend vollständig auflösen. Ähnlich wie bei den B-Lymphozyten gibt es auch bei den T-Zellen Gedächtniszellen, welche gelernte Immunantworten anhand der Form des Antigens speichern.

---

<sup>7</sup>Kurzform: B-Zellen und T-Zellen

### 3.4 Die Arbeitsweise des Immunsystems

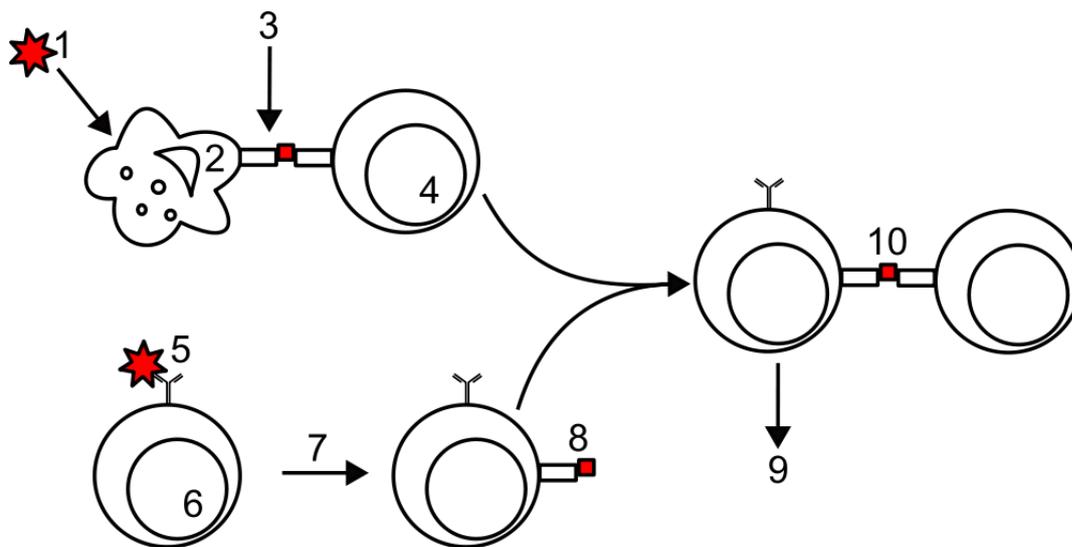
Die Möglichkeiten für Pathogene im Körper sind sehr vielfältig. So dienen beispielsweise kleinste Verletzungen als Angriffspunkt. Einige Eindringlinge nutzen die Gelegenheit und wandern in das Gewebe unter die Haut, von wo sie sich über Lymph- und Blutbahnen über den gesamten Körper verteilen. Allerdings treffen die Angreifer noch in der Haut auf das angeborene, humorale Immunsystem in Form von Defensinen, welche die Hülle der Zellen zerstören und sie dadurch töten. Trotzdem schaffen einige Pathogene den Weg in das Gewebe, wo Lysozyme - auch ein Bestandteil des angeborenen, humoralen Immunsystems - versuchen, die Hülle der Pathogene zu zerstören. Geraten die Eindringlinge weiter in die Blutbahnen, greift das humorale Immunsystem mit voller Härte durch. Dabei kommt das Komplementsystem mit 20 verschiedenen Plasmaproteinen zum Einsatz, welche nicht nur Zellwände auflösen, sondern auch weitere Fresszellen anlocken.

Die Fresszellen gehören zur angeborenen, zellulären Verteidigung und verdauen alles Fremde in ihrer Umgebung. Monozyten, Makrophagen und Granulozyten sind Fresszellen, die in unterschiedlichen Teilen des Körpers Angreifer bekämpfen und dabei mittels Zytokinen weitere Immunzellen aktivieren. Die Aufgaben der Monozyten und Makrophagen gehen allerdings noch weiter. Bei der Zerstörung eines Eindringlings werden kleine Teile verschont, welche zur Aktivierung des adaptiven Immunsystems dienen. Dazu werden die sogenannten Antigene in den MHC<sup>8</sup> eingeklinkt, welcher nicht nur zum Auslösen der spezifischen Immunantwort, sondern auch als Identifikation der körpereigenen Bestandteile dient. Nachher begeben sich diese Immunzellen auf den Weg zu den Lymphorganen.

In Bewegung Richtung der Lymphknoten befinden sich auch die dendritischen Zellen, welche bereits im Bereich der Haut mit dem Prozess der Verdauung begonnen haben. Anschließend begeben sie sich zur Präsentation der Antigene auf die Reise in die Lymphorgane. Diese frühe Aktivierung der spezifischen Immunantwort bringt große Vorteile bei der Verteidigung des Körpers gegen die Pathogene.

---

<sup>8</sup>Major Histocompatibility Complex



**Abbildung 3.3:** Aktivierung von T-Zelle und B-Zellen durch Antigen-präsentierende Immunzellen<sup>9</sup>- 1 Antigen; 2 Antigen-präsentierende Zelle; 3 Antigenpräsentation; 4 T-Helfer-Zelle; 5 gebundenes Antigen; 6 B-Lymphozyt; 7 Antigen-Prozessierung; 8 Antigenpräsentation; 9 Produktion Antigen-spezifischer Antikörper; 10 Aktivierung eines B-Lymphozyten

### Aktivierung der T-Zellen

In den Lymphknoten angelangt, präsentieren die Immunzellen des angeborenen Immunsystems die Antigene den T-Zellen, welche dadurch die Immunantwort des adaptiven Immunsystems starten können. Dabei tasten T-Killerzellen und T-Helfer-Zellen die dargestellten Antigene mit ihren Rezeptoren ab (Abbildung 3.3 Markierung 1-4). Sobald sich die Antigene in die Rezeptoren eingeklinkt haben, werden diese aktiv. T-Killerzellen beginnen damit Viren zu bekämpfen. T-Helferzellen greifen Bakterien an. Außerdem sind aktivierte Helfer-Zellen in der Lage, sich weiter zu entwickeln. Einige verwandeln sich in T-Suppressorzellen zur Regulierung der Immunantwort, andere in T-Gedächtniszellen für die Abwehr zukünftiger Angriffe. Die Stimulation der B-Lymphozyten ist eine weitere Aufgabe der T-Helfer-Zellen, wodurch der nächste Abwehrmechanismus des Immunsystems ausgelöst wird (Abbildung 3.3 Markierung 10).

<sup>9</sup>Urheber: Masur - <http://de.wikipedia.org/wiki/Antigenpräsentation> zuletzt besucht 18.07.2012

## Aktivierung der B-Zellen

Die erste Reifungsphase der B-Zellen erfolgt im Knochenmark, wobei einige Zellen bereits von Makrophagen eliminiert werden, da sie sich mittels Rezeptoren an Knochenmarkzellen binden. Die Rezeptoren der B-Lymphozyten liegen in der Zellmembran und werden auch als Antikörper bezeichnet. Jede Zelle besitzt nur einen Typ von Rezeptoren und kann daher bei einer Aktivierung nur einen Typ an Antikörpern sezernieren<sup>10</sup> (Abbildung 3.3 Markierung 5-9).

Die Antikörper ermöglichen eine sehr schnelle Vernichtung der Pathogene, da sie zu einer äußerst spezifischen Erkennung fähig sind. Weiters übernehmen die sezernierten Antikörper die Aufgabe der Markierung von Eindringlingen. Sie binden dazu an das angreifende Antigen und locken Stoffe des Komplementsystems an. Dadurch kommt es zu einer neuerlichen Stimulation des angeborenen Immunsystems, was abermals eine Kette von Reaktionen auslöst.

Werden B-Lymphozyten von T-Helfer-Zellen stimuliert, erfolgt eine Teilung. Einerseits entwickeln sich B-Gedächtniszellen, welche eine lange Lebensdauer aufweisen und bei erneuten Infektionen schnell reagieren, andererseits Plasmazellen. Diese Immunzellen begeben sich zunächst auf den Weg zum Angriffspunkt der Pathogene. Während der Reise durch die Blutbahnen sondern die Plasmazellen eine große Zahl an Antikörpern ab. Außerdem sind auch Plasmazellen in der Lage, von Antikörpern markierte Angreifer aufzulösen, wodurch wiederum andere Fresszellen angelockt werden.

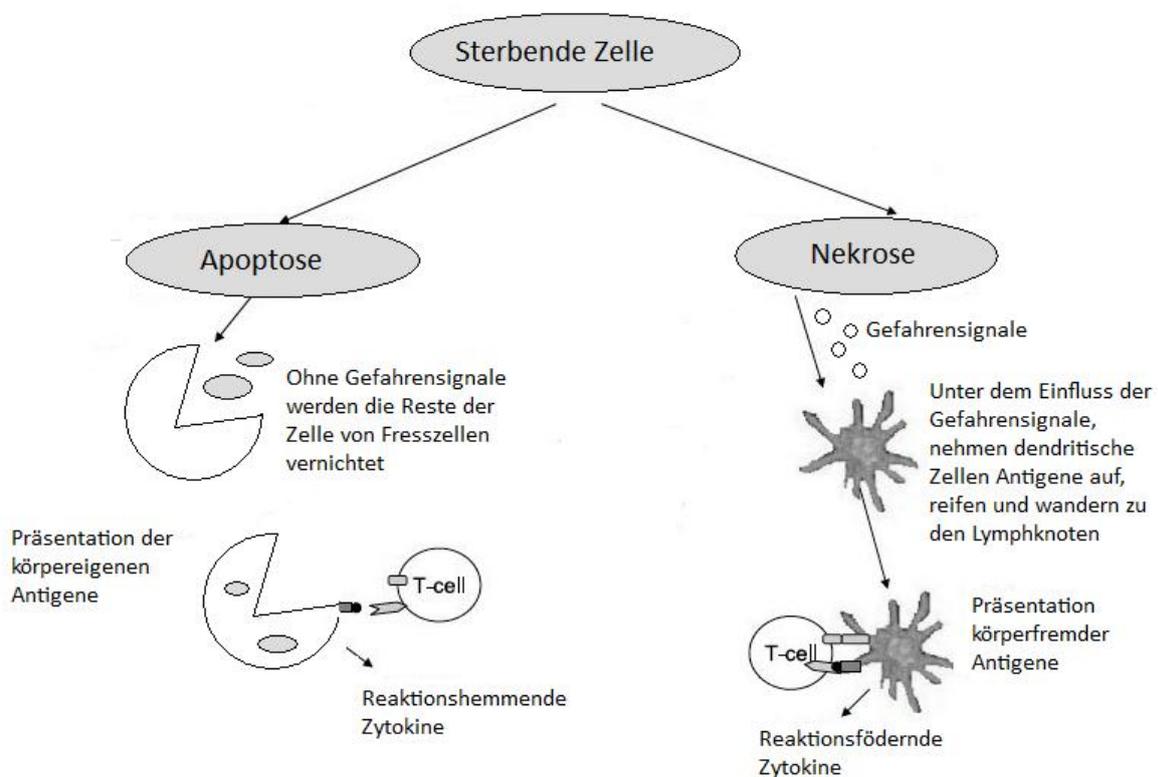
## 3.5 Gefahrentheorie

Im Jahre 1994 veröffentlichte Polly Matzinger [25] die Gefahrentheorie, die das Selbst-Nichtselbst-Modell [5] komplettiert. Bei der Aktivierung des adaptiven Immunsystems spielen allerdings weitere Komponenten eine tragende Rolle, nämlich die Gefahrensignale. Diese Signale werden allerdings nicht von den Immunzellen erzeugt, sondern von den Zellen des umgebenden Gewebes. Dadurch ändert sich die Betrachtungsweise auf das Immunsystem, welches in Wechselwirkung mit dem Gewebe steht und darin integriert ist.

Die Grundlage für die Theorie von Matzinger bildet der Tod von Zellen. Die Unterscheidung zwischen einer natürlichen Todesursache und der Zerstörung durch einen Angreifer stellt da-

---

<sup>10</sup>absondern



**Abbildung 3.4:** Gefahrenmodell (danger model)

bei den ausschlaggebenden Ansatzpunkt dar, wie in Abbildung 3.4 dargestellt wird. Bei einem geplanten Tod oder Apoptose<sup>11</sup> verdaut sich die Zelle zu einem Großteil selbst ohne dabei die äußere Hülle zu zerstören. Das Entweichen jeglicher Komponenten der Zelle wird dadurch verhindert. Gegen Ende der Apoptose werden Makrophagen durch spezifische Moleküle an der Zellmembran angelockt. Die sterbende Zelle wird von den Fresszellen aufgenommen und verdaut.

Bei einer Infektion oder Verletzung der Zelle durch Pathogene (Bakterien oder Viren) und dem daraus resultierenden Zelltod laufen die einzelnen Zersetzungsschritte hochgradig chaotisch ab. Bei der inneren Verdauung der Nekrose<sup>12</sup> wird durch die "wilden" Zerstörungen die äußere

<sup>11</sup>griech. apoptose = abfallen

<sup>12</sup>griech. nekrose = absterben

Hülle der Zelle zerrissen. Die ausgetretenen Zellbestandteile bewegen sich in den extrazellulären Flüssigkeiten und werden als Gefahrensignale erkannt. Wahrscheinlich besitzt der Körper noch viele weitere Gefahrensignale, allerdings sind bis heute nicht alle bekannt.

Als Folge der Nekrose ändert sich die Konzentration derartiger Moleküle in den Körperflüssigkeiten, worauf besonders dendritische Zellen reagieren. Bei ausreichenden Gefahrensignalen wandern diese Zellen in die Lymphorgane und präsentieren den aufgesammelten "Schmutz" als Antigen, wodurch wiederum eine Aktivierung der T-Zellen erfolgt.

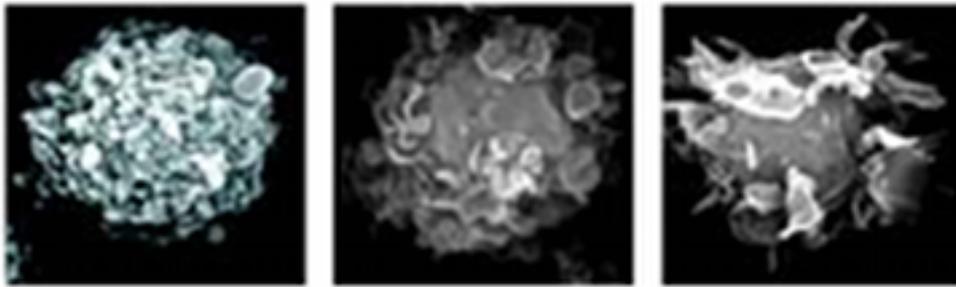
Das Gefahrenmodell fügt bei der Erkennung von Angriffen eine zweite Dimension hinzu, die die Unterscheidung zwischen Selbst und Nichtselbst durch die Abgrenzung zwischen gefährlich und ungefährlich ergänzt. Dementsprechend ist in Folge eine Erkennung eines "gefährlichen Selbst" (Tumore oder Mutation der Zellen) und "ungefährlichen Nichtselbst" (Fötus) möglich [12].

### **3.6 Dendritische Zelle**

Dendritische Zellen werden oft als Verbindungsglied zwischen angeborenem und adaptivem Immunsystem bezeichnet, da sie die einzigen Zellen im Organismus sind, welche eine adaptive Immunantwort effektiv einleiten können. Erstmals beschrieben wurden diese Zellen 1973 von Ralph M. Steinman, der dafür 2011 den Nobelpreis für Medizin erhalten hat. Aktuelle Forschungen beschäftigen sich auf der molekularen und zellulären Ebene mit diesen äußerst vielseitigen Zellen.

Den bevorzugten Lebensraum der dendritischen Zellen stellen die meisten Oberflächen des Körpergewebes dar, wobei sich eine große Anzahl der Zellen auf der Haut, im Rachen oder in den inneren Schleimhäuten finden. Durch ihre verzweigten Ausläufer sind die dendritischen Zellen in der Lage, durch das Deckgewebe der Oberflächen hindurch Antigene zu erreichen und aufzunehmen, ohne dass die Oberfläche beschädigt sein muss. Direkt im Anschluss wandern die Zellen zum nächsten lymphatischen Organ, wo sie durch die Präsentation der Signatur möglicher Eindringlinge deren Erkennung durch T-Zellen ermöglichen. Außerdem wird auf humoraler Ebene durch die Verbreitung von Zytokinen die adaptive Immunantwort ausgelöst oder verstärkt.

Im Laufe ihres Lebens nehmen dendritische Zellen drei verschiedene Reifestadien ein (siehe Abbildung 3.5). Die sehr verbreitete sternförmige Form der Zellen beschreibt eine unreife dendritische Zelle im peripheren Gewebe. Die Fortsätze, welche auch als Dendriten bezeichnet werden



**Abbildung 3.5:** Unreife dendritische Zellen, halb-reife dendritische Zellen und reife dendritische Zellen [12]

und in alle Richtungen vom Körper der Zelle abstrahlen, sind in diesem Reifestadium durchgehend in Bewegung, um mögliche Antigene zu erreichen. Nach einer Aufnahme von Antigenen bewegt sich die Zelle in Richtung eines Lymphknotens, wobei sie ihre Gestalt verändert. Die Dendriten werden durch Membranausstülpungen und schleierartige Falten ersetzt. Außerdem verlieren sie die Fähigkeit der Antigenaufnahme. In den lymphatischen Organen erreichen die dendritischen Zellen ihr letztes Reifestadium und beginnen mit der Aktivierung von mehreren tausend T-Zellen, wozu keine andere antigenpräsentierende Zelle in der Lage ist [31].

Neben dem Präsentieren der Antigene von Pathogenen haben dendritische Zellen noch eine zweite wichtige Aufgabe. Eine unreife dendritische Zelle sammelt ständig Antigene in ihrer Umgebung. Da oftmals keinerlei Infektion im Körper vorliegt, sind dies häufig Proteine von toten körpereigenen Zellen. Diese Fähigkeit bildet die Grundlage für die Gefahrtheorie nach Matzinger (Kapitel 3.5). Nach der Reise zu den lymphatischen Zellen und der Reifung der dendritischen Zelle können je nach Signal verschiedene Szenarien ablaufen. Bei einem programmierten Zelltod kommt es zur Entwicklung von T-Suppressorzellen, welche eine Autoimmunreaktion verhindern. Eine Nekrose hingegen bewirkt die Aktivierung der adaptiven Immunantwort.

### 3.7 Zusammenfassung

Das biologische Immunsystem ist durch eine effiziente Mustererkennung sowie Interpretation von Signalen in der Lage, sich gegen nahezu jeglichen Angriff zu wehren. Diese enorme Robustheit fundiert nicht nur auf der komplexen Architektur aus Mehrschichtigkeit, Verteiltheit

und autonomes Handeln, sondern auch auf der äußerst dynamischen Anpassung, dem Gedächtnis und der Fähigkeit des Lernens. Durch die Evolution hat sich dieses imposante System ständig weiterentwickelt und ist eine reiche Quelle der Inspiration verschiedenster Disziplinen.

Für die Fehlererkennung in einem Robotersystem werden Teile der biologischen Abwehrmechanismen abstrahiert, wobei durch Abwesenheit von Angreifern deutliche Unterschiede zu erkennen sind. Antigene werden somit nicht von Angreifern erzeugt, sondern stellen systemeigene Zustände dar, wodurch folglich auch die Vernichtung der Angreifer in diesem Fall bedeutungslos ist. Andere Methoden wie beispielsweise große Teile der Gefahrentheorie fließen abstrahiert in die künstlichen Abläufe ein und spielen eine tragende Rolle beim Erkennen von Fehlverhalten.

Die Übertragung der äußerst komplexen Vorgänge in ein Computersystem konnte bisher nur eingeschränkt erreicht werden. Dabei gibt es in vielen Bereichen noch offene Probleme, aber auch Errungenschaften, welche im folgenden Kapitel angeführt werden.

# Künstliche Immunsysteme

Künstliche Immunsysteme fassen eine Vielzahl von Algorithmen zusammen, welche ihre Ansätze im biologischen Immunsystem finden. In den neunziger Jahren hat sich der Zweig der Computational Intelligence (CI) etabliert, dessen Ursprung zwanzig Jahre früher liegt. Es existieren allerdings keine spezifischen Prozeduren der künstlichen Immunsysteme, vielmehr handelt es sich um eine Sammlung an Algorithmen, die über die letzten zwanzig Jahre entwickelt wurden und in verschiedensten Bereichen wie Muster- und Fehlererkennung, IT-Sicherheit und einer Reihe weiterer Forschungsanwendungen eingesetzt werden. Dennoch gibt es einige Konzepte, welche aufbauend auf Modellen der Immunologie das Bild der künstlichen Immunsysteme maßgeblich geprägt haben: Negative Selektion, Klonale Selektion, Netzwerkmodelle und in den letzten Jahren die Gefahrentheorie. Im folgenden Kapitel werden diese geschichtlichen Meilensteine der künstlichen Immunsysteme im Detail angeführt. Dies erfolgt in Anlehnung an die Arbeit von Greensmith, Whitbrook und Aickelin [16].

## 4.1 Modelle der Immunologie

Die ersten Schritte in Richtung künstlicher Immunsysteme wurden Mitte der siebziger Jahre gemacht, als die ersten beschränkten Abbildungen von biologischen Abläufen im Immunsystem erstellt wurden. Diese Modelle bildeten die Grundlage für weiterführende Forschungen in den unterschiedlichsten Disziplinen und in späteren Jahren waren sie der Ausgangspunkt für die



Algorithmen der künstlichen Immunsysteme. Allerdings werden die biologischen Prozesse des Immunsystems bis heute erforscht, dabei werden ständig neue Fragen aufgeworfen. Dennoch ist die Wissenschaft durch interdisziplinäre Erkenntnisse in der Lage sehr viel komplexere und genauere Modelle zu erstellen, welche schrittweise auch Einfluss auf künstliche Immunsysteme nehmen werden.

Erste Theorien und Modelle [21] beschäftigten sich mit immunen Netzwerken. Die vorgestellte Theorie stellt das Immunsystem als ein Netz aus Molekülen und Zellen dar. Dabei handelt es sich um miteinander verbundene B-Zellen für die Antigen-Erkennung, welche sich auch in Abwesenheit von Antigenen gegenseitig beeinflussen. Außerdem wurden erste Abstraktionen bezüglich der Affinität<sup>1</sup> zwischen Zellen in die Hypothese eingearbeitet. Durch Paratope<sup>2</sup> und Epitope<sup>3</sup> (bei Molekülen als Idiotope bezeichnet) ist jedes Element des Netzes in der Lage zu “erkennen” und “erkannt zu werden”. Paratope der Antikörper und Epitope der Antigene sind zueinander komplementär und agieren in einer Schloss-Schlüssel-Beziehung, wobei ein Paratop mehrere Schlösser öffnen kann. Dadurch entsteht ein selbstregulierendes System, welches mittels Verstärkung und Unterdrückung einen stabilen Immunspeicher bildet. Ergänzend wurde es durch die Verteilung über mehrere Ebenen zu einem hoch komplexen reaktiven Netzwerk. Die B-Zellen werden nicht länger als isoliertes Subsystem betrachtet, sondern agieren miteinander, woraus sich dynamisches Netzwerkverhalten entwickelt. Das Netzwerk selbst erhält also die Fähigkeit sich ständig zu kontrollieren, zu warten und anzupassen.

Etwa zur selben Zeit, 1978, legte Burnet [6] mit seiner Arbeit über die klonale Selektion und den daraus resultierenden Metriken der Affinität den Grundstein für den gleichnamigen Algorithmus, der lange Zeit das Fundament für viele Entwicklungen im Bereich künstlicher Immunsysteme darstellte. Ein neues Modell, abseits von neuronalen Netzen und genetischen Algorithmen entwickelte Farmer u. a. in seiner Arbeit [10], in dem er den Ansatz von Burnet in Kombination mit dem Netzwerkmodell von Jerne [21] brachte und die Interaktion zwischen Antikörpern mathematisch beschrieb. Die Theorie der klonalen Selektion beschreibt die Vermehrung von Zellen bei einer Stimulation durch ein Antigen. Dabei erfolgt die Vermehrung durch eine Kopie der Eltern sowie einen spezifischen Mutationsmechanismus. Außerdem erfolgt nach jedem Evolutionszyklus aufgrund der Anpassung an das vorliegende Antigen ein Ausschluss gewisser Zellen. Auf weitere Konzepte der klonalen Selektion wird im Abschnitt 4.3 eingegangen.

---

<sup>1</sup>Maß für die Bindungsstärke

<sup>2</sup>Spezifische Bindungsstelle eines Antikörpers

<sup>3</sup>Molekülabschnitte eines Antigens

Zum Zweck der Toleranz gegenüber eigenen Zellen wurde die Theorie der negativen Selektion entwickelt. Dieses Prinzip behandelt ebenfalls die Erkennung von fremden Antigenen, aber auch die Akzeptanz von körpereigenen Zellen. Dabei werden gereifte Fresszellen einer Kontrolle unterzogen, damit nur jene Fresszellen in den Kreislauf gelangen, welche fremde Zellen angreifen. Diese Art der Kontrolle wurde von Boer, Perelson und Kevrekidis 1993 vorgestellt [9].

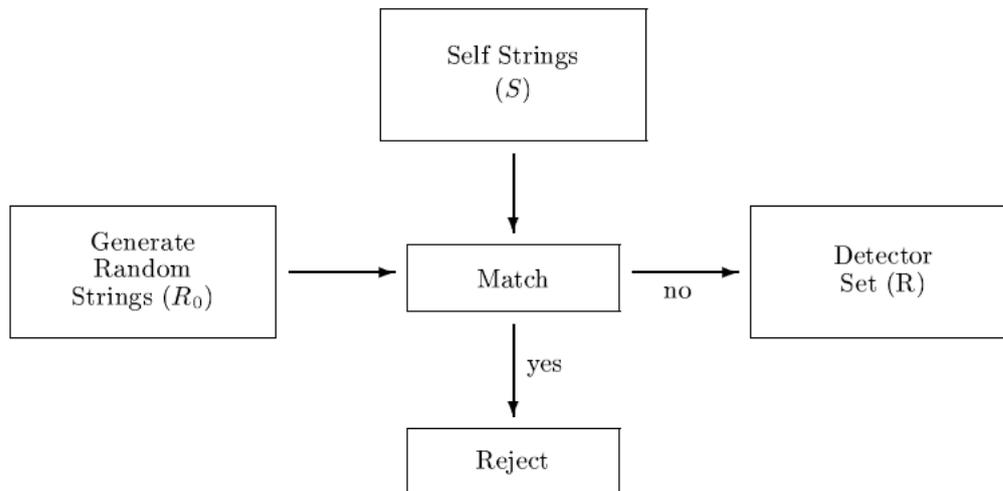
Um die Jahrtausendwende bekamen die theoretischen Modelle der Immunsysteme und damit die gesamte Forschungsrichtung neuen Auftrieb. Aus der neuen Betrachtungsweise des Immunsystems als komplexe Struktur und den zeitgleichen interdisziplinären Forschungsergebnissen entwickelten sich unkonventionelle Theorien. Besonders das Buch "Tending Adam's Garden : Evolving the Cognitive Immune Self" [8] von Cohen löste eine unübliche Akzeptanz von theoretischen Modellen im Bereich künstlicher Immunsysteme aus.

## **4.2 Konzept der negativen Selektion**

Das Konzept der negativen Selektion baut auf der Grundlage des Selbst-Nichtselbst-Modelles auf, welches vor allem durch die Arbeit von Forrest getragen wurde [11]. Dabei war diese Software für Virenerkennung das erste System, welches auf dem Prinzip der künstlichen Immunsysteme beruhte. Im Bereich der IT-Sicherheit schlug dieser Ansatz hohe Wellen und bildet bis heute die Grundlage für weitreichende Forschungen.

Das Prinzip dieses Algorithmus beruht auf der Idee, dass eine gewisse Menge an zufälligen binären Zeichenketten in einer Phase der Reifung mit "sicheren" Wörtern verglichen wird (siehe Abbildung 4.2). Übersteigt dabei die Ähnlichkeit der beiden Zeichenfolgen einen Grenzwert, wird das neue Muster eliminiert. Dadurch entsteht eine Sammlung an Mustern, welche als Detektoren Angriffsmuster aufzeigen sollen. Nach der eigentlichen Selektion läuft die Lebensdauer eines jeden Detektors ab. In dieser Zeit kann er "gefährliche" Muster anhand seiner Ähnlichkeit zu ihnen erkennen und eine künstliche Immunantwort auslösen.

Das System LYSIS von Forrest u. a. [19] war eine der weitreichendsten Entwicklungen in diesem Teilbereich künstlicher Immunsysteme. Das System diente zur Überwachung des Netzwerkverkehrs. In einer Trainingsphase werden zufällig Detektoren erzeugt und mittels negativer Selektion aussortiert. In der nächsten Phase müssen die Detektoren innerhalb ihrer Lebenszeit einen Grenzwert an Bindungen erreichen, um die Möglichkeit zu erhalten durch Kostimulation zu



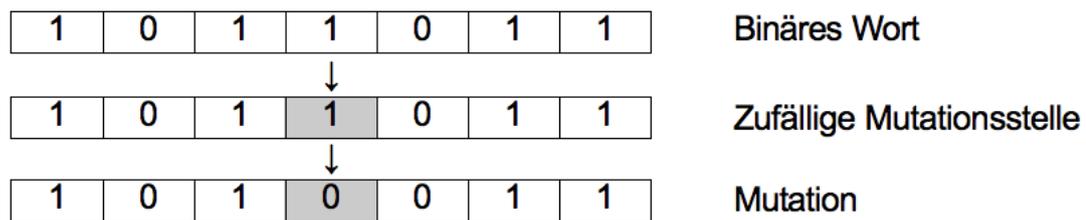
**Abbildung 4.2:** Prinzip der negativen Selektion [11]

“Gedächtniszellen” zu werden. Die Kostimulation erfolgt dabei allerdings durch den Menschen, welcher dadurch die schlussendliche Entscheidung trifft.

Untersuchungen dieser Vorgehensweisen haben einige Probleme aufgezeigt. So wird der Eingriff des Menschen in den Entscheidungsprozess als Nachteil angesehen, da dieser oft selbst zu Fehlern neigt. Außerdem ist die zufällige Erzeugung von Detektoren bei großen Wertebereichen ein Problem, da die meisten Fehler nicht erkannt werden. Einen detaillierten Überblick von den Entwicklungen im Bereich der negativen Selektion geben Ji und Dasgupta in ihrer Arbeit “Revisiting negative selection algorithms” [22].

### 4.3 Konzept der klonalen Selektion

Als Klon wird in der Biologie ein Organismus bezeichnet, der ein genetisch identisches Individuum zu seinem Klonzwilling darstellt. Dies kann durch Zellteilung oder durch Zellabspaltung geschehen, wobei es dabei auch zu Mutationen kommen kann. Im Bereich des Immunsystems wird dieser Prozess bei der Erzeugung von Antikörpern durch die B-Lymphozyten eingesetzt. Dieses Modell bildete auch die Vorlage für die Ansätze der klonalen Selektion in der Informatik.



**Abbildung 4.3:** Mutation eines Datensatzes

Im Bereich der künstlichen Immunsysteme werden hierbei Datenstrukturen oder Detektoren kopiert, welche somit dieselben Werte wie die originale Struktur enthält. Eine Verbesserung kann allerdings nur durch zufällige Mutationen (siehe Abbildung 4.3) erreicht werden. Durch eine anschließende Selektion werden nur die besten Detektoren der Population ausgewählt, welche wiederum in den Mutationszyklus eingebracht werden.

Im Jahre 2000 wurden die diesbezüglichen Ansätze von Brunet [6] aufgegriffen und in einem der bekanntesten Algorithmen künstlicher Immunsysteme, dem CLONALG [7], umgesetzt. Der Algorithmus hat hohe Ähnlichkeit mit evolutionären Mutationsalgorithmen und besitzt des Weiteren einige herausragende Eigenschaften. Diese reichen von der dynamisch einstellbaren Populationsgröße, über die Nutzung und Erforschung des Suchraumes, Erkennung von multiplen Optima, sowie definierter Abbruchkriterien.

Die klonale Selektion zeigt durch den Kreislauf von Mutation und anschließenden Filterungen Eigenschaften einer Optimierung, welche die optimalen Antikörper entwickeln sollte. In der Arbeit "Artificial immune systems: today and tomorrow" [33] zeigte Timmis, dass der klonale Selektionsalgorithmus zu den robustesten Lösungen tendiert, aber nicht notwendig zu den optimalen. Aufgrund dieser Robustheit sind diese Algorithmen besonders im Bereich der Mehrzieloptimierung äußerst erfolgreich.

#### **4.4 Konzept idiotopischer Netzwerke**

Die Modelle und Theorien von Jerne [21] wurden im Bereich der Immunologie lange Zeit wenig beachtet. Erst mit dem starken Strom der künstlichen Immunsysteme konnten seine Ideen aufgrund ihrer Fähigkeit zur flexiblen Selektion großen Anklang finden. In Anlehnung an diese Modelle wurden in der Informatik einige Lernalgorithmen entworfen, welche durch die Adap-

tivität auf lokaler und globaler Ebene sowie die Autonomie des Systems in Robotern [23], bei Clustern von Dokumenten [17] und bei der Konfiguration von Kommunikationssoftware [32] zum Einsatz kamen.

Als eine Art von Hybrid wurde eine erste Anwendung idiotopischer Netzwerke 1998 von Hunt, Timmis, Cooke und King vorgestellt [20]. Das Prinzip gegenseitiger Verstärkung und Unterdrückung bei Strukturen innerhalb des Netzwerkes war ebenso zu erkennen wie die Verwendung klonaler Ansätze. Das System mit dem Namen "JISYS" diente in weiterer Folge als Inspirationsquelle für zahlreiche weitere Projekte.

## **4.5 Konzept der Gefahrentheorie**

Vor wenigen Jahren begannen einige Forscher, sich mit der Geschichte künstlicher Immunsysteme genauer zu befassen, da sie eine gewisse Stagnation in der Entwicklung ihrer Systeme erkannten. Im Verlauf der Recherchearbeit konnten zwei fundamentale Probleme lokalisiert und abgegrenzt werden. Eine der ersten Erkenntnisse war, dass alle bisher vorgestellten Konzepte, wie negative Selektion 4.2, klonale Selektion 4.3 oder idiotopische Netzwerke 4.4, kaum von den Ansätzen der etablierten Techniken des maschinellen Lernens unterscheidbar sind. Diese Erkenntnis wurde im weiteren Verlauf als Folge eines weitaus gravierenderen Problems gedeutet.

In diesen umfangreichen Forschungen wurde darüber hinaus erkannt, dass die gesamte Geschichte künstlicher Immunsysteme und daher auch ihre Erkenntnisse auf Modellen aufbauen, die großteils sehr einfach gehaltenen wurden. Diese Abstraktionen des menschlichen Immunsystems wurden meist von einzelnen Forschern entworfen und entsprechen in einigen Punkten nicht mehr dem Stand der Wissenschaft, was eine lange Zeit niemanden gestört hat.

Denn bis zu diesem Zeitpunkt waren die Entwickler künstlicher Immunsysteme mit den bestehenden Modellen sehr zufrieden, da sie aufgrund ihres hohen Grades der Abstraktion einfach in Softwaresysteme umzusetzen waren und zu akzeptablen Resultaten führten. Obwohl diese Algorithmen der ersten Generation sehr verbreitet in den Bereichen der Anomalie-Detektion, Klassifizierung oder Mustererkennung eingesetzt werden, konnten in den letzten Jahren nur geringe Fortschritte in diesen Bereichen erbracht werden.

Aufgrund dieser Erkenntnisse und den daraus entstehenden Herausforderungen haben sich vor

einiger Zeit Forscher aus den Bereichen künstlicher Immunsysteme und Immunologie zu einer Gruppe zusammengeschlossen, um einige grundlegende Fragen zu beantworten. Besonders das Modell der Gefahrentheorie von Matzinger [25] (siehe Abschnitt 3.5), welches dem Projekt auch seinen Namen “Danger-Project” [1] gab, wurde im Detail untersucht.

Die Autoren um Aickelin versuchten in ihrer Arbeit [1] durch die Integration der Danger-Theory in ihr Modell die Effizienz künstlicher Immunsysteme zu steigern. Die Hypothese gründete sich auf die Annahme, dass Anomalien nur dann erkannt werden sollten, falls auch die entsprechenden Gefahrensignale vorhanden waren. Vor allem die false-positive-Rate sollte durch diese Erweiterung verbessert werden. Außerdem entwickelten die Autoren in weiterführenden Arbeiten die Theorie, dass künstliche Immunsysteme in ein künstliches Gewebe eingebettet werden müssen, da sie ohne umgebendes Gewebe keine Aufgabe erfüllen.

Neben den theoretischen Grundlagen gingen aus dem Danger-Project zwei weitere Entwicklungen hervor. Zunächst war dies das agentenbasierende Framework “libtissue” [34], welches ein Software-System für die Durchführung und Bewertung von AIS-Algorithmen mit realen Überwachungs- und Steuerungs-Problemen ermöglicht. Zur Evaluierung des Systems wurde ein neuartiger Algorithmus namens “tlr” entwickelt, welcher auf Modellen der dendritischen Zellen und der T-Zellen aufbaut und daher selbst einige interessante Ergebnisse lieferte.

Ein weiterer Algorithmus wurde als zweite größere Errungenschaft des Projektes veröffentlicht, der dendritische Zell-Algorithmus [12]. Auf den Modellen der dendritischen Zellen ist dieser Algorithmus der zweiten Generation in der Lage eine Kopplung zwischen der klassischen Mustererkennung und den Gefahrensignalen aus der Danger-Theory zu erzeugen. Die daraus entstehende Robustheit ermöglicht ein breites Anwendungsspektrum für diese Art von Algorithmen. So wurde der dendritische Zell-Algorithmus bereits erfolgreich im Bereich der Robotik [27] [28] sowie im Bereich der Anomalieerkennung [14] eingesetzt.

## **4.6 Zukünftige Herausforderungen**

In ihrer Arbeit “Application areas of AIS: The past, the present and the future” [18] zeigen Hart und Timmis Herausforderungen der weiteren Entwicklung mächtigerer künstlicher Immunsysteme auf. Die Autoren bezeichnen die folgenden Bereiche als fehlende “Zutaten” der Systeme, um nicht als weiteres Werkzeug in der Toolbox der Software-Ingenieure zu enden.

Lange Zeit haben sich die Entwicklungen künstlicher Immunsysteme ausschließlich mit den Modellen des adaptiven Immunsystems beschäftigt. Doch das biologische Vorbild besteht aus zwei kooperierenden Teilen, dem adaptiven und angeborenen Immunsystem, welche sich in vielerlei Hinsicht ergänzen. Der volle Wertumfang des menschlichen Immunsystems kann somit nur durch ein vollständiges Modell und darauf aufbauende Entwicklungen erreicht werden. Hierbei sehen die Autoren Hart und Timmis den größten Nachholbedarf und das umfangreichste Weiterentwicklungspotenzial.

Eine weitere Möglichkeit die fundamentalen Modelle zu erweitern, sehen die Autoren in der Aufhebung der Isolation künstlicher Immunsysteme. Diese Systeme sollten wie ihr biologisches Vorbild mit ihrer Umwelt agieren und kommunizieren. In diesem Zusammenhang wird die Aufnahmefähigkeit der Immunsysteme angesprochen, welche bisher zu großen Teilen vernachlässigt wurde. Diese neuen Perspektiven öffnen ein neues Spektrum an Forschungsrichtungen, welche kaum untersucht sind und daher ein ungeahntes Potenzial in sich bergen.

Eine der herausragendsten Leistungen des biologischen Immunsystems ist die ständige Verbesserung seiner Fähigkeiten und Vorgehensweisen im Verlauf seines Lebens. Dies unterscheidet dieses System auch von vielen anderen Bereichen der Biologie. Allerdings wurde diese Erkenntnis bisher nur sehr schwach in künstlichen Immunsystemen genutzt, wodurch viele Aspekte unerforscht blieben [18].

Einen großen Teil der angeführten "zukünftigen" Herausforderungen haben die Entwickler um Aickelin im Danger-Project versucht zu lösen. In einigen Punkten sind ihnen dabei revolutionäre Ansätze gelungen, was der Entwicklung künstlicher Immunsysteme neuen Schwung gegeben hat. Aufgrund dieser zukunftsweisenden Ergebnisse und erfolgreicher Erstversuche in der Robotik wird ein Teil des Danger-Projects, der dendritische Zell-Algorithmus, als Grundlage der vorliegenden Arbeit dienen und im folgenden Kapitel detailliert behandelt.



## Dendritischer Zell-Algorithmus

Vor einigen Jahrzehnten haben diverse Forscher die Idee zu einem künstlichen Immunsystem veröffentlicht. Nach ersten vielversprechenden Erfolgen wurden sehr hohe Erwartungen in diesen Forschungsbereich gesteckt, die allerdings nur in den seltensten Fällen erfüllt werden konnten. Aus diesem Grund waren künstliche Immunsysteme mit Beginn des neuen Jahrtausends am Scheidepunkt angelangt, was eine interdisziplinäre Forschungsgruppe um Uwe Aickelin [1] dazu veranlasste, neue Inspiration in der Biologie zu suchen.

Aus dieser Zusammenarbeit ging neben einigen neuen richtungsweisenden Veröffentlichungen der dendritische Zell-Algorithmus hervor. Inspiriert von der Funktion der dendritischen Zellen im menschlichen Immunsystem sollte der Algorithmus einige Schwachstellen bisheriger künstlicher Immunsysteme beseitigen. Da die dendritischen Zellen eine tragende Rolle bei der Früherkennung von Eindringlingen im menschlichen Körper spielen, empfand Greensmith dies als passenden Ansatz zur Weiterentwicklung künstlicher Abwehrsysteme. Besonders die revolutionäre Kommunikation der Zellen mit dem umliegenden Gewebe mittels Signalen erschien ihr vielversprechend, was sie dazu veranlasste 2007 in ihrer Doktorarbeit [12] einen Algorithmus begründet auf diesen Inspirationen zu veröffentlichen.

## 5.1 Idee

Die Aufgabe dendritischer Zellen im menschlichen Immunsystem ist die Erkennung von Angreifern und die daraus resultierende Aktivierung des adaptiven Immunsystems. Außerdem sollen sie dabei in der Lage sein, Signale von sterbenden Zellen zu empfangen und auszuwerten. Diese tragende Rolle in der Darstellung des Übergangs von angeborener zu adaptiver Immunantwort zeigt die Bedeutung der dendritischen Zellen in biologischer Hinsicht und damit indirekt das Potenzial dieses Ansatzes. Diese Verwendung von Informationen, die vom umgebenden Gewebe zur Verfügung gestellt wurde, stellt einen neuen Ausgangspunkt in der Erkennung von Schädlingen im menschlichen als auch bei künstlichen Immunsystemen dar.

Für diese neue Herangehensweise wurden im Laufe des Danger Projects [1] grundlegende Abstraktionsmodelle beschrieben und entwickelt. Greensmith [12] entwickelte einen Algorithmus, der ähnliche, abstrahierte Aufgaben in Computersystemen erfüllen soll. Dabei werden Antigene ebenso wie in der Biologie als Muster verwendet, welche es ermöglichen, einen Eindringling oder Zustand eindeutig zu identifizieren. Die Signale hingegen werden von verschiedenen Faktoren des Systems beeinflusst, wie beispielsweise CPU-Auslastung, und lassen nach ihrer Auswertung auf das aktuelle Gefahrenpotenzial im System schließen [12].

Wird im System ein neues Muster erkannt, kann der Algorithmus mittels entsprechender Signale eine Einschätzung abgeben, nach welcher Wahrscheinlichkeit das Muster einem Fehler oder Eindringling zuzuordnen ist. Dadurch sollen die wahren positiven Bewertungen gesteigert und vor allem die falschen positiven Einschätzungen von Angreifern gesenkt werden, was bei der negativen Selektion ein ständiges Problem darstellte [12].

Bei der Veröffentlichung des Algorithmus wurde das Augenmerk der Anwendung hauptsächlich auf die Computer Security gelegt. Vor allem beim Port-Scanning wurde der Algorithmus schon sehr erfolgreich eingesetzt [15] sowie beim Vergleich großer Datenmengen auf selbstorganisierenden Karten. Des Weiteren konnte der dendritische Zell-Algorithmus erfolgreich bei der Erkennung von botnets [2] im Internet und Überläufen beim *Schedule* von Prozessen verwendet werden. Außerdem konnte Oates u.a. in seiner Arbeit [27] zeigen, dass der Algorithmus in der Robotik vielversprechendes Potenzial aufweist [13].

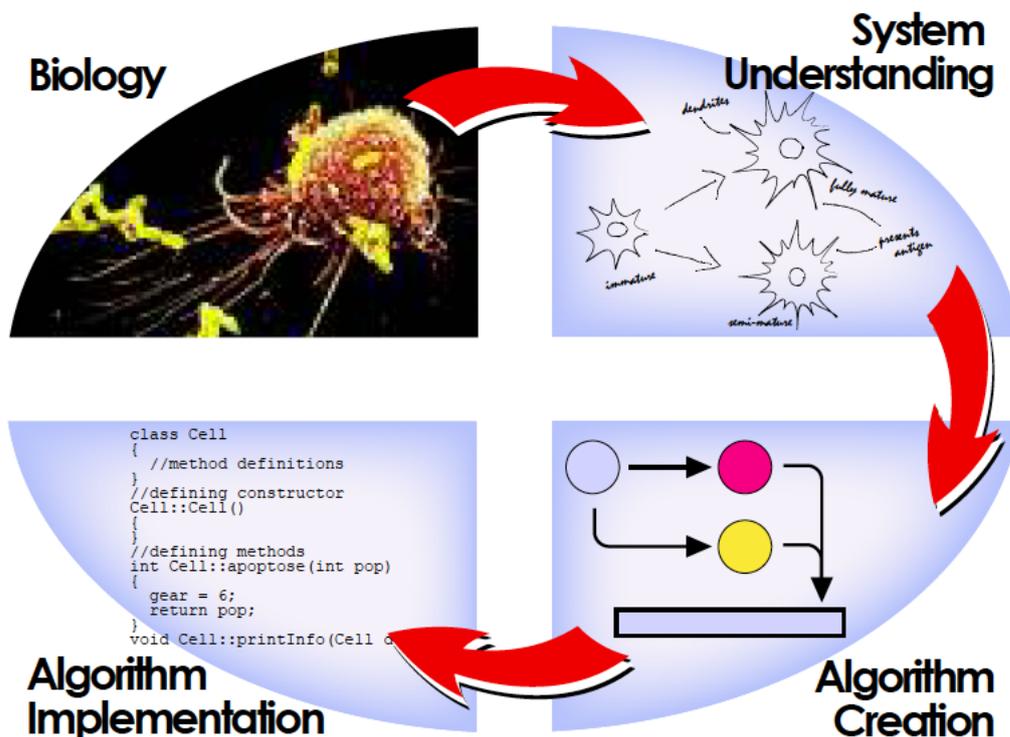


Abbildung 5.1: Verlauf der Abstraktion in [13]

## 5.2 Abstraktion

Die übliche Vorgehensweise bei der Entwicklung eines Algorithmus eines künstlichen Immunsystems war die Verwendung von bestehenden theoretischen Grundlagen und Modellen aus der Biologie. Greensmith stand am Anfang ihrer Arbeit vor dem Problem, dass diese Modelle für ihre Idee noch nicht existierten und dadurch erst entwickelt werden mussten. Dieser Übergang von der Biologie in eine abstrakte Welt wird als einer der kritischsten Punkte in der Entwicklung des dendritischen Zell-Algorithmus angesehen, da hier die Priorität der einzelnen Komponenten und deren Verknüpfungen abgewogen werden müssen. In Abbildung 5.1 wird skizzenhaft das Vorgehen bei der Algorithmusentwicklung aufgezeigt.

Die anfängliche Rechercharbeit wurde stark durch die interdisziplinäre Zusammenarbeit im Rahmen des Danger Projects [1] geprägt, da selbst die Forschung in der Immunologie bezüglich der dendritischen Zellen noch in ihren Anfängen steckte. Innerhalb dieser Forschungsgruppe konnten anschließend die Schlüsselpunkte dieses Modelles festgelegt und definiert werden. Be-

sonders die Analyse von Charakteristika und Funktionsweise der eingebundenen Signale bedurfte nach Greensmith höherer Aufmerksamkeit. Schlussendlich konnte ein umfassendes Modell vorgestellt werden [13].

## Modell

Das beschriebene Modell zeigt die Abgrenzung der Umgebungen, die Unterscheidung der Zellzustände, die Verarbeitung von Antigenen und Signalen sowie die Population der Zellen als markante Schlüsselkomponenten auf.

Eine strikte Trennung der zwei Gebiete, in denen sich die Zellen befinden können, entspricht der zugrunde liegenden Biologie. Dabei sind dendritische Zellen im Gewebe in der Lage, Antigene zu sammeln und Signale zu interpretieren, nach der Reifung haben sie die Aufgabe der Präsentation von Antigenen in den Lymphknoten. Diese Aufteilung der Funktionsweisen nach dem aktuellen Standort der Zelle wird in der Folge auch im Modell und Algorithmus berücksichtigt.

In Abbildung 5.2 werden die drei Zustände dargestellt, welche von den abstrakten Zellen im Modell eingenommen werden können. Der initiale Zustand wird hierbei als unreif<sup>1</sup> bezeichnet, welcher beim Sammeln von Antigenen und Signalen im Gewebe eingenommen wird. Nach der durch Signale beeinflussten Reifung der Zelle nimmt diese entweder einen semi-reifen<sup>2</sup> oder reifen<sup>3</sup> Zustand ein. Diese Zustände stellen folglich das Gefahrenpotenzial ihres Kontextes, welches durch die Signale übermittelt wurde, dar, wobei die Semi-Reife Sicherheit und die Reife Gefahr implizieren lässt. Diese Zustandsübergänge bilden einen Eckstein des dendritischen Zell-Algorithmus.

Bei der Bearbeitung der Antigene gibt es beim dendritischen Zell-Algorithmus eine erhebliche Änderung im Vergleich zu vorhergehenden künstlichen Immunsystemen, da hierbei durch den Vergleich von Antigenen keine Immunantwort ausgelöst wird. Die Antigene dienen im konkreten Fall ausschließlich der Klassifizierung von Einheiten, wobei jede unreife dendritische Zelle eine Teilmenge von Antigenen sammelt und diese ähnlich einer Stichprobe zum späteren Vergleich speichert.

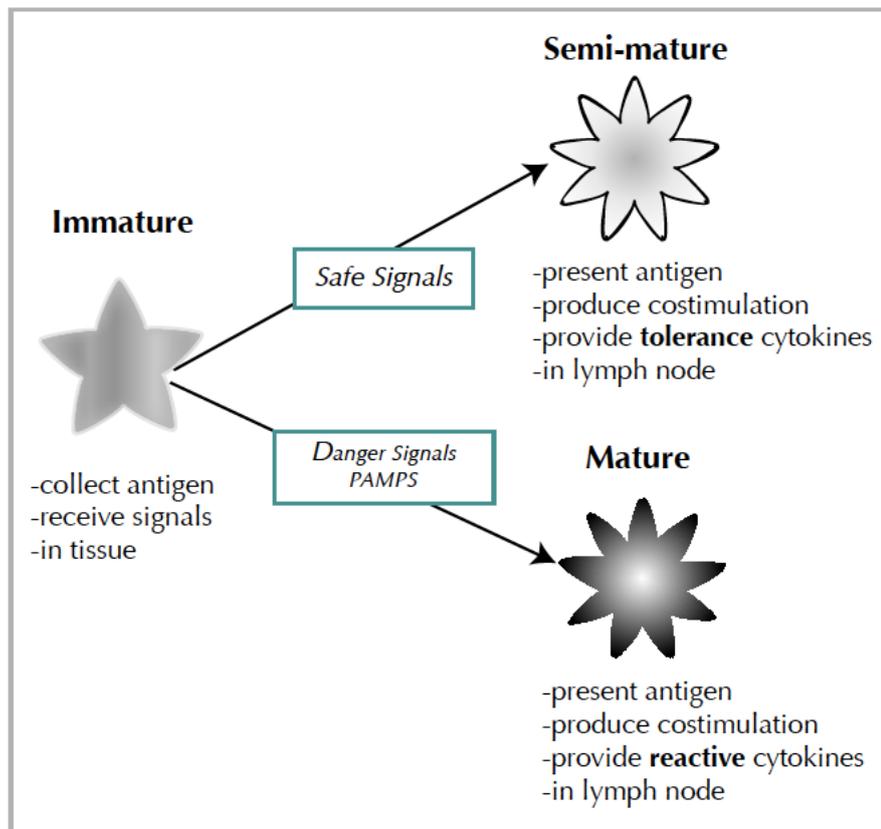
Die Verarbeitung der Signale bildet ein weiteres Kernelement des Algorithmus. Dabei handelt es

---

<sup>1</sup>engl. immature

<sup>2</sup>engl. semi-mature

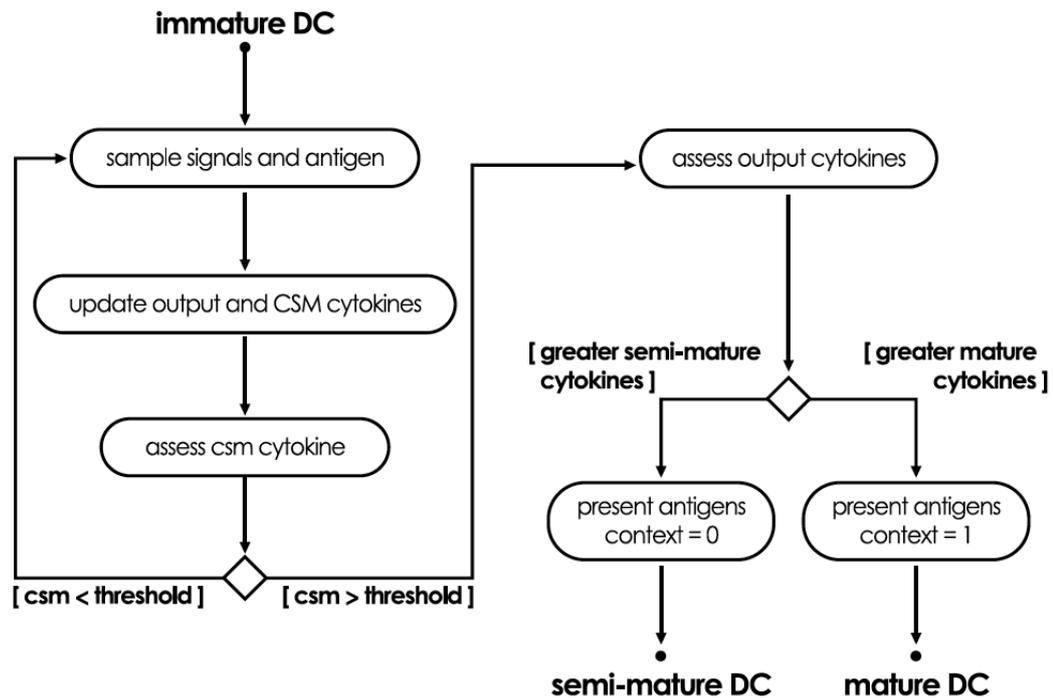
<sup>3</sup>engl. mature



**Abbildung 5.2:** Modell der Transformation von dendritischen Zellen und der Einfluss entsprechender Signale [13]

sich um ein komplexes Zusammenspiel aus Eingangssignalen, welche vom System an den Algorithmus übergeben werden und den Ausgangssignalen, welche der Algorithmus selbst erzeugt. Eine detaillierte Beschreibung der Signale, der Verbindungen untereinander und der Wirkung auf andere Bestandteile im System wird im folgenden Abschnitt angeführt.

Dendritische Zellen treten in der Natur meist in einer Population auf, was eine Reihe von Vorteilen mit sich bringt und deshalb auch in das abstrakte Modell übernommen wurde. Jede Zelle soll unterschiedlichste Antigen-Stichproben sammeln, wodurch eine Vielfalt von Variationen in der Population entsteht. Zur Aktivierung einer Immunantwort für einen spezifischen Antigentyp benötigt es allerdings mehrerer anfordernder Zellen, dadurch entsteht in Folge eine Fehlertoleranz, welche falsche positive Klassifizierungen vermeiden soll. Diese Fehlertoleranz und Robustheit findet sich auch im abschließenden Algorithmus (Abbildung 5.3) wieder [13].



**Abbildung 5.3:** Aktivitätsdiagramm der Entwicklung einer dendritischen Zelle [13]

## Signale

Im biologischen Immunsystem beeinflussen verschiedene Konzentrationen von Signalstoffen das Verhalten der dendritischen Zellen und diese produzieren wiederum neue Signalstoffe. Im abstrakten Modell werden diese in Eingangs- und Ausgangssignale unterteilt.

Im konkreten Modell wurden als Ausgangssignale drei verschiedene Typen berücksichtigt: Interleukin-10, Interleukin-12 und CD80/86. Wobei Letzteres Moleküle darstellt, welche den Reifegrad einer Zelle beschreiben, welcher in der Biologie durch einen komplexen Mechanismus der Rezeptorproduktion abgewickelt wird. Wogegen im Modell nicht nur der Name, kostimulatorische Moleküle (CSM)<sup>4</sup>, sondern auch die Funktionsweise stark vereinfacht wurde. Hierbei stellt CSM die Wahrscheinlichkeit des Abschlusses des Reifeprozesses innerhalb einer dendritischen Zelle dar. Aus den Signalen zur Aktivierung der Immunantwort wurden lediglich zwei Signalstoffe ins Modell einbezogen und abgeändert. Interleukin-10 ist nicht länger ein Signalstoff, sondern wird im Modell durch den reifen Zustand der dendritischen Zelle verwirklicht. Dassel-

<sup>4</sup>englisch: costimulatory molecule

**Tabelle 5.1:** Abstraktion der Signale

Biologie	Abstraktes Modell
PAMP	PAMP
Nekrotische Produkte	Gefahrensignal
Apoptotische Zytokine	Sicherheitssignal
CD80/86	CSM
IL-10	Semi-reifer Zellzustand
IL-12	Reifer Zellzustand

be Konzept wird beim Interleukin-12 verfolgt, welches die Form des semi-reifen Zellzustandes einnimmt.

Beeinflusst werden diese Zellzustände nun vom umgebenden Gewebe, welches seinerseits Signale erzeugt, die sogenannten Eingangssignale. Stirbt eine Gewebezelle in einem Organismus einen unerwarteten Tod, so sendet diese Gefahrensignale aus. Im Detail wurden diese Vorgänge im Kapitel 3.5 erläutert. Durch die höhere Konzentration von Gefahrensignalen werden bei dendritischen Zellen das Interleukin-12 und das CD80/86 erhöht, wobei im abstrakten Modell analog dazu die Ausgangssignale CSM sowie die Tendenz zum voll reifen Zellzustand gesteigert werden. Allerdings besitzen die Gefahrensignale im Vergleich zu anderen Eingangssignalen nur geringen Einfluss auf die Entwicklung der dendritischen Zellen.

Das Gegenstück zu den Gefahrensignalen stellen sowohl in der Biologie als auch im Modell Sicherheitssignale dar, welche ebenfalls den Reifegrad (CSM) der dendritischen Zelle erhöhen und dazu die Tendenz zum semi-reifen Zellzustand steigern. Zusätzlich wird im biologischen Immunsystem die Produktion des Interleukin-12 durch Sicherheitssignale gedrosselt, was ebenfalls im entwickelten Modell Beachtung findet.

Zwischen den beiden Extremen Gefahren- und Sicherheitssignalen liegt im Modell ein weiterer Signaltyp, die *pathogen-associated molecular patterns* (im folgenden kurz PAMP-Signale genannt). Diese Signalart ist in der Lage, das CSM rapide zu steigern, und zieht dabei die dendritische Zelle stark in die Richtung des voll reifen Zustandes, wobei damit die Aufgabe der Warnung bei "Abnormalitäten" im System erfüllt wird [13].

## Antigene

Neben den Signalen stellen die Antigene die zweite Säule im folgenden Algorithmus dar. Allerdings hat sich ihre Rolle und Aufgabe im Vergleich zu bisherigen künstlichen Immunsystemen deutlich geändert. Antigene dienen hierbei ausschließlich als Muster zur Klassifizierung von Angreifern oder Fehlern und beinhalten keinerlei Wertung, wie es bislang der Fall war. Erst die Signale und deren Konzentration geben einem Antigen eine Wertigkeit bei seinem Auftreten im System. Die vollständige Beschreibung des Algorithmus wird im folgenden Abschnitt angeführt [13].

### 5.3 Algorithmus

Im Gesamtsystem eines künstlichen Immunsystems nach Greensmith nimmt der dendritische Zell-Algorithmus eine zentrale Rolle ein, da ihm die Aufgabe der Beurteilung von Antigenen obliegt. Dabei werden ihm, neben den gesammelten Antigenen, Eingangssignale des untersuchten Systems zugeführt. Hierbei handelt es sich trotzdem um keinen Klassifizierungsalgorithmus, da das Ergebnis seiner Einschätzung bezüglich der Antigene nicht aussagt, ob diese anormal sind. Der dendritische Zell-Algorithmus geht noch einen Schritt weiter und liefert einen umfassenderen Informationsgehalt, indem er beschreibt, wie stark ein Antigen vom “normalen” Zustand abweicht.

Am Beginn einer jeden Bewertung eines Antigens steht dessen Einführung in eine Population von dendritischen Zellen, jede von ihnen verkörpert dadurch ein eigenständiges Objekt. Das Antigen wird dabei von jeder Zelle einzeln bewertet und abgespeichert, was im folgenden Codeausschnitt beschrieben wird.

**Listing 5.1:** Pseudocode einer dendritischen Zelle [13]

```
input : antigen a and signalmatrix S
output: cell with antigens and appraisal

initialise cell;
while CSM < boundary of migration do
    save a in cell;
    foreach s in S do
```

```

    s of the cell = s ;
  end
  calculate semi-mature , mature and CSM;
end

put cell to lymph node ;

if semi-mature > mature then
  cell appraisal is assigned as safe ;
else
  cell appraisal is assigned as danger ;
end

```

Nach der Initialisierung der Zelle sammelt sie so lange Antigene, bis ihr CSM-Ausgangssignal die festgelegte Migrationsgrenze überschreitet. Während dieser Sammlung werden kontinuierlich die weiteren Ausgangssignale Semi-reife und Reife der Zelle mittels erhaltener Eingangssignale adaptiert. Erreicht das CSM die Migrationsgrenze, wird die Zelle in den abstrakten Lymphknoten verschoben und als sicher oder gefährlich eingestuft. Die gesammelten Antigene erhalten somit eine indirekte Bewertung durch die Zelle, in der sie erfasst sind. Anschließend werden die Aufgaben der Zelle auf die Präsentation der enthaltenen Antigene beschränkt.

Verwendung finden diese reifen Zellen bei der Berechnung des Mittelwertes des Kontextes für einen Antigenotyp (MCAV)<sup>5</sup>, was die abschließende Einschätzung der Antigene darstellt. Dabei werden ausschließlich dendritische Zellen, welche sich im Lymphknoten aufhalten berücksichtigt, was wiederum die Menge der Antigene auf diesen Bereich begrenzt. Konkret werden die Bewertungen nach folgendem Muster abgearbeitet.

**Listing 5.2:** Pseudocode zur Berechnung des MCAV-Koeffizienten [13]

```

input: Antigens A with context
output: MCAV coefficient per antigen type

foreach a in A do
  increment count for antigen type of a ;
  if context of a is danger then

```

---

<sup>5</sup>englisch: mean value of context per antigen type

```

    increment danger count for antigen type of a;
  end
end

foreach antigen types do
  MCAV of antigen type = danger count / count;
end

```

Für jedes Antigen werden dessen gesamte Vorkommen im Lymphknoten sowie die Vorkommen in gefährlichen Zellen aufsummiert. Die Anzahl der gefährlichen Vorkommen gebrochen durch die gesamten Vorkommen bilden als Ergebnis, den MCAV-Koeffizienten, und somit die Bewertung des Antigens. Der MCAV-Koeffizient besitzt somit einen Wertebereich von 0 bis 1, wobei 0 sicheren oder normalen Antigenen als Einschätzung dient und mit steigendem Koeffizienten die Anomalität des Antigens steigt.

Der innere Ablauf einer Zelle sowie die Berechnung des MCAV-Koeffizienten bildet Subroutinen im Gesamtablauf des dendritischen Zell-Algorithmus und fließen daher folgendermaßen in diesen mit ein.

**Listing 5.3:** Pseudocode des dendritischen Zell-Algorithmus [13]

Parameter:

```

  I = Anzahl der Eingangssignale pro Signaltyp;
  J = Anzahl der Typen fuer Eingangssignale;
  L = Anzahl der Algorithmus-Durchlaeufer;
  M = Populationsanzahl;
  N = max. Anzahl der Antigene in einer Zelle;
  P = Anzahl der Ausgangssignale einer Zelle;
  Q = Anzahl der Antigene welche pro Zelle und Durchlauf entnommen werden;
  A = alle bekannten Antigene;
  S = Signalmatrix;

```

**input:** S and A

**output:** antigen with appraisal

```

create dendritic cells;

```

```

initialise parameters {I, J, L, M, N, P, Q};
while l < L do
  update A and S;
  foreach m in M do
    foreach n in Q do
      cell m takes n antigen from A;
    end

    foreach i in I and j in J do
      signal ij of the cell m = signal ij of S;
    end

    foreach n in N do
      cell m processes his n antigen;
    end

    foreach p in P do
      compute p output signal of the cell m;
    end

    if CSM > boundary of migration then
      cell m removed from population;
      cell m migrate, print antigen and context;
      cell m reset antigens and signals;
    end
  end
  l++;
end

analyse antigen and calculate MCAV;

```

Die Anzahl der Sammeliterationen wird als Parameter L im Vorfeld der Abarbeitung definiert. Nach einer Aktualisierung der Antigene A sowie Signalmatrix S kommt es zum bereits beschriebenen inneren Ablauf einer jeden dendritischen Zelle der Population M, wobei jede reife und somit verschobene Zelle in der Population ersetzt wird. Nach der Beendigung der Sammeli-

terationen kommt es zur Berechnung des MCAV-Koeffizienten und damit zur Einschätzung der Antigene, was den Abschluss des Verfahrens darstellt.

## Vorbereitung und Umsetzung

Für die Entwicklung eines künstlichen Immunsystems und dessen Implementierung werden in diesem Kapitel weitere Voraussetzungen geschaffen. Im ersten Schritt werden dabei die zu erkennenden Fehler anhand von Klassen festgelegt. Dadurch werden neben den verschiedenen Methoden künstlicher Immunsysteme auch im Bereich des untersuchten Robotersystems klare Grenzen gezogen. Diese gesetzten Rahmenbedingungen sollen das Ausmaß der folgenden Implementierung und Evaluierung begrenzen und die konkreten Arbeitsschritte erleichtern.

Die Basis der Umsetzung bildet das bereits vorgestellte Framework des NAO-Roboters im Abschnitt 2.2, welches eingehend analysiert wird. Das Hauptaugenmerk liegt dabei auf der Identifikation einer Schnittstelle, durch welche ein ausreichender Datenfluss zum künstlichen Roboterimmunsystem (im folgenden kurz “KRIS” genannt) erfolgen wird. Außerdem wird dabei der Bereich der Informationsbeschaffung begrenzt, um den grundlegenden Charakter dieser Arbeit nicht zu sprengen.

Parallel dazu wird ein Konzept zur Integration des dendritischen Zell-Algorithmus erarbeitet, wobei eine klare Grenze zum bestehenden System gezogen werden muss, um dieses im Verlauf der Analyse nicht zu beeinflussen. Die Arbeit “The implementation of a novel, bio-inspired, robotic security system” [28] von Robert Oates bildet das Fundament für die Implementierung des dendritischen Zell-Algorithmus, welcher nach einigen Adaptionen alle entsprechenden Anforderungen erfüllt.

Zur weiteren Abgrenzung dieser Arbeit dient der folgende Schritt, welcher sich mit der Klassifizierung der Fehler sowie Ursachen und Folgen im Robotersystem beschäftigt. Die Generierung von Antigenen, welche diese Fehler als eindeutige Zustände identifizieren lassen und die Definition der Signale, die diese Anomalien als solche bewerten, bilden anschließend den Kern der Arbeit, wobei besonders die Feinabstimmung der Signale eine erhebliche Herausforderung darstellt.

## **6.1 Fehlerklassen**

Ein NAO stellt während eines Fußballspiels, wie ein Großteil aller Roboter, konzeptuell einen Agenten dar, welcher zu selbstständiger Interaktion mit seiner konkreten Umgebung in der Lage ist. Dabei wird diese Interaktion in mehrere Schritte aufgeteilt. Den ersten Schritt bildet dabei die Wahrnehmung des Roboters, welcher Gegenstände wie Ball und Tor erkennen soll. Die Wahrnehmung dient damit der Beschaffung von Informationen, welche im zweiten Schritt analysiert werden. Beim Fußballspielen werden dabei im Roboter die unterschiedlichsten Daten errechnet, wie beispielsweise die Lokalisierung, welche alle der Verhaltenskontrolle oder auch der künstlichen Intelligenz zugeführt werden. Dieses bereits vorgestellte Modul trifft aufgrund der erhaltenen Daten eine Entscheidung, wodurch der letzte Schritt der Roboterinteraktion eingeleitet wird, die Handlung. Der Roboter versucht durch die Handlung seine Umwelt zu verändern, beispielsweise das Fußballspiel zu gewinnen.

In diesem Interaktionskreislauf finden sich ungeahnte Möglichkeiten für Fehler, welche teils weitreichende Folgen für das Gesamtsystem mit sich bringen. Einen besonders prekären Bereich dafür stellt die Wahrnehmung dar, da sie den Beginn des beschriebenen Kreislaufes darstellt. Aber auch bei Abweichungen oder Ausfällen im Bereich der Handlung kann ein Robotersystem schnell unbrauchbar werden.

### **Positionsänderung**

Die Anforderung für eine Positionsänderung wird im Roboter von der künstlichen Intelligenz erstellt und an die entsprechenden Module weitergeleitet, wodurch sich der Roboter im Normalfall in Bewegung setzt. Diese Positionsänderung wird in der Wahrnehmung registriert und wiederum

der künstlichen Intelligenz zugeführt. Im Fehlerfall wird dieser Kreislauf unterbrochen, wodurch eine Positionsänderung oder deren Erkennung unmöglich wird.

Die Ursachen eines solchen Fehlerfalls sind äußerst vielfältig und können im Bereich der Hardware in Motoren, Zahnrädern oder anderen Bauteilen auftreten. Kameras oder Beschleunigungssensoren können keine oder Fehlinformationen liefern. Ebenso komplex wird das Problem, falls Fehler in der Lokalisierung auftreten. Außerdem können Kombinationen verschiedenster Fehlerarten zu diesem Problem führen.

Trotz dieser breiten Fächerung der möglichen Ursachen bleibt das Ergebnis für den Roboter dasselbe, da er ausschließlich die Differenz zwischen Anforderung und Resultat erfassen kann. Dieser Fehlerfall, dass eine Positionsänderung vom System angefordert wird und in weiterer Folge allerdings nicht erkennbar ist, soll von KRIS wahrgenommen und entsprechend verarbeitet werden.

## **Gegenstandserkennung**

Die Wahrnehmung von Fehlerfällen bei der Gegenstandserkennung stellt sich ähnlich jener der Positionsänderung dar. Dabei wird ebenfalls eine Art von Kontrollkreislauf aufgebaut, welcher durch eine Anforderung beginnt und durch die entsprechende Wahrnehmung geschlossen wird. Allerdings liegt die Anforderung hierbei nicht konkret vor und bedarf einer geringen Abstraktion. Der einzige erkennbare Grund für den Roboter seinen Kopf zu bewegen, ist die Hoffnung, dadurch einen bekannten Gegenstand zu sehen. Dadurch wurde in den folgenden Versuchen die Bewegung des Kopfes als Anforderung zum "Suchen eines bekannten Gegenstandes" interpretiert. Daraus resultiert im Normalfall das Wahrnehmen des Balles oder eines Tores.

Bei einer Störung der Gegenstandserkennung werden diese Spielelemente trotz entsprechender Anforderung für längere Zeit nicht gesehen. Dabei können menschliches Versagen bei der Kalibrierung, schlechte Lichtverhältnisse, Kameraausfall oder andere Hardwarefehler, sowie eine Reihe von Softwareabweichungen als Ursache für diesen Fehlerfall verantwortlich sein.

Allerdings ist der Fußballspieler, der keinen Ball und keine Tore sieht, nutzlos für das Spiel und das Ergebnis bleibt dasselbe. Für den Roboter zeigt sich auch hier eine auftretende Differenz zwischen Anforderung und Ergebnis, welche vom dendritischen Zell-Algorithmus als Fehlerfall erkannt werden soll.

## 6.2 Daten aus dem Robotersystem

In Anlehnung an die Terminologie biologischer Immunsysteme wird die zu untersuchende Struktur als Gewebe bezeichnet. Im konkreten Fall wird der NAO Roboter zusammen mit seinem Framework als solches aufgefasst und übernimmt demzufolge die Aufgaben seines biologischen Vorbildes. Neben der Erzeugung und Bereitstellung der benötigten Signale wird das künstliche Gewebe des Weiteren die Generierung der Antigene verantworten.

In Bezug auf die definierten Fehlermodelle bedarf es einer geeigneten Schnittstelle im System, welche auf die benötigten Informationen zugreifen kann. Dennoch sollte sich der Zugriffspunkt am Rand des Frameworks, wie beispielsweise in Komponenten der Kommunikation, befinden, um die Beeinflussung der Datenentnahme auf das Systemverhalten möglichst gering zu halten.

### Informationsfluss

Das verwendete Framework besteht in erster Linie aus drei Prozessen, wobei der Bewegungsprozess sowie der Wahrnehmungsprozess die Steuerung des Roboters übernehmen und der Prozess des Debug die Kommunikation mit einem externen Computer ermöglicht. Die Aufgaben der einzelnen Abläufe sind, wie bereits in Abschnitt 2.2 erwähnt, strikt voneinander getrennt. Der Prozess der Bewegung regelt den Zugriff auf die einzelnen Motoren und bewegt damit den Roboter. Außerdem erfasst er die Daten aller Sensoren mit Ausnahme der Kamera. Die so gesammelten Daten werden dem Wahrnehmungsprozess zur Verfügung gestellt, welcher diese durch Informationen der Bildverarbeitung aus der Kamera vervollständigt. Durch die Zusammenführung aller Daten wird das Modul der Modellierung in die Lage versetzt, ein umfassendes Weltbild für den Roboter zu erstellen.

Alle gesammelten sowie berechneten Informationen von Sensoren, Bewegung, Bildverarbeitung und dem Modell der Umgebung werden schlussendlich der Verhaltenskontrolle zugeführt, welche abermals weitere Daten, wie den aktuellen Zustand der künstlichen Intelligenz, hinzufügt. Nach entsprechender Auswertung aller Daten wird in diesem Modul eine Entscheidung über das weitere Verhalten des Roboters getroffen. Die Entscheidung wird anschließend umgehend allen betroffenen Komponenten mitgeteilt, wobei dies in erster Linie die Module des Bewegungsprozesses betrifft.

Der Zusammenfluss von Informationen im Framework lässt somit eine ständige Steigerung erkennen und erreicht mit der Entscheidungsfindung im Modul der Verhaltenskontrolle seinen Abschluss. Zur späteren Abgrenzung der vorliegenden Fehlermodelle vom Normalzustand des Gewebes benötigt KRIS die Gesamtheit der erzeugten Informationen, ähnlich der Komponente zur Entscheidungsfindung. Folglich liegt der ideale Punkt zur Datenerfassung von KRIS im Bereich der Verhaltenskontrolle.

## **Verhaltenskontrolle**

Die Beschreibung des Verhaltens des Roboters erfolgt im verwendeten Framework mittels einer domainspezifischen Sprache: Xabsl. Xabsl wurde 2003 von M. Löttsch, J. Bach, H.-D. Burkhard und M. Jünger in der Arbeit “Designing Agent Behavior with the Extensible Agent Behavior Specification Language XABSL” [24] vorgestellt. Sie bietet die Möglichkeit, das Verhalten des Roboters anhand hierarchisch angeordneter Zustandsmaschinen darzustellen, was eine Vielzahl von Vorteilen mit sich bringt. Die geringe Einarbeitungszeit, die gute Lesbarkeit sowie die Möglichkeit der äußerst effizienten Programmierung werden durch die parallele Abarbeitung der einzelnen Zustandsmaschinen vervollständigt.

Die Sprache Xabsl besteht aus einfachen Symbolen, welche der Ein- und Ausgabe dienen. Diese werden in c++- und Header-Dateien definiert und bilden damit die Schnittstelle zum restlichen Framework. Die Strukturierung dieser Symbole erfolgt mittels *options*, *states* und *decisions*, wie dies beispielhaft im folgenden dargestellt wird.

Das Verhalten des Roboters wird so aufgebaut, dass mehrere *options* gleichzeitig aktiv sein können und innerhalb einer *option* muss genau ein Zustand (*state*) angenommen werden. Durch die *decisions* können Zustände gewechselt oder beibehalten werden, was den ersten Teil eines Zustandes darstellt. Der zweite Teil *action* beschreibt die aktuelle Handlung, welche Bewegungen, Berechnungen oder Zuweisungen sein können. Eine ausführliche Beschreibung von Xabsl wird in [24] oder [29] angeboten.

Die Verarbeitung dieser Verhaltensstrukturen erfolgt in der Xabslengine, welche vollständig in das Framework integriert wurde. Dabei werden sämtliche Symbole erfasst, mögliche Zustandsübergänge hierarchisch abgearbeitet und entsprechende Anforderungen an die restlichen Module erstellt. Dieser äußerst komplexe und ebenso sensible Prozess bildet die “Schaltzentrale” des Roboters und wird aufgrund seiner zentralen Rolle nicht als Datenschnittstelle von KRIS

```

option example_option {
    initial state first_state
    {
        decision
        {
            if( boolean_expression )
                goto second_state;
            else
                stay;
        }

        action
        {
            output_symbol = input_symbol * 3;
        }
    }
    ...
}

```

**Abbildung 6.1:** Codeauszug aus Xabsl-Beispiel

benutzt.

### **XabslInfo**

Die Grenzen des Wahrnehmungsprozesses bilden die beiden Schnittstellen zu den anderen Prozessen. Deshalb wird die Interprozesskommunikation als mögliche Datenschnittstelle von KRIS analysiert. Dabei wird allerdings nur jene Kommunikation untersucht, welche vom Wahrnehmungsprozess ausgeht, da nur dieser über vollständige Informationen verfügt. Die Interprozesskommunikation mit dem Bewegungsprozess scheidet dabei im Vorherein als möglicher Ansatzpunkt aus, da nur entsprechende Anforderungen an die Bewegung mitgeteilt werden. Hingegen werden bei der Kommunikation mit dem Debug-Prozess alle bekannten Informationen der Verhaltenskontrolle übermittelt, wodurch eine vollständige Auswertung durch KRIS ermöglicht wird.

Der Debugprozess erhält über das Blackboard von nahezu allen Modulen Informationen über ihren aktuellen Zustand. Bei der Verhaltenskontrolle erfolgt diese Mitteilung mithilfe der *Re-*

```

struct ActiveOption
{
    std::string name,
                durationOfActivation,
                activeState,
                durationOfStateActivation;

    int depth;
    std::list<NameValue> parameters;
};

```

**Abbildung 6.2:** Struktur einer aktiven Option

*presentation* XabsInfo, welche neben allen bekannten Symbolen und allen aktiven Optionen und Zuständen auch eine Reihe von möglichen Parametern enthält. Das Objekt selbst wird im laufenden System kontinuierlich von der Verhaltenskontrolle aktualisiert, was eine hervorragende Grundlage für die Untersuchungen von KRIS bildet. Die bereitgestellten Informationen in Form von XabsInfo finden im aktuellen System im Simulator Anwendung, wobei sie hierbei zur Visualisierung des aktuellen KI-Zustandes dienen.

Für die weitere Arbeit sind zwei Datenstrukturen der Klasse XabsInfo von tragender Rolle. Der aktuelle Zustand der künstlichen Intelligenz wird dabei in Form einer Liste von aktiven Optionen<sup>1</sup> (Abbildung 6.2) erfasst. Dabei enthält jede Option ihren Namen, ihre Hierarchietiefe, den aktiven Zustand sowie die Dauer der Option und des Zustandes. Außerdem wird eine Möglichkeit für etwaige Parameter bereitgestellt.

Die zweite Datenstruktur von höherer Priorität stellen die Eingabesymbole<sup>2</sup> dar, welche ebenfalls als Liste abgespeichert werden. Die konkrete Implementierung der Eingabesymbole wird in Abbildung 6.2 angeführt. Diese Struktur beinhaltet den Typ und den Namen des Symbols. Sollte das Symbol vom Type “enumerated” sein, werden dessen Werte gesondert im Datentyp “Enumeration” abgelegt, ansonsten werden die Werte als String gespeichert. Außerdem können Parameterobjekte behandelt werden, welche in dieser Arbeit allerdings vernachlässigt werden. Besondere Beachtung sollte hierbei der Standardkonstruktor erhalten, innerhalb welchem die Variable “show” auf wahr gesetzt wird. Nur so werden die Aktualisierungen dieser Symbole nach außen mitgeteilt und verwendet werden.

---

<sup>1</sup>im Code: ActiveOption

<sup>2</sup>im Code: InputSymbol

```

enum SymbolType {decimal , boolean , enumerated };

struct NameValue
{
    std::string name ,
                value ;
};

struct InputSymbol : public NameValue , public ParameterizedObject
{
    bool show ;

    SymbolType type ;
    Enumeration* enumeration ;

    InputSymbol() : show(true) {}
};

```

**Abbildung 6.3:** Struktur der Symbole

Weitere Datenstrukturen, wie beispielsweise Ausgangssymbole<sup>3</sup>, Parameter oder Zeitangaben bezüglich der Informationen, werden im Verlauf dieser Arbeit nicht berücksichtigt, können aber bei möglichen Folgearbeiten ohne größere Änderungen als Informationen für KRIS verwendet werden.

Diese beschriebenen Gegebenheiten der idealen Lage von XabsInfo am Rande des Wahrnehmungsprozesses sowie die vollständig enthaltene Information bieten hervorragende Bedingungen zur Datensammlung für KRIS. Außerdem werden die Informationen kontinuierlich durch das Modul der Verhaltenskontrolle aktualisiert, wodurch ein diesbezüglicher Eingriff in das System vermieden wird. Somit wird der Wahrnehmungsprozess eine vernachlässigbare Beeinträchtigung seiner Aufgaben erfahren, was eine fundamentale Anforderung für die Integration von KRIS darstellt.

---

<sup>3</sup>im Code: OutputSymbol

## 6.3 Migration des Algorithmus

Die gesammelten Informationen des Robotersystems werden mittels des dendritischen Zell-Algorithmus analysiert und eingeschätzt. Dazu wird zunächst die entsprechende Implementierung des Algorithmus vorgestellt. Des Weiteren werden nötige Anpassungen der zugrunde liegenden Bibliothek<sup>4</sup> von Robert Oates an die vorliegenden Anforderungen durchgeführt.

Die Integration des adaptierten Codes in das Robotersystem sowie der Anschluss an die bereits lokalisierte Datenschnittstelle werden in den folgenden Abschnitten im Detail aufgearbeitet.

### Implementierung des Algorithmus

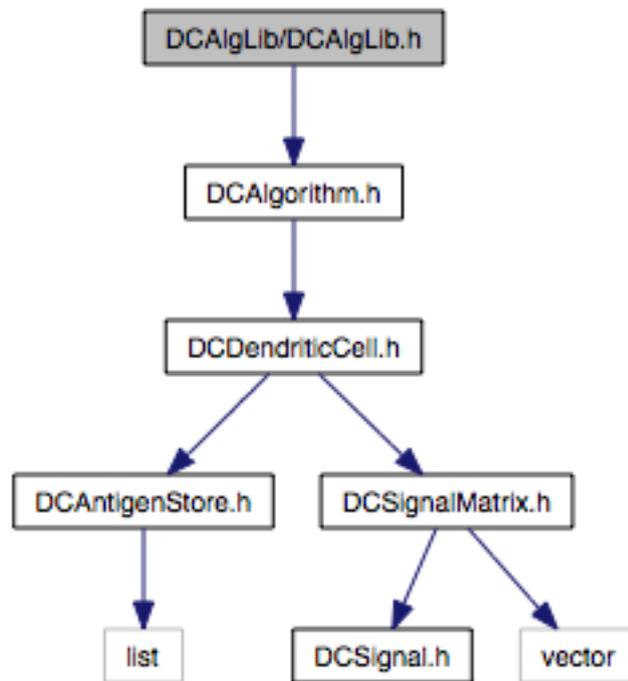
Der Algorithmus selbst wurde in der Programmiersprache c++ implementiert und umfasst sechs Klassen. Die Abbildung 6.4 stellt die Abhängigkeiten der einzelnen Header dar, wobei hierbei der Header "DCAntigen.h" nicht berücksichtigt wurde, da er erst durch diverse Adaptionen in die Struktur mit einfließt.

Durch die Einbindung des DCAAlgLib-Header wird somit ein System in die Lage versetzt, auf den Algorithmus zuzugreifen. Innerhalb des DCAlgorithm wird während des Programmablaufs eine Population mit vorgegebener Größe an DCDendriticCells, also dendritischen Zellen, verwaltet. Diese wiederum besitzen neben einer Speichereinheit für entsprechende Antigene, nämlich DCAntigenStore, eine DCSignalMatrix zur Berechnung und Ablegung ihrer Signale, das DCSignal. Diese Bibliothek ist in der Lage, verschiedenste Datentypen zu verarbeiten. So können beispielsweise einzelne Zeichen, Zahlen oder Zeichenketten als Antigene verwendet werden. Die Signale selbst fließen als Gleitkommazahlen in die Berechnungen ein. Dies gilt sowohl für die Eingangs- als auch für die Ausgangssignale.

Neben der Festlegung des Antigentyps bedarf es zusätzlich einiger Schnittstellen, welche den Datenaustausch zwischen dem Gewebe, also dem untersuchten System und dem Algorithmus regeln. Die Klasse "NaoStateStore" (Abbildung 6.5) stellt eine solche Schnittstelle dar, welche von der Bibliotheksklasse "DCAntigenStore" abgeleitet wird und deren abstrakte Methoden implementiert. Dabei handelt es sich um Methoden eines Stapelspeichers "Push(void\* newData)" und "Pop()" sowie "GetData(int index)", um Antigene anhand ihres Indexes auszulesen. Dieser

---

<sup>4</sup><http://www.cs.nott.ac.uk/~rxo/> zuletzt besucht am 20.03.2012



**Abbildung 6.4:** Aufbau der Algorithmus-Bibliothek

simple Stapelspeicher erfüllt damit die Aufgabe der Antigenverwaltung, welche im konkreten Fall in Form einer Zeichenkette abgelegt werden.

Analog zu den Antigenen wird eine ähnliche Schnittstelle für die behandelten Signale erstellt, nämlich “NaoSignal” (Abbildung 6.5). Innerhalb dieser Klasse werden die abstrakten Methoden “UpdateData()” und “GetValue()” ihrer Elternklasse “DCSignal” implementiert. Außerdem beinhaltet jedes Signal seinen aktuellen Wert sowie seinen Signaltyp. Mit Hilfe der Funktion “UpdateData()” kann der eigene Wert aktualisiert werden kann.

Den Dreh- und Angelpunkt dieses Datenaustausches bildet die Klasse “NaoManager” (Abbildung 6.5), welche die gegebenen Daten aus dem Robotersystem liest, diese interpretiert und abschließend an die definierten Schnittstellen abgibt. Konkret werden dazu genau definierte Zeichenketten gelesen, wobei zunächst zwischen Signalen oder Antigenen unterschieden wird. An-

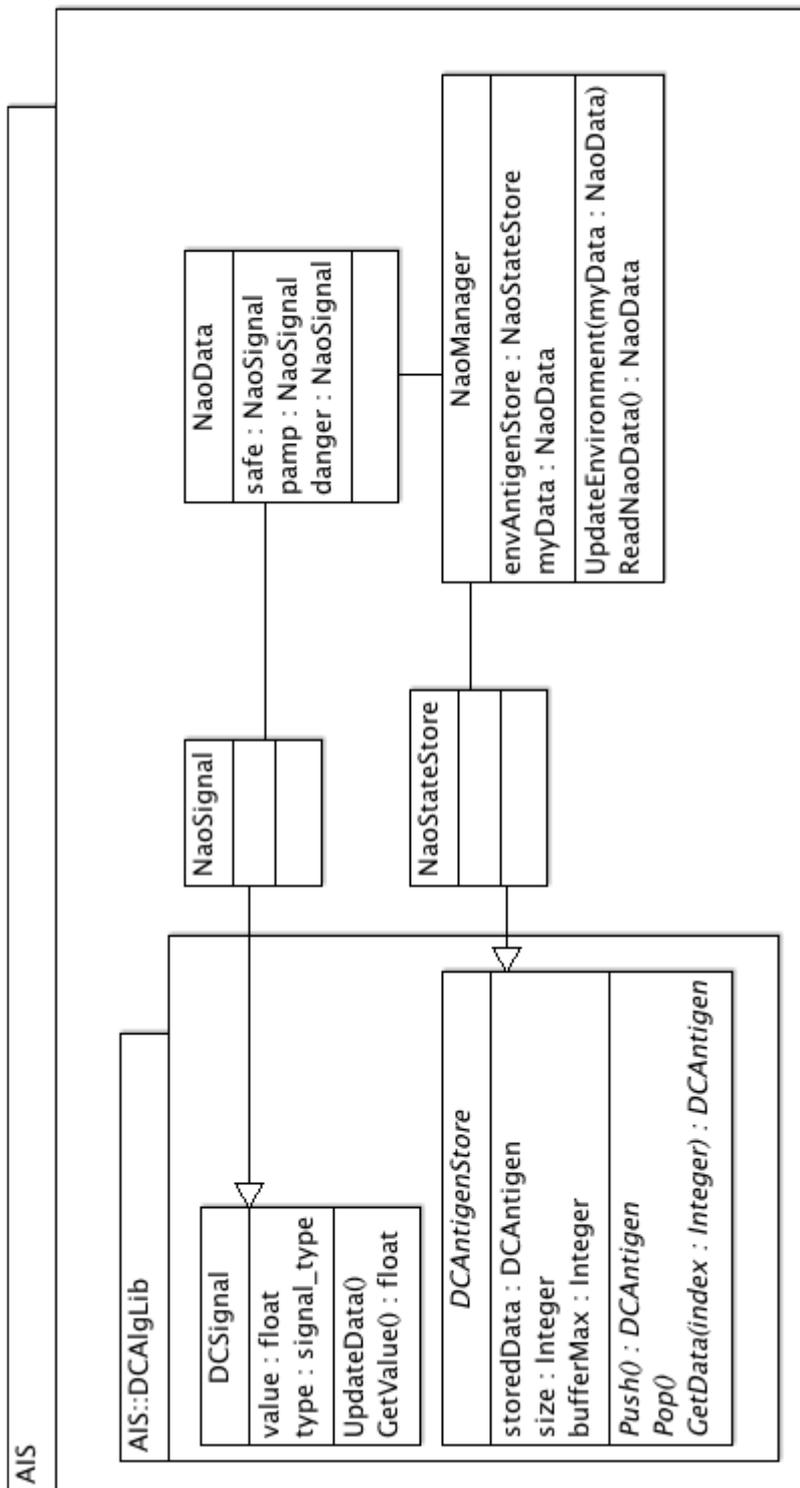


Abbildung 6.5: Schnittstellen für Antigene und Signale aus Nao

schließlich werden die erhaltenen Daten je nach Typ über die Klasse "NaoData" in eine globale Signalmatrix oder in einem Antigen Speicher namens "envAntigenStore" abgelegt. Diese Datensammlung wird zeitlich begrenzt, da das Robotersystem ununterbrochen Daten bereitstellen kann.

Nach der Datenerfassung mittels "NaoManager" folgt die Phase des Trainings, bei der den Zellen der aktuellen Population die gesammelten Daten aus der Signalmatrix und dem Antigen Speicher zugeführt werden. Jedes Objekt der Klasse "DCDendriticCell" holt sich die Antigene aus dem globalen Speicher und adaptiert seine Ausgangssignale entsprechend der vorliegenden Eingangssignale. Sollte nach diesem Vorgang eine Zelle als reif oder semi-reif erkannt werden, scheidet diese aus der Population aus und wird gesondert abgespeichert.

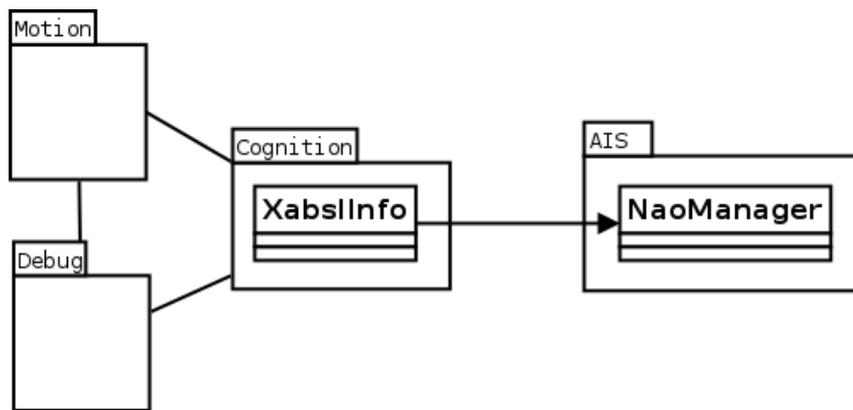
Den Abschluss eines jeden Zyklus bildet die Berechnung des MCAV-Koeffizienten für jedes Antigen anhand der reifen und semi-reifen Zellen. Die Ergebnisse dieser Berechnung werden schlussendlich in einer Datei für die spätere Analyse abgelegt.

## **Verbindung zum System**

Nach der Lokalisierung der Datenschnittstelle im Gewebe und der Umsetzung des dendritischen Zell-Algorithmus bedarf es einer Verbindung der beiden Komponenten, welche einige Anforderungen erfüllen soll. Zunächst wird der Algorithmus selbst in das Framework integriert, worauf im nächsten Schritt ein Kommunikationskanal zwischen beiden Komponenten eingerichtet wird.

Der Algorithmus wird als Teil eines neuen Prozesses, welchen das Framework startet, integriert. Durch diese Lage ist eine ausreichende Abgrenzung zu den Steuerungsprozessen des Roboters gegeben, dennoch ist der Aufbau eines Kommunikationskanals ohne größere Probleme möglich. Dafür wird eine Interprozesskommunikation zwischen dem neuen KRIS-Prozess und jenem der Wahrnehmung eingerichtet.

Der Kommunikationskanal zwischen den beiden Prozessen wird mittels eines gepufferten unidirektionalen Datenstromes, einer Pipe, realisiert (Abbildung 6.6). Dieses einfache Prinzip ermöglicht einen Datenaustausch zwischen zwei Prozessen durch das Öffnen einer eindeutig benannten Datei, wobei ein Prozess schreibend und der andere lesend auf diese zugreift. Im beschriebenen Szenario stellt der Wahrnehmungsprozess den schreibenden Teil dar, welcher die Datei mit benötigten Daten aus dem Robotersystem befüllt. Der KRIS-Prozess liest diese Informationen durch



**Abbildung 6.6:** Verbindung zwischen dem Roboter und dem künstlichem Immunsystem

die Klasse “NaoManager” aus und verarbeitet diese, wie im vorherigen Abschnitt beschrieben wurde.

## 6.4 Antigene und Signale

Durch die Integration des dendritischen Zell-Algorithmus in das Framework des Roboters sowie den Aufbau eines Kommunikationskanals zwischen diesen beiden Komponenten werden alle benötigten Werkzeuge für die durchzuführenden Versuche bereitgelegt. Der letzte Schritt in der Integration von KRIS stellt die Selektion der zu verwendenden Daten sowie deren Definition als Antigen und Signal dar.

Die bereits festgelegte Datenschnittstelle im Robotersystem zeigt dabei die Rahmenbedingungen für die Datenauswahl auf. Es werden somit nur Daten verwendet, welche um die Verhaltenskontrolle entstehen und als die bereits beschriebenen Symbole erfasst werden. Des Weiteren wird der Zustand der künstlichen Intelligenz selbst in Form von Optionen und Zuständen für die Beschreibung von Antigenen und Signalen herangezogen.

Die Erkennung der Fehlerklassen innerhalb des Systems beruht auf dem Prinzip des Agenten-Kreislaufes zwischen Wahrnehmung, Analyse und Entscheidung sowie Handlung. Dabei soll der dendritische Zell-Algorithmus in die Lage versetzt werden, Differenzen zwischen den Anforderungen aus der Verhaltenskontrolle und der tatsächlichen Handlung oder Wahrnehmung zu erkennen und als Fehler zu bewerten.

In den folgenden Abschnitten werden zunächst die Antigene und Signale in allgemeiner Form beschrieben, worauf eine Aufspaltung der Eigenschaften bezüglich der einzelnen Fehlerklassen folgt. Abschließend wird eine Kombination der Fehlerklassen innerhalb des Systems vorgestellt.

### **Anforderung im Antigen**

Die Aufgaben des Antigens bestehen darin, einen Zustand des Systems so darzustellen, dass eine Möglichkeit besteht, zwischen “gesunden” und “kranken” Stadien des Roboters zu unterscheiden. Basierend auf dem Prinzip des Agenten-Kreislaufes stellt eine Differenz zwischen Anforderung und Umsetzung einen Fehlerfall dar. Bei einer Übereinstimmung zwischen Anfrage und Resultat hingegen wird der Zustand als “gesund” bewertet. Aufgrund dessen beinhaltet auch das konstruierte Antigen diese beiden Bausteine.

Der Zustand der künstlichen Intelligenz enthält implizit eine Anforderung an das System. Diese kann von einfacher Struktur sein, wie eine Bewegung, oder komplexer, wie die Suche nach einem bekannten Gegenstand. Aufgrund des hohen Informationsgehaltes dieses Zustandes bildet dieser in codierter Form den ersten Teil des verwendeten Antigens.

Die Codierung des Zustandes erfolgt in mehreren Schritten und beinhaltet schlussendlich alle aktiven Optionen sowie deren aktuellen Zustand. Die Kombination dieser beiden Elemente ist im gesamten Verhalten des Roboters eindeutig, was von der Verarbeitungseinheit des Verhaltens gewährleistet wird. Zur Codierung des Verhaltens wird jede unbekannte Kombination von Option und Zustand zunächst in eine Liste geschrieben. In einem zweiten Schritt werden alle Listenindizes der aktiven Kombinationen in hexadezimaler Form zu einer Zeichenkette zusammengefasst.

Die so erhaltene Zeichenkette beinhaltet in kompakter Form den aktuellen Zustand der Verhaltenskontrolle und damit die impliziten Anforderungen an das restliche Robotersystem. Zur Kenntlichmachung von Fehlern bedarf es allerdings der Differenz zwischen Anforderung und Umsetzung, welche im Folgenden spezifisch für die einzelnen Fehlerklassen an das bestehende Antigen angefügt wird. Dasselbe gilt für die verwendeten Signale, wobei alle drei Typen, “Safe”, “PAMP”<sup>5</sup> und “Danger” speziell auf die einzelnen Fehlerklassen zugeschnitten werden.

---

<sup>5</sup>engl. pathogen-associated molecular patterns

## Positionsänderung

Den nächsten Teil des Antigens bildet, wie bereits angesprochen, die Umsetzung der Anforderung oder deren Resultat. Auf die Anforderung der Bewegung sollte somit eine Positionsänderung erfolgen. Allerdings würde das Anführen einzelner Positionen oder Unterschiede zwischen zweier Positionen zu einer unüberschaubaren Menge an Antigenen führen, da sich kaum zwei gleiche Antigene bilden würden. Aus diesem Grund wird ausschließlich die Information, ob es zu einer Positionsänderung gekommen ist, angehängt.

Die Änderung der Position wird anhand von drei Eingabesymbolen erkannt. Dabei werden die x- und y-Koordinate sowie der Drehwinkel des Roboters mit vorherigen Werten verglichen. Weicht dabei nur einer dieser drei Werte von seinem Vorgänger ab, wird dies als Positionsänderung angesehen.

Die Signale für die konkrete Fehlerklasse können aus dem konstruierten Antigen abgeleitet werden. Die Aufforderung zur Bewegung dient hierbei als Trigger für eine Bewertung der folgenden Antigene, da nur unter der Bedingung der Anforderung eine Positionsänderung von Interesse ist. Dazu wird der Inhalt des Symbols "executed-motion.type" abgefragt, wobei ausschließlich der Wert "motion.type.walk" als PAMP-Signal gewertet wird.

Durch das Auftreten des PAMP-Signals tendiert die aktuelle Population an dendritischen Zellen im künstlichen Immunsystem zu einer Einschätzung auf Gefahr. Dies kann nur durch das Auftreten des Sicherheitssignals (Safe) verhindert werden. In Bezug auf das erstellte Antigen liegt das Sicherheitssignal beim Auftreten einer Positionsänderung im Robotersystem vor. Aufgrund der hemmenden Wirkung des Sicherheitssignals wird die aktuelle Population der Zellen wieder beruhigt.

Genau das Gegenteil ist der Fall, falls keine Positionsänderung erkannt wird, da in diesem Fall neben dem PAMP-Signal auch das Gefahrensignal auftritt und die Tendenz der dendritischen Zellen verstärkt. Dadurch werden die auftretenden Antigene in Zellen abgelegt, welche nach ihrer Reifung auf eine Gefahr ihrer enthaltenen Muster hinweisen. Die Gewichtung der einzelnen Signale beruht auf den Standardwerten aus der Literatur [28].

## **Gegenstandserkennung**

Auf ähnliche Weise erfolgt die Erweiterung des Antigens bezüglich der Fehler bei Gegenstandserkennungen, da auch hierbei die entsprechende Anforderung aus dem codierten Verhaltenszustand kommt. Die folgende Antwort auf die Anforderung der "Suche nach einem bekannten Gegenstand" sollte das Erkennen eines solchen darstellen. Allerdings kann das Finden eines Gegenstandes nicht unmittelbar nach der Anforderung eintreten, was wiederum einiger Vorversuche und Annahmen bedarf.

Als bekannte Gegenstände werden hierbei das gegnerische und das eigene Tor sowie der Ball verwendet. Verschiedene Versuche im Vorfeld haben gezeigt, dass bei normalem Spielverlauf im schlimmsten Fall etwa fünf Sekunden benötigt werden, um eines dieser drei Objekte zu finden. Aus diesem Grund wird dieser heuristische Wert als Toleranzzeit für das Finden von bekannten Gegenständen einbezogen.

Auch bei der Umsetzung der Signale bedarf es einiger Annahmen. So wird beispielsweise die Bewegung des Roboterkopfes als "Suche von bekannten Gegenständen" gewertet, was auch den einzigen rationalen Grund für eine Kopfbewegung darstellt. Folglich gilt diese Bewegung als Bedingung für eine Bewertung durch KRIS und stellt damit das PAMP-Signal dar.

Zur Unterdrückung der Tendenz zur Bewertung als Gefahr dient analog zur vorherigen Fehlerklasse das Erkennen eines bekannten Gegenstandes. Dieses Sicherheitssignal stellt wiederum das Gegenstück zum Gefahrensignal dieser Fehlerklasse dar, welches bei einer erfolglosen Suche nach bekannten Gegenständen im System auftritt.

## **Kombination der Fehlerklassen**

Das Ziel eines künstlichen Immunsystems ist es, mehrere verschiedene Störfälle im System zu erkennen und entsprechend zu reagieren. Diese Anforderung bedarf einer Verschmelzung der Antigene sowie der Signale der vorgestellten Fehlerklassen sowie eine Untersuchung der Beziehungen zwischen den beiden Fehlern. Ansonsten werden weder am Algorithmus noch an der Verbindung zum Robotersystem Änderungen vorgenommen. Es bleibt ein Prozess mit einer Zellpopulation für die Einschätzung aller Systemzustände verantwortlich.

Die Abgrenzung der beiden Fehlerklassen bildet die Grundlage für deren Kombination inner-

halb des Robotersystems. Dabei ist die Unabhängigkeit der beiden Fehler zu beachten, welche im vorliegenden Fall gegeben ist. Die Anforderungen der beiden Aktionen kommen zwar beide aus der Verhaltenskontrolle, doch falls einer der beiden Fehler auftritt, kann der andere Bereich weiterhin korrekt ablaufen. Beispielsweise kann eine Positionsänderung auch ohne die Suche nach bekannten Gegenständen ablaufen. Die Resultate der einzelnen Anforderungen werden dabei von unterschiedlichen Sensoren erfasst. Die Positionsänderung wird von Kreisel- und Beschleunigungssensoren erkannt, wobei die Gegenstandserkennung von den Kameras abhängig ist. Dadurch können die beiden Fehlerklassen unabhängig voneinander auftreten und untersucht werden.

Zur Kombination dieser Klassen werden zunächst beide Antigene verknüpft. Beide Fehlerklassen weisen bei den Antigenen eine Gemeinsamkeit auf, den Zustand der künstlichen Intelligenz. Diese Übereinstimmung bildet den ersten Teil des gemeinsamen Antigens. Dazu werden die aktuellen Daten bezüglich Positionsänderung und Gegenstandserkennung gefügt. Das dadurch entstandene Antigen ist in der Lage, Differenzen zwischen Anforderung und Resultat beider Fehlerklassen aufzuzeigen.

Einige Vorversuche bildeten das Fundament für die Anpassung der Signale. Dabei wurde eine häufige Abdämpfung der Gefahrensignale durch die Sicherheitssignale der jeweilig anderen Fehlerklasse erkannt, woraufhin eine Erhöhung der Gewichtung um 200% durchgeführt wurde. Die Gewichtung der Sicherheitssignale sowie der PAMP-Signale blieb unverändert. Somit sollen Fehler trotz des Sicherheitssignales der anderen Fehlerklasse erkannt werden.

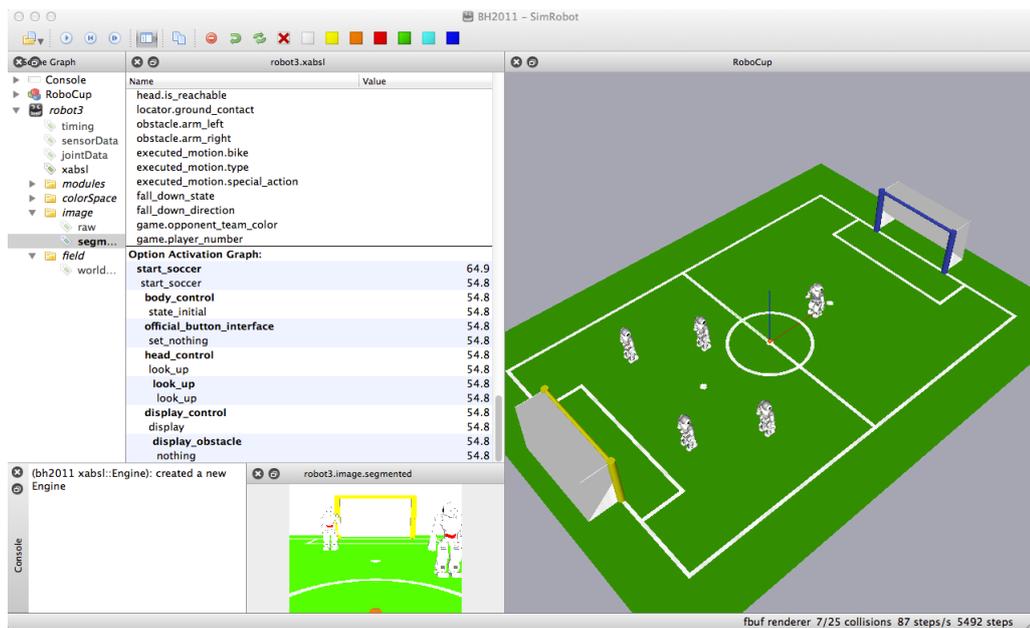


## Evaluierung

Die Evaluierung des implementierten Systems erfolgt anhand umfangreicher Datensammlungen, welche im Laufe spezifischer Versuche erfasst werden. Der Großteil der Experimente wird dabei mithilfe des bereits vorgestellten Simulators durchgeführt, wodurch binnen kurzer Zeit erste Ergebnisse vorliegen. In weiterer Folge werden ausgewählte Versuche am Roboter NAO durchgeführt, was einen Vergleich mit den Werten aus den Simulationsversuchen ermöglicht. Die abgespeicherten Daten werden grafisch aufbereitet und ausgewertet, auf diesem Weg kann eine Bewertung von KRIS vollzogen werden.

Die Erfassung der Daten erfolgt durch einen neu integrierten Logger, welcher nach je hundert verarbeiteten Antigenen den aktuellen Stand von KRIS darstellt. Dabei werden zu jedem erkannten Antigen dessen Vorkommen in reifen und semi-reifen Zellen vermerkt. Des Weiteren wird die gesamte Anzahl der Vorkommen sowie der errechnete MCAV-Koeffizient in die Log-Datei geschrieben. Anhand dieser Werte und des zeitlichen Verlaufs können in einer späteren Phase Rückschlüsse auf das ‐Lernen‐ von KRIS und dessen Effektivität gezogen werden.

Die Durchführung am Simulator wird über mehrere Fenster kontrolliert und gesteuert, welche in Abbildung 7.1 dargestellt sind. Dabei liegt das Hauptaugenmerk auf der visuellen 3-D-Ansicht an der rechten Seite, wodurch die aktuelle Situation am Spielfeld verfolgt werden kann. Auf der linken Seite des Simulatorfensters befinden sich im unteren Bereich zum einen die Konsole, welche eine direkte Steuerung des Roboters ermöglicht, und zum anderen das aktuelle Bild der Roboterkamera. Im linken oberen Bereich veranschaulicht die Abbildung den aktuellen



**Abbildung 7.1:** Experimente im Simulator

Verhaltenszustand sowie die entsprechenden Eingangssymbole für das Verhaltensmodul. Der verwendete Simulator stellt noch zahlreiche weitere Fenster zur Verfügung [29], welche für die Versuche allerdings keine Relevanz besitzen.

Außerdem wird für die Simulation der Versuche ein Szenario angefertigt, welches vier unbewegliche Roboter, einen Ball und einen spielenden Roboter beinhaltet. Der agierende Roboter spielt nach dem bereits vorgestellten Verhaltensmuster aus Abschnitt 2.3 und versucht somit ständig den Ball in das gegnerische Tor zu schießen. Nach jedem Tor wird der Ball zurück auf den Mittelpunkt gelegt und der Roboter, der nicht neu positioniert wird, beginnt erneut mit dem Spiel.

Am Roboter werden anschließend eine Reihe unterschiedlicher Tests durchgeführt. Zunächst werden Szenarien, ähnlich jenen der Simulatorversuche durchgespielt. Des Weiteren sollen realistische Spielszenen beobachtet werden, um das Verhalten von KRIS im Spielverlauf analysieren zu können. Abschließend wird eine weitere Szene zur Präsentation von KRIS erzeugt, wobei das System um eine simple Immunreaktion erweitert werden soll. Die Datenerfassung erfolgt in allen Fällen über eine Log-Datei, die im System des Roboters selbst abgelegt wird. Außerdem existiert bei den Experimenten am Roboter die Möglichkeit, eine Kontrolle der laufenden Daten mit dem Simulator und dessen Remote-Funktion zu vollziehen.

## 7.1 Simulationsexperimente

Der folgende Abschnitt zeigt das Verhalten von KRIS auf einzelne Fehlerklassen und deren Kombination. Zu diesem Zweck wird zunächst jeweils nur eine der beiden Fehlerklassen aus Abschnitt 6.1 aktiviert.

### Experimente zu Positionsänderungen

KRIS soll in die Lage versetzt werden, gefährliche von harmlosen Zuständen zu unterscheiden. Im Falle der Positionsänderungsfehlerklasse sind gefährlich jene Zustände, bei denen eine Bewegung von der KI gefordert wird und der Roboter diese Forderung nicht erfüllt. Die folgenden Versuche sollen zeigen, ob die implementierte Variante des künstlichen Immunsystems mit dendritischem Zell-Algorithmus in der Lage ist, diese gefährlichen Zustände ohne Positionsänderung zu erkennen und von den harmlosen Zuständen abzugrenzen.

Das Auftreten dieses Fehlerfalls kann in den Experimenten durch das einfache Festhalten des Roboters hervorgerufen werden. Sollte sich dabei der Roboter in einem jener Zustände befinden, welche eine Positionsänderung fordern, kann diese durch das Festhalten nicht erfüllt werden und sollte dadurch als Fehler erkennbar sein. In den Versuchen werden dazu mehrere Szenarien dargestellt, innerhalb welcher der Roboter für unterschiedlich lange Zeit an der Positionsänderung gehindert wird. Im Gegensatz dazu sollten alle Zustände, bei denen sich der Roboter frei bewegen kann, als fehlerfrei bewertet werden. Dafür wird im folgenden Versuch ein Spielverlauf ohne Fehlerfall erfasst.

### Ergebnisse

Nach einer Testdauer von zehn Minuten hat KRIS etwa 15.000 Antigene behandelt und daraus deren Gefahrenpotenzial errechnet (Tabelle 7.1). Alle gesammelten Antigene können auf 41 Antigentypen aufgeteilt werden. Jeder Antigentyp erhält durch die Berechnung des MCAVs eine Einschätzung seines Gefahrenpotenzials, welches der Anzahl von Vorkommen in reifen Zellen gebrochen durch die gesamte Anzahl an Vorkommen entspricht. Diese Werte werden zusammen mit der Anzahl der Vorkommen in semi-reifen Zellen im Verlauf des Experiments erfasst und abgespeichert.

**Tabelle 7.1:** Ergebnisse ohne Fehler bei Positionsänderung

Körperbewegung	Kopfbewegung	$\Delta$ Position	semireif	reif	MCAV
gehe_zum_Ball	schaue_zum_Ball	Ja	684	23	0.032532
gehe_zum_Ball	keine_Kopfbewegung	Ja	502	29	0.054614
gehe_zum_Ball	keine_Kopfbewegung	Nein	251	56	0.182410
gehe_zum_Ball	schaue_auf_und_ab	Ja	441	41	0.085062
gehe_zum_Ball	schaue_auf_und_ab	Nein	322	86	0.210784
positioniere_zum_Ball	schaue_zum_Ball	Ja	446	118	0.209220
positioniere_zum_Ball	schaue_zum_Ball	Nein	315	232	0.424132
kontrolliere_Position	schaue_auf_und_ab	Ja	670	0	0.000000
kontrolliere_Position	schaue_auf_und_ab	Nein	55	2	0.035088
schieße	schaue_auf_und_ab	Nein	513	0	0.000000
schieße	schaue_auf_und_ab	Ja	203	0	0.000000
drehe_zum_Ball	schaue_auf_und_ab	Nein	826	15	0.017836
drehe_zum_Ball	schaue_auf_und_ab	Ja	504	0	0.000000
suche_nach_Ball	schaue_auf_und_ab	Ja	443	0	0.000000
positioniere_zum_Tor	schaue_auf_und_ab	Ja	107	0	0.000000
positioniere_zum_Tor	keine_Kopfbewegung	Ja	183	0	0.000000
positioniere_zum_Tor	schaue_zum_Ball	Ja	31	0	0.000000
positioniere_zum_Tor	keine_Kopfbewegung	Nein	7	0	0.000000
suche_nach_Ball	schaue_auf_und_ab	Nein	4	0	0.000000
positioniere_zum_Tor	schaue_auf_und_ab	Nein	4	3	0.428571
positioniere_zum_Tor	schaue_zum_Ball	Nein	1	0	0.000000

Während des Versuchs ist kein Fehler bezüglich der Positionsänderungsfehlerklasse aufgetreten, was sich auch in den Ergebnissen des Experiments widerspiegelt. Alle Einschätzungen befinden sich unter 45% und damit unter der kritischen Wahrscheinlichkeit von 50%. Dabei stechen die Antigene “positioniere\_zum\_Tor und schau\_auf\_und\_ab” sowie “positioniere\_zum\_Ball und schau\_zum\_Ball” hervor, welche jeweils eine Einschätzung von etwa 42% erreicht haben. Bei beiden Zuständen ist keine Positionsänderung zu erkennen, was durch das “Nein” in der entsprechenden Spalte beschrieben wird. In beiden Verhaltenszuständen versucht der Roboter sich möglichst genau zum Ball hinzubewegen, was nur durch sehr langsame Bewegungen sowie durch eine laufend neue Berechnung der eigenen Position erfolgreich durchgeführt werden kann. Aufgrund dieser zögernden Bewegungen des Roboters kann es in einzelnen Fällen dazu kommen, dass der Beschleunigungssensor keine Positionsänderung erkennt. Der Großteil aller Antigene, welche in zeitlicher Nähe zu diesen Verhaltenszuständen auftreten, kann allerdings eine Positionsänderung nachweisen. Dadurch wird anhand der Gruppenbewertung von Antigenen ein sehr geringes Gefahrenpotenzial für diese theoretischen Fehlerzustände berechnet.

**Tabelle 7.2:** Ergebnisse mit Fehler bei Positionsänderung (60 Sekunden)

Körperbewegung	Kopfbewegung	$\Delta$ Position	semireif	reif	MCAV
gehe_zum_Ball	schaue_zum_Ball	Ja	501	18	0.034682
gehe_zum_Ball	keine_Kopfbewegung	Nein	154	138	0.472603
gehe_zum_Ball	keine_Kopfbewegung	Ja	309	20	0.060790
positioniere_zum_Ball	schaue_zum_Ball	Ja	695	122	0.149327
positioniere_zum_Ball	schaue_zum_Ball	Nein	330	252	0.432990
kontrolliere_Position	schaue_auf_und_ab	Ja	597	0	0.000000
kontrolliere_Position	schaue_auf_und_ab	Nein	64	0	0.000000
schieße	schaue_auf_und_ab	Ja	199	0	0.000000
schieße	schaue_auf_und_ab	Nein	423	0	0.000000
drehe_zum_Ball	schaue_auf_und_ab	Nein	846	38	0.042986
drehe_zum_Ball	schaue_auf_und_ab	Ja	314	3	0.009464
gehe_zum_Ball	schaue_auf_und_ab	Ja	247	34	0.120996
gehe_zum_Ball	schaue_auf_und_ab	Nein	195	132	0.403670
gehe_zum_Ball	schaue_zum_Ball	Nein	0	29	1.000000
suche_nach_Ball	schaue_auf_und_ab	Nein	3	1390	0.997846

Ungefähr 20 Prozentpunkte tiefer liegt die Einschätzung weiterer drei Antigene, bei denen besonders der Zustand “positioniere\_zum\_Ball und schaue\_zum\_Ball” hervorgehoben wird, da es sich dabei wiederum um die Ausrichtung des Roboters zum Ball handelt. In diesem Fall hat sich allerdings der Roboter bewegt, aufgrund dessen diese hohe Einschätzung zunächst etwas verwundern mag. Wie bereits beschrieben werden in diesen Verhaltenszuständen teilweise kaum erkennbare Bewegungen durchgeführt, wodurch häufig zwischen Positionsänderung “Ja” und “Nein” gewechselt wird. Da die entstehenden Antigene zeitlich eng zusammenliegen, werden sie im Immunsystem teilweise in denselben Zellen abgelegt und bewertet. Dies führt in weiterer Folge zu einer vergleichsweise hohen Einschätzung des Gefahrenpotenzials. Zwei andere Zustände haben sich bei einer Wahrscheinlichkeit von etwas über 10% eingependelt und alle anderen Einschätzungen liegen bei 0% oder knapp darüber.

Vier weitere Experimente werden durchgeführt, bei denen jeweils ein Fehler mittels Festhaltens hervorgerufen wird. Die Fixierung des Roboters erstreckt sich dabei über unterschiedliche Zeitspannen von 6 bis 60 Sekunden; auf diesem Weg soll die Effizienz des Algorithmus aufgezeigt werden. Die folgenden Ergebnisse zeigen einen Versuch mit einer Fixierung des Roboters für 60 Sekunden, was zum eindeutigen Erkennen mehrerer Fehlerzustände der Positionsänderungsfehlerklasse führt.

Tabelle 7.2 zeigt einen Ausschnitt der wichtigsten durch KRIS erlernten Antigene. Das Experiment wird nach ungefähr 15.000 verarbeiteten Verhaltenszuständen beendet, so dass die Werte mit dem ersten Versuch vergleichbar sind. Entgegen den Erwartungen werden in diesem Fall allerdings nur 38 Antigentypen gespeichert, was einer Differenz von vier Antigentypen zum Referenzversuch entspricht. Bei genauerer Beobachtung ist zu erkennen, dass einige neue Zeichenketten hinzugekommen sind und andere nicht mehr auftauchen.

Die 60-sekündige Fixierung des Roboters erfolgt, während er sich in Richtung des Balles bewegt. Dieser Vorfall spiegelt sich auch in den Ergebnissen des Versuches anhand mehrerer Antigene wieder. Der Zustand “gehe\_zum\_Ball und schaue\_zum\_Ball” ohne Positionsänderung hebt sich aus den anderen Werten hervor, da er eine Einschätzung von 100% erhalten hat. Dieses Antigen entspricht also einem sehr gefährlichen Verhaltenszustand. Eine derart hohe Einschätzung lässt außerdem darauf schließen, dass auch alle Antigene in der zeitlichen Nähe als gefährlich einzuschätzen sind und dass dieses Antigen ausschließlich im Fehlerfall entstehen kann. Das nächste Antigen, welches mit einer Wahrscheinlichkeit von über 99% ein sehr hohes Gefahrenpotenzial in sich birgt, beschreibt den Zustand “suche\_nach\_Ball und schaue\_auf\_und\_ab” ohne Positionsänderung. Ebenso muss dieses Antigen im Fehlerfall sehr stark vertreten gewesen sein, was anhand des hohen Gefahrenpotenzials erkennbar wird. Allerdings kann dieses Antigen in vereinzelten Fällen auch im fehlerfreien Spielbetrieb oder aber am Ende des Fehlerfalls auftreten, wodurch einzelne Vertreter in den Zellen mit harmlosen Antigenen zusammengelegt werden.

Die restlichen Zeichenketten bleiben alle unter der 50%-Marke und werden somit von KRIS als harmlos eingestuft. Trotzdem treten vier Zustände mit Wahrscheinlichkeiten von 32 bis 47% auf, wovon zwei Antigene ebenfalls die Aufforderung “gehe\_zum\_Ball” enthalten. Allerdings unterscheiden sie sich durch die Kopfbewegungen “schaue\_auf\_und\_ab” und “keine\_Kopfbewegung”. Auf den ersten Blick findet sich kein Zusammenhang zwischen den Kopfbewegungen des Roboters und seinen Positionsänderungen. Eine genauere Analyse des Verhaltens zeigt aber, dass dem Roboter bei “keine\_Kopfbewegung” und “schaue\_auf\_und\_ab” die Lage des Balles nicht ausreichend bekannt ist. Durch die vertikale Kopfbewegung und eine zögerliche Bewegung in eine vermutete Richtung, versucht der Roboter den Ball zu finden. Durch die langsame Bewegung entstehen in dieser Phase wiederum viele Antigene, welche Zustände mit und ohne Positionsänderung beschreiben, wodurch eine Einschätzung nahe den 50% entsteht. Erst nach einer festgelegten Zeitgrenze wird der Verhaltenszustand gewechselt. Eine kleinere Gruppe an Zuständen hat eine Einschätzung von etwa 10% erhalten. Allerdings weist der

größte Teil der Antigene wiederum eine Wahrscheinlichkeit von 0% oder wenige Prozentpunkte darüber auf, wobei diese Zeichenketten eine erhebliche Überschneidung mit den Antigenen aus dem Referenzversuch erkennen lassen.

Bei weiteren Versuchen wird die Zeitspanne der Fixierung des Roboters immer weiter verkürzt, so dass Experimente mit einer Festhaltungsdauer von 24 und 12 Sekunden durchgeführt werden. Ein Vergleich der Experimente zeigt, dass sich in allen Fällen ähnliche Daten ergeben haben. Besonders auffällig tritt dabei das Antigen "gehe\_zum\_Ball und schaue\_zum\_Ball" ohne Positionsänderung hervor, welches durchgehend mit dem Wert "1.0" eingeschätzt wird. Eine Abweichung lässt sich hingegen beim zweiten als gefährlich eingeschätzten Zustand erkennen, welcher den Verhaltenszustand "suche\_nach\_Ball und schaue\_auf\_und\_ab" ohne Positionsänderung beschreibt. Dabei ist mit geringerer Fixierungsdauer des Roboters eine Senkung der Gesamtanzahl an Vorkommen des Antigens zu erkennen, aufgrund dessen in weiterer Folge dessen MCAV gemindert wird. Diese Verringerung des Gefahrenpotentials bei kürzerer Fehlerdauer weist auf Vorkommen des Antigens im fehlerfreien Spielbetrieb hin. Dies führt bei längerem Ausbleiben eines Fehlers zum "Vergessen" der Gefährlichkeit dieses Verhaltenszustandes.

Auch in den Versuchen mit einer Fixierungsdauer von 24 beziehungsweise 12 Sekunden hat sich eine kleinere Gruppe im Umkreis von 40 bis 45% gebildet, welche zu großen Teilen mit den bereits bekannten Antigengruppen in diesem Bereich übereinstimmt. Eine weitere Anhäufung an Antigenen bildet sich im Wertebereich von 10 bis 20% des MCAV, wie bereits aus dem vorhergehenden Versuch bekannt ist. Der größte Teil der Verhaltenszustände hat wiederum in beiden Fällen eine Einschätzung unter 1% erhalten, welche ungefähr 65% aller gelernten Antigene umfasst. Beim Versuch mit einer Fixierungsdauer von 10 Sekunden werden 39 Antigentypen erkannt, während bei 5 Sekunden 38 Antigentypen von KRIS erlernt werden, wobei die Dauer der Experimente konstant bei zehn Minuten und damit bei 15.000 verarbeiteten Verhaltenszuständen gehalten wird.

Dies gilt auch bei einem weiteren Versuch, bei dem der Roboter für nur 6 Sekunden festgehalten wird. Dieses Experiment bildet den Abschluss der Versuchsreihe für der Positionsänderungsfehlerklasse und soll zeigen, dass selbst bei diesem kurzen Zeitfenster bereits ausreichende Daten zur Einschätzung der Zustände gesammelt werden können.

Ein Ausschnitt der Ergebnisse des letzten Versuchs wird in Tabelle 7.3 angeführt und zeigt ähnliche Muster wie die restlichen Versuchsergebnisse. Abermals werden 38 Antigentypen erkannt und bewertet, obwohl nicht durchgehend dieselben Zeichenketten gesammelt werden. Die Er-

**Tabelle 7.3:** Ergebnisse mit Fehler bei Positionsänderung (6 Sekunden)

Körperbewegung	Kopfbewegung	$\Delta$ Position	semireif	reif	MCAV
gehe_zum_Ball	schaue_zum_Ball	Ja	633	38	0.056632
gehe_zum_Ball	keine_Kopfbewegung	Nein	204	113	0.356467
gehe_zum_Ball	keine_Kopfbewegung	Ja	449	21	0.044681
positioniere_zum_Ball	schaue_zum_Ball	Ja	545	55	0.091667
positioniere_zum_Ball	schaue_zum_Ball	Nein	347	140	0.287474
kontrolliere_Position	schaue_auf_und_ab	Ja	566	1	0.001764
kontrolliere_Position	schaue_auf_und_ab	Nein	57	1	0.017241
schieße	schaue_auf_und_ab	Ja	284	1	0.003509
schieße	schaue_auf_und_ab	Nein	427	5	0.011574
drehe_zum_Ball	schaue_auf_und_ab	Nein	886	8	0.008949
drehe_zum_Ball	schaue_auf_und_ab	Ja	508	3	0.005871
gehe_zum_Ball	schaue_auf_und_ab	Ja	378	50	0.116822
gehe_zum_Ball	schaue_auf_und_ab	Nein	277	108	0.280519
gehe_zum_Ball	schaue_zum_Ball	Nein	0	29	1.000000
suche_nach_Ball	schaue_auf_und_ab	Nein	5	0	0.000000

gebnisse geben durchwegs deutlich den Verlauf des Versuches wieder. So wird durch die Fixierung des Roboters für 6 Sekunden der Zustand “gehe\_zum\_Ball und schaue\_zum\_Ball” ohne Positionsänderung erlernt und mit der Einschätzung von “1.0” bewertet, was auch in allen anderen Experimenten der Fall ist. Zum einen wird dadurch die Robustheit des dendritischen Zell-Algorithmus, zum anderen die Gefährlichkeit des erkannten Zustandes aufgezeigt.

Der zweite Verhaltenszustand, welcher in den bisherigen Versuchen als gefährlich eingestuft wird, “suche\_nach\_Ball und schaue\_auf\_und\_ab” ohne Positionsänderung – kommt auch in den vorliegenden Ergebnissen vor – wobei sich dessen MCAV-Wert deutlich zu “0.0” abgeändert hat. Dieser Zustand tritt allerdings nur in vereinzelten Situationen auf, da sich die Anzahl der Vorkommen auf fünf beschränkt. Durch die kurze Spanne der Fehlersituation kann der Roboter nicht in den Verhaltenszustand der Ballsuche wechseln, weswegen das Auftreten des Antigens außerhalb des Fehlerzeitraumes liegen muss und so die vorangegangene Vermutung bestätigt.

Der MCAV der weiteren Antigene liegt mit 30% deutlich unter den Werten der vorhergehenden Versuche. Nur der Zustand “gehe\_zum\_Ball und keine\_Kopfbewegung” ohne Positionsänderung wird hierbei durch eine Wahrscheinlichkeit von 35% hervorgehoben. Diese Gegebenheit lässt sich anhand des bereits beschriebenen Phänomens der zögernden Bewegungen des Roboters erklären, welches derart niedrige Bewertungen zur Folge hat. Wiederum hat der größte Teil

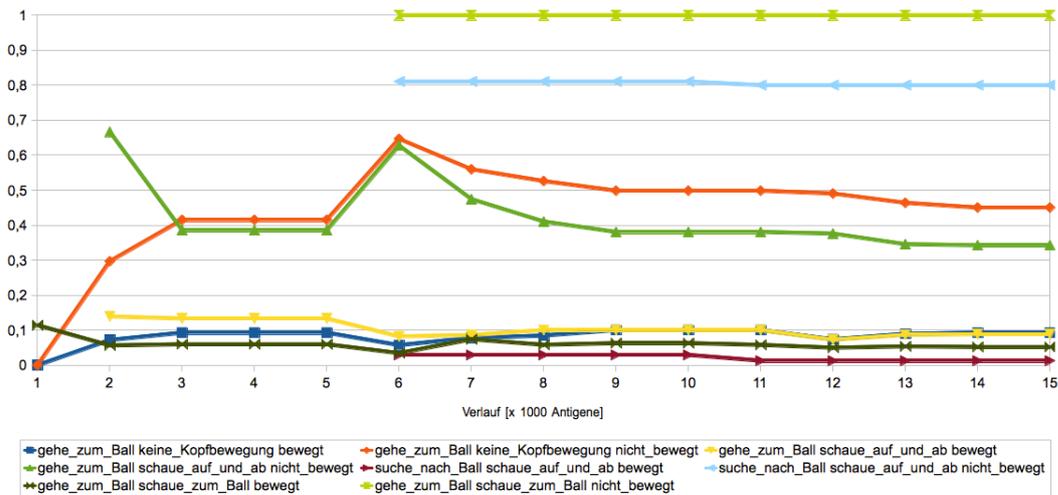


Abbildung 7.2: Verlauf mit Positionsänderungsfehler (12 Sekunden)

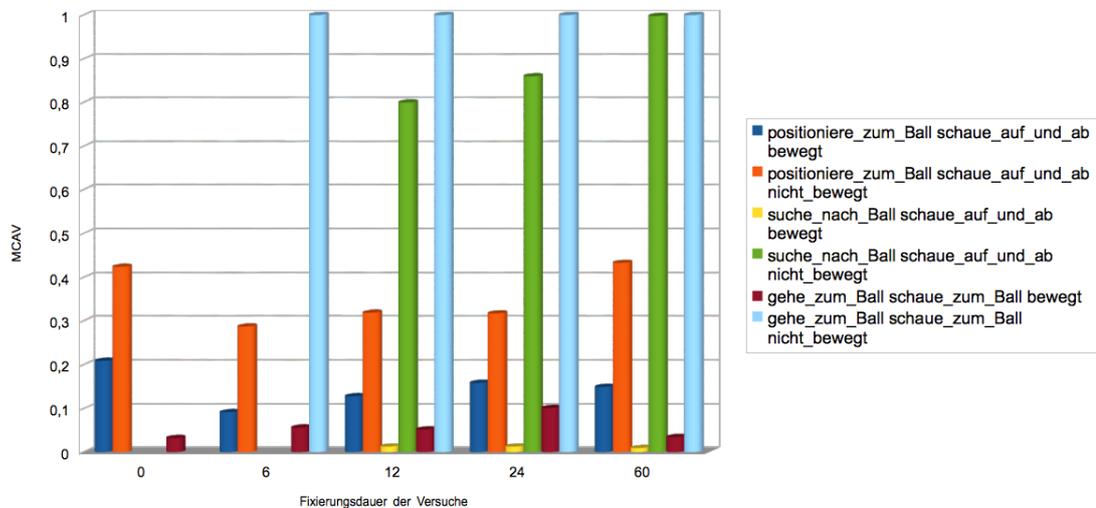
der Antigene einen MCAV-Koeffizienten von 0 oder knapp darüber zugewiesen bekommen.

### Analyse

Die vorgestellten Ergebnisse der durchgeführten Experimente zeigen eine Reihe an Gemeinsamkeiten und Muster, welche durch einen genaueren Vergleich in der Folge hervorgehoben werden sollen. Außerdem werden Besonderheiten in den einzelnen Versuchen aufgedeckt und bezüglich ihrer Ursache untersucht, was abschließend zu einer Bewertung der vorgestellten Methode sowie zu Verbesserungsvorschlägen führen soll.

Abbildung 7.2 zeigt die Veränderung der MCAVs einiger erlernter Antigene über den Verlauf eines Experiments, wobei vergleichbare Ergebnisse in allen Versuchen auftreten. Am Beginn des Versuchs werden enorme Veränderungen bei vielen Werten deutlich, welche sich durch die anfänglich geringe Anzahl der Vorkommen der einzelnen Antigentypen in den Zellen erklären lassen. Mit fortschreitendem Verlauf erhöht sich die Gesamtanzahl der Vorkommen und damit pendeln sich die MCAV-Werte ein.

Einen weiteren Einschnitt im Experimentverlauf stellt der Eintritt der Fehlersituation dar, welcher durch das Erscheinen neuer, unbekannter Antigene deutlich wird. Dabei erhöht sich im Robotersystem die Konzentration der Gefahrensignale, wodurch einige Antigene, welche noch in



**Abbildung 7.3:** Vergleich ausgewählter Antigene für die Positionsänderungsfehlerklasse

unreifen Zellen enthalten sind, beeinflusst werden. Allerdings werden diese gesteigerten MCAV-Werte durch den normalen Spielverlauf wieder ausgeglichen, wie die vorliegenden Werte beweisen.

Beim Auftreten des Fehlers werden umgehend die Gefahrensignale angehoben, aufgrund dessen alle aktuellen Zellen in der Population durch das weitere Aufnehmen von Antigenen stetig mehr gefährliche Antigene beinhalten. Aus diesem Grund erhöht sich später bei der Berechnung des MCAV das Gefahrenpotenzial der bereits vorher vorhandenen Antigene. Nach einiger Zeit erreichen auch Zellen mit den neuen Antigenen den festgelegten Reifegrad und fließen in die Bewertung mit ein.

Ein Problem stellt hierbei die kleine Veränderung des MCAVs von Antigenen “suche\_nach\_Ball und schaue\_auf\_und\_ab” ohne Positionsänderung dar, welche durch das erneute Auftreten dieses Antigens bewirkt wird. Zu diesem Zeitpunkt wird im Experiment keinerlei Fehler ausgelöst. Trotzdem ist das als gefährlich eingeschätzte Antigen aufgetreten und könnte im schlimmsten Fall zu falschen Reaktionen von KRIS führen. Derartige Vorfälle können beim Antigen “gehe\_zum\_Ball und schaue\_zum\_Ball” ohne Positionsänderung nicht erkannt werden, welches damit tatsächlich nur im Fehlerfall auftritt und damit eine Reaktion von KRIS auslösen soll.

Der Unterschied der MCAV-Werte zwischen harmlosen und gefährlichen Antigenen ist in allen Experimenten mit über 50 Prozentpunkten sehr groß und damit ausreichend für deren konkrete

Klassifizierung. In Abbildung 7.3 werden einige Antigene mit deren MCAV aus allen Versuchen dargestellt, wodurch diese beschriebene Differenz deutlich hervortritt.

Die vereinzelt Vorkommen von gefährlichen Antigenen bei fehlerfreiem Betrieb, wie beispielsweise das Antigen “suche\_nach\_Ball und schau\_auf\_und\_ab” ohne Positionsänderung in allen Versuchen zeigt, stellt ein erhebliches Problem in der Behandlung von Fehlern durch KRIS dar. Ein möglicher Lösungsansatz diesbezüglich könnte eine minimale Vorkommensdauer des gefährlichen Antigens vor der Reaktion des Immunsystems sein. Damit ist das implementierte künstliche Immunsystem in der Lage, innerhalb von 6 Sekunden gefährliche Zustände zu erkennen, als solche zu klassifizieren und zu reagieren.

Im Bezug auf die Genauigkeit stellt der Anteil an falsch als gefährlich eingestuften Antigenen ein wichtiges Merkmal bei KRIS dar. Dies ist bei allen Versuchen nur einmal aufgetreten, wobei der MCAV des Antigens einen stark sinkenden Verlauf aufwies und bei einer längeren Versuchsdauer sicher die 50%-Marke unterschritten hätte. Zur Verhinderung derartiger Probleme sollte der Schwellwert zur Einteilung der Antigene heuristisch erforscht und festgelegt werden.

## **Experimente zu visueller Suche**

Neben der Fähigkeit zu Positionsänderungen stellt das Sehen und Erkennen von Gegenständen eine zentrale Fertigkeit des Roboters dar. Bei einer Anforderung der Verhaltenskontrolle, welche die Suche eines bekannten Gegenstandes betrifft, sollte innerhalb eines vorgegebenen Zeitrahmens zumindest ein Tor oder der Ball erkannt werden. Ansonsten kommt dies einem Ausfall der visuellen Suche gleich und das Robotersystem kann als handlungsunfähig eingestuft werden. Die folgenden Experimente sollen aufzeigen, mit welcher Genauigkeit und innerhalb welcher Zeitspanne das implementierte künstliche Immunsystem in der Lage ist, derartige Gefahrensituationen zu erkennen.

Zur Laufzeit des Robotersystems kann die beschriebene Gefahrensituation durch das Löschen der Farbkalibrierung herbeigeführt werden. Die Verhaltenskontrolle wird im weiteren Verlauf eine Anforderung zur Suche des Balls oder eines Tores ausgeben, welcher der Roboter nicht gerecht werden kann. Den Verlust dieser erforderlichen Fertigkeit sollte KRIS als Fehlerfall erkennen und dementsprechend bewerten. Analog zu den vorhergehenden Versuchen wird die Zeitspanne der Behinderung des Roboters im Verlauf der Versuchsreihe weiter verkürzt, um die Lerngeschwindigkeit des Systems aufzuzeigen. Als Referenzversuch wird anfangs ein Experi-

**Tabelle 7.4:** Ergebnisse ohne Fehler bei visueller Suche

Körperbewegung	Kopfbewegung	Sucherfolg	semireif	reif	MCAV
gehe_zum_Ball	schaue_zum_Ball	Ja	1110	0	0.000000
gehe_zum_Ball	keine_Kopfbewegung	Ja	805	0	0.000000
positioniere_zum_Ball	schaue_zum_Ball	Ja	1089	0	0.000000
kontrolliere_Position	schaue_auf_und_ab	Ja	716	0	0.000000
schieße	schaue_auf_und_ab	Ja	714	0	0.000000
drehe_zum_Ball	schaue_auf_und_ab	Ja	1309	0	0.000000
gehe_zum_Ball	schaue_auf_und_ab	Ja	823	0	0.000000
suche_nach_Ball	schaue_auf_und_ab	Ja	426	0	0.000000
positioniere_zum_Tor	schaue_auf_und_ab	Ja	171	0	0.000000
positioniere_zum_Tor	keine_Kopfbewegung	Ja	301	0	0.000000
positioniere_zum_Tor	schaue_zum_Ball	Ja	104	0	0.000000

ment ohne Behinderung durchgeführt, bei dem alle gesammelten Antigene als fehlerfrei bewertet werden sollten.

## Ergebnisse

Der Roboter erhält die Aufgabe, den Ball in das Tor zu befördern, weshalb er sich die gesamte Versuchsdauer mit Ball und Tor beschäftigt. Diese Vorgabe führt dazu, dass der Roboter im normalen Spielverlauf ständig einen der beiden Gegenstände sucht und im Normalfall auch findet. Die Ergebnisse aus den Versuch ohne Fehler geben das beschriebene Szenario exakt wieder, da kein Antigen erkannt wird, bei welchem die Suche nach bekannten Gegenständen erfolglos ist. Tabelle 7.4 beinhaltet die erfassten Werte aus diesem Referenzversuch.

Das Experiment wird nach zehn Minuten beendet, wobei keinerlei Störfälle für die visuelle Suche aufgetreten sind. In diesem Zeitraum sammelt KRIS ungefähr 15.000 Antigene, welche in 23 Antigentypen eingeteilt werden. Die Einschätzungen aller Antigene belaufen sich auf einen MCAV von 0% und liegen damit deutlich unter den Erwartungen. Dieser reibungslose Spielverlauf kommt aufgrund der hohen Toleranzzeit für die visuelle Suche zustande, welche durch den simplen Versuchsaufbau nicht überschritten wird. Die Zeitspanne von fünf Sekunden für die visuelle Suche muss trotzdem beibehalten werden, da diese die Dauer einer halben Drehung des Roboters umfasst. Bei Voruntersuchungen des Verhaltens ist erkannt worden, dass jede 180° Drehung des Roboters bei kürzerer Toleranzzeit einen Fehler zur Folge hätte.

**Tabelle 7.5:** Ergebnisse mit Fehler bei visueller Suche (60 Sekunden)

Körperbewegung	Kopfbewegung	Sucherfolg	semireif	reif	MCAV
gehe_zum_Ball	schaue_zum_Ball	Ja	1012	0	0.000000
gehe_zum_Ball	keine_Kopfbewegung	Ja	772	0	0.000000
gehe_zum_Ball	schaue_auf_und_ab	Ja	783	0	0.000000
positioniere_zum_Ball	schaue_zum_Ball	Ja	1523	0	0.000000
kontrolliere_Position	schaue_auf_und_ab	Ja	759	0	0.000000
schieße	schaue_auf_und_ab	Ja	715	0	0.000000
drehe_zum_Ball	schaue_auf_und_ab	Ja	1159	0	0.000000
kontrolliere_Position	schaue_auf_und_ab	Nein	0	35	1.000000
suche_nach_Ball	schaue_auf_und_ab	Nein	2	1274	0.998433
suche_nach_Ball	schaue_auf_und_ab	Ja	245	0	0.000000
positioniere_zum_Tor	schaue_auf_und_ab	Ja	114	0	0.000000
positioniere_zum_Tor	keine_Kopfbewegung	Ja	197	0	0.000000
positioniere_zum_Tor	schaue_zum_Ball	Ja	74	0	0.000000

Die Ergebnisse, welche in Tabelle 7.5 angeführt werden, lassen hingegen deutlich das Auftreten eines Fehler erkennen. Bei diesem Testlauf wird die Farbkalibrierung der Kamera für 60 Sekunden gelöscht, aufgrund dessen der Roboter nach kurzer Zeit mit der Suche nach dem Ball begonnen hat. Diese Situation wird auch durch einen Teil der vorliegenden Ergebnisse beschrieben.

Der MCAV mehrerer gesammelter Antigene hat einen Wert über 80% angenommen, welcher somit mit sehr großer Wahrscheinlichkeit gefährliche Verhaltenszustände darstellt. Zwei dieser Antigene beschreiben Zustände, in denen die Suche nach bekannten Gegenständen erfolglos blieb. Das Antigen mit dem höheren Gefahrenpotenzial beschreibt den Zustand "kontrolliere\_Position und schaue\_auf\_und\_ab" ohne dabei Ball oder Tor zu erkennen, was zu einer Einschätzung von "1.0" führt. Die Gesamtanzahl der Vorkommen dieses Zustandes deutet darauf hin, dass er nur für einen kurzen Zeitraum eingenommen wird. Nach dem Erkennen des Ballverlustes tritt direkt der Zustand "suche\_nach\_Ball und schaue\_auf\_und\_ab" auf, ohne dabei bekannte Gegenstände zu erkennen. Dieser Zustand kann bis zum Ende der Behinderung nicht mehr verlassen werden und weist daher eine sehr hohe Zahl an Vorkommen auf.

Die sehr hohen Bewertungen der gefährlichen Antigene zeigen, dass diese wirklich nur im Fehlerfall auftreten und sich KRIS daher sehr sicher sein kann. In weiterer Folge bleiben daher komplexere Effekte wie beispielsweise in der Positionsänderungsfehlerklasse aus. Der Unterschied zwischen gefährlichen und harmlosen Antigenen ist in der Fehlerklasse der visuellen Suche mit

90 Prozentpunkten sehr groß, wodurch die restlichen Antigene alle einen MCAV von unter 1% aufweisen, wobei der Großteil der Zustände einen Wert von 0% einnimmt.

Analog zur Versuchsreihe bezüglich der Positionsänderungsfehlerklasse werden die Zeiten der Behinderung des Robotersystems verkürzt und die erhaltenen Ergebnisse verglichen. Aus diesem Grund werden Experimente mit einer Fehlerauftrittsdauer von 24 und 12 Sekunden durchgeführt. Die erhaltenen Ergebnisse weichen nur in geringem Maße von den bereits beschriebenen Werten des Versuchs mit einer Behinderungsdauer von 60 Sekunden ab.

Besonders auffällig tritt dabei der Verhaltenszustand “kontrolliere\_Position und schau\_auf\_und\_ab” ohne dabei Ball oder Tor zu erkennen hervor, welcher in allen weiteren Experimenten eine Bewertung von 100% erreicht hat. Dies unterstreicht zunächst die Robustheit des verwendeten Algorithmus und in weiterer Folge das Gefahrenpotenzial der Situation “kontrolliere\_Position” ohne visuelle Suche. Der Grund dafür liegt an der Lage des Zustandes im Verhaltensmuster, da er gegen Ende der Schusspositionierung auftritt. Die Endphase der Positionierung des Roboters zum Ball bei einem Schuss läuft in mehreren Verhaltenszuständen ab, wobei die Positionierung in Torrichtung den ersten Zustand bildet. Der nächste Zustand wird durch Ausrichtung zum Ball dargestellt, worauf eine Kontrolle der eingenommenen Position folgt. Tritt ein Fehler nun zeitlich knapp vor dieser Positionierung auf, so wird dieser durch die hohe Toleranzzeit der visuellen Suche erst im Letzten der drei Verhaltenszustände erkannt und dann entsprechend eingeschätzt. Dem Antigen für die erfolglose Suche nach dem Ball ohne dabei ein Tor zu sehen werden in allen Versuchen Werte um die 98% zugewiesen. Der Abstand zu den als harmlos eingestuften Antigenen ist in allen Fällen immens, wobei kein Antigen eine Bewertung über 1% erhalten hat. Etwa 72% aller erlernten Antigene haben eine Einschätzung von 0% zugeschrieben bekommen, während bei jedem Versuch zwischen 27 und 29 Antigentypen abgespeichert und bewertet werden.

Bei einem weiteren Experiment der Versuchsreihe wird die Zeitspanne der Behinderung auf 6 Sekunden reduziert, womit der Roboter nur sehr geringfügig eingeschränkt wird. Diese kurze Dauer erlaubt KRIS nur wenige Vertreter des jeweiligen Antigentyps zu sammeln, womit eine Bewertung der Zeichenkette erschwert wird.

Ein Ausschnitt der gesammelten Antigene wird in Tabelle 7.6 angeführt, in welchem insgesamt 27 Antigentypen bewertet werden. Wiederum weist ein Antigen einen MCAV von 100% auf. Dabei ergibt die Entschlüsselung den Verhaltenszustand “kontrolliere\_Position und schau\_auf\_und\_ab” ohne bekannte Gegenstände zu erkennen. Dieses Antigen hat in allen Experimenten

**Tabelle 7.6:** Ergebnisse mit Fehler bei visueller Suche (6 Sekunden)

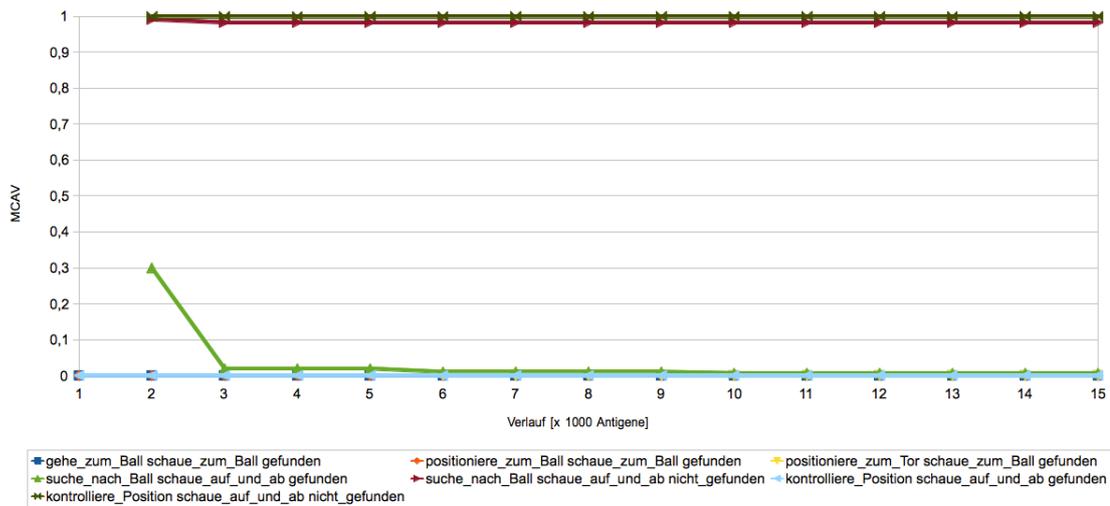
Körperbewegung	Kopfbewegung	Sucherfolg	semireif	reif	MCAV
gehe_zum_Ball	schaue_zum_Ball	Ja	1153	0	0.000000
gehe_zum_Ball	keine_Kopfbewegung	Ja	822	0	0.000000
gehe_zum_Ball	schaue_auf_und_ab	Ja	828	0	0.000000
positioniere_zum_Ball	schaue_zum_Ball	Ja	1041	0	0.000000
kontrolliere_Position	schaue_auf_und_ab	Ja	625	0	0.000000
schieße	schaue_auf_und_ab	Ja	712	0	0.000000
drehe_zum_Ball	schaue_auf_und_ab	Ja	1183	0	0.000000
positioniere_zum_Ball	schaue_zum_Ball	Nein	2	22	0.916667
kontrolliere_Position	schaue_auf_und_ab	Nein	0	5	1.000000
suche_nach_Ball	schaue_auf_und_ab	Ja	473	0	0.000000
positioniere_zum_Tor	schaue_auf_und_ab	Ja	171	0	0.000000
positioniere_zum_Tor	keine_Kopfbewegung	Ja	307	0	0.000000
positioniere_zum_Tor	schaue_zum_Ball	Ja	63	0	0.000000

ten eine identische Bewertung vom Immunsystem erhalten, wogegen der Zustand “positioniere\_zum\_Ball und schaue\_zum\_Ball” noch nicht in der Versuchsreihe aufgetreten ist. Dabei ist der Fehler im Vergleich zu den bisherigen Versuchen etwas früher in der Positionierungsphase aufgetreten und wird bereits im zweiten Teilzustand erkannt. Der Anteil der Antigene mit einem MCAV von 0% bleibt wiederum sehr hoch und nur wenige Ausreißer haben eine höhere Einschätzung erhalten.

## Analyse

Alle durchgeführten Experimente zeigen vergleichbare Ergebnisse, was auf eine hohe Robustheit des dendritischen Zell-Algorithmus bezüglich dieser Fehlerklasse hinweist. Kaum ein Versuch weist Besonderheiten oder Schwankungen innerhalb der Ergebnisse auf. Weitere Vergleiche werden in der Folge diese ersten Erkenntnisse untermauern und zusätzliche Eigenschaften der Fehlerklasse und von KRIS aufzeigen.

In Abbildung 7.4 wird der MCAV einiger erlernter Antigene eines Experiments abgebildet. Vergleichbare Werte treten bei allen Versuchen dieser Fehlerklasse auf, was erneut deren Stabilität unterstreicht. Des Weiteren ist abermals der große Unterschied zwischen den harmlosen und gefährlichen Antigenen in der Abbildung deutlich erkennbar. Alle Antigene mit geringerem oder keinem Gefahrenpotenzial bewegen sich bei 0%, wogegen die Antigene der Fehlerzustände nicht



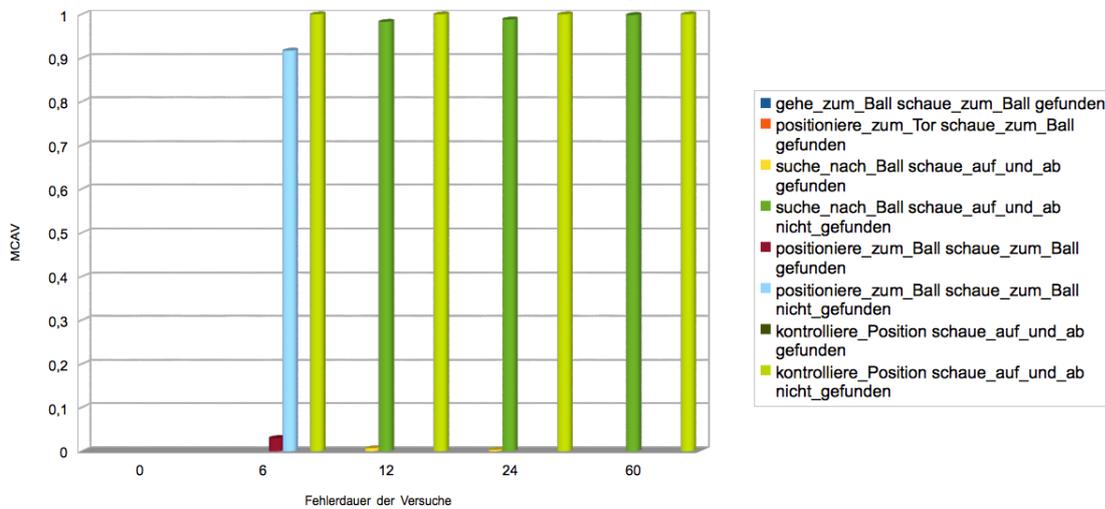
**Abbildung 7.4:** Verlauf mit Fehler bei visueller Suche (12 Sekunden)

unter die Marke von 95% fallen.

Die Grafik zeigt, dass die MCAVs nahezu aller Antigene über die gesamte Versuchsdauer unverändert bleiben. Nur der Zustand “suche\_nach\_Ball und schaue\_auf\_und\_ab” mit dem Erkennen von Ball oder Tor wird zunächst auf etwa 30% eingeschätzt, was aber bereits nach 1000 verarbeiteten Antigenen korrigiert wird. Dies deutet darauf hin, dass dieser Verhaltenszustand knapp vor oder nach dem Zeitpunkt des Fehlervorkommens eingenommen wird. Durch das Auftreten des Fehlers können einige neue Antigene erlernt werden, welche ein hohes Gefahrenpotenzial aufweisen. Allerdings werden dabei keine anderen Antigene beeinflusst, was auf deren Abwesenheit während oder knapp nach dem Fehler hinweist.

Abbildung 7.5 gibt eine Gegenüberstellung markanter Antigene aus allen durchgeführten Experimenten wieder. Die hohe Übereinstimmung der Werte in den letzten drei Versuchen zeigt die Robustheit der Fehlerklasse und des implementierten Algorithmus. Allerdings sollte dabei bedacht werden, dass der Fehler bei allen drei Versuchen im selben Verhaltenszustand des Roboters ausgelöst wurde. Dies ist beim Experiment mit einer Behinderungsdauer von 6 Sekunden nicht gelungen, wodurch auch die Abweichung der erhaltenen Ergebnisse erklärbar ist.

Bei allen Versuchen wird der Fehler im Verhaltenszustand “kontrolliere\_Position und schaue\_auf\_und\_ab” erkannt, ausschließlich beim Versuch mit 6 Sekunden der Behinderung nimmt das System gerade den Verhaltenszustand “positioniere\_zum\_Ball und schaue\_zum\_Ball” ein. Die



**Abbildung 7.5:** Vergleich ausgewählter Antigene für die Fehlerklasse der visuellen Suche

Verhaltenskontrolle setzt trotzdem den Positionierungsverlauf bereits ohne funktionierende visuelle Suche fort, da viele Vorgänge mittels vermuteter Ballposition ablaufen können. Nach Ablauf von 6 Sekunden ist die Fehlersituation in diesem Versuch beendet, wogegen in allen anderen Versuchen für längere Zeit der Verhaltenszustand “suche\_nach\_Ball und schaue\_auf\_und\_ab” eingenommen wird.

Das implementierte Immunsystem ist in der Lage, innerhalb weniger Sekunden die gesammelten Antigene dieser Fehlerklasse korrekt einzuschätzen. Es wird in der gesamten Versuchsreihe kein Antigen fälschlich als gefährlich oder fälschlich als harmlos eingestuft. Der große Unterschied zwischen den unterschiedlichen Bewertungen bietet zudem eine erhebliche Toleranz für Abweichungen in weiteren Spielszenarien. Der Schwellenwert für eine Reaktion von KRIS auf Fehlerfälle kann in dieser Fehlerklasse sehr hoch angesetzt werden, da alle gefährlichen Antigene einen MCAV von nahezu 100% aufweisen.

Diese exakte Bewertung der Fehlerzustände legt die Vermutung nahe, dass eine Immunantwort bei Vorkommen der Gefahrensignale den Anforderungen ebenfalls gerecht werden könnte und somit der dendritische Zell-Algorithmus sinnlos eingesetzt wird. Innerhalb des vorliegenden Versuchsaufbaues würde diese Annahme mit hoher Wahrscheinlichkeit zutreffen, doch kann es bei Spielen gegen andere Roboter oder im Team zu weitaus komplexeren Situationen kommen, welche durch bekannte Eigenschaften des Algorithmus behandelt werden.

**Tabelle 7.7:** Ergebnisse mit Fehler bei visueller Suche und verringerter Toleranzzeit

Körperbewegung	Kopfbewegung	Sucherfolg	semireif	reif	MCAV
gehe_zum_Ball	schaue_zum_Ball	Ja	1218	0	0.000000
gehe_zum_Ball	keine_Kopfbewegung	Ja	896	0	0.000000
gehe_zum_Ball	schaue_auf_und_ab	Ja	846	3	0.003534
positioniere_zum_Ball	schaue_zum_Ball	Ja	1102	0	0.000000
kontrolliere_Position	schaue_auf_und_ab	Ja	625	0	0.000000
schieße	schaue_auf_und_ab	Ja	718	0	0.000000
drehe_zum_Ball	schaue_auf_und_ab	Ja	1258	0	0.000000
suche_nach_Ball	schaue_auf_und_ab	Nein	46	602	0.929012
suche_nach_Ball	schaue_auf_und_ab	Ja	69	20	0.224719
positioniere_zum_Tor	schaue_auf_und_ab	Ja	172	10	0.054945
positioniere_zum_Tor	keine_Kopfbewegung	Ja	245	2	0.008097
positioniere_zum_Tor	schaue_zum_Ball	Ja	81	4	0.047059
positioniere_zum_Tor	schaue_zum_Ball	Nein	2	3	0.600000
positioniere_zum_Tor	keine_Kopfbewegung	Nein	13	72	0.847059
gehe_zum_Ball	schaue_auf_und_ab	Nein	1	6	0.857143
positioniere_zum_Ball	schaue_zum_Ball	Nein	0	32	1.000000
kontrolliere_Position	schaue_auf_und_ab	Nein	0	63	1.000000
gehe_zum_Ball	schaue_zum_Ball	Nein	1	0	0.000000
positioniere_zum_Tor	schaue_auf_und_ab	Nein	4	3	0.428571

Zur Überprüfung dieser Behauptung und detaillierten Analyse dieser Eigenschaften wird die Toleranzzeit der visuellen Suche auf 1/10 Sekunde verringert. In einem Experiment mit einem Fehlervorkommen von 12 Sekunden sollen ähnliche Eigenschaften von KRIS wie in den Versuchen zur Positionsänderungsfehlerklasse hervortreten. Die Einschätzungen der Antigene werden durch diese Konstellation enorm ansteigen, besonders der Verhaltenszustand “suche\_nach\_Ball und schaue\_auf\_und\_ab” wird durch die beinhaltete Drehung ein hohes Gefahrenpotenzial aufweisen. Dennoch sollten Antigene, welche bei Fehlern auftreten, von den restlichen Zuständen abgrenzbar sein.

Tabelle 7.7 gibt die erhaltenen Ergebnisse wieder. Es gibt deutlich mehr Zustände, bei denen die visuelle Suche erfolglos verlief, wobei nicht jedes Antigen als hochgradig gefährlich eingestuft wird. Beispielsweise wird der Zustand “gehe\_zum\_Ball und schaue\_zum\_Ball” ohne bekannte Gegenstände zu finden mit 0% bewertet, was auf ein Vorkommen im fehlerlosen Betrieb hinweist. Hingegen wird die Suche nach dem Ball ohne dabei Ball oder Tor zu sehen erwartungsgemäß sehr hoch bewertet, da es sich dabei um eine Drehbewegung des Roboters handelt. Der Roboter braucht etwa 5 Sekunden um eine Drehung von 180° zu vollziehen und sieht in die-

sem Zeitraum keine bekannten Gegenstände, aufgrund dessen der MCAV derart hoch berechnet wird.

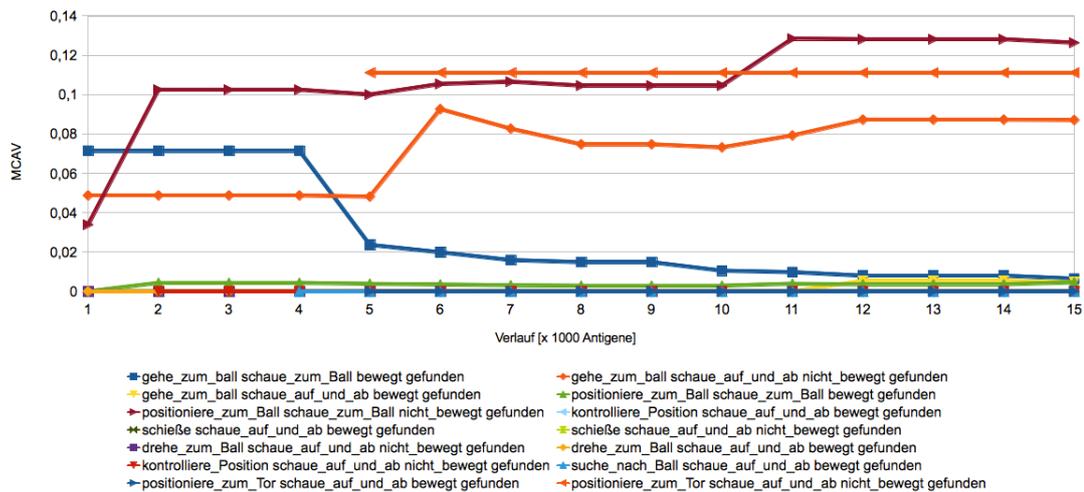
Der hervorgerufene Fehler umfasst drei Verhaltenszustände. Neben “suche\_nach\_Ball und schaue\_auf\_und\_ab”, welcher aufgrund seiner Vorgeschichte nur einen leichten Anstieg aufweist, werden zwei neue Antigene erzeugt. “Positioniere\_zum\_Ball und schaue\_zum\_Ball” sowie “kontrolliere\_Position und schaue\_auf\_und\_ab” ohne erfolgreiche visuelle Suche sind bis zum Fehler unbekannt und werden umgehend mit einer Einschätzung von 100% belegt. Dieses hohe Gefahrenpotenzial ist auch aus vorhergehenden Versuchen bekannt und kann mit dem Positionierungsverlauf bei einem Schuss sowie der sinkenden Wahrscheinlichkeit den Ball oder das Tor nicht zu sehen begründet werden.

Zusammenfassend zeigt der Versuch, dass auch in dieser Fehlerklasse ähnliche Effekte bei der Einschätzung auftreten, wie beispielsweise bei den Positionsänderungsfehlern. Dabei weisen Zustände, welche zwar durch die erfolglose Suche die Definition der Fehlerklasse erfüllen, aber durch ihre häufigen Vorkommen im fehlerlosen Betrieb dies entschärfen, ein deutlich geringeres Gefahrenpotenzial als reine Fehlerzustände auf. Diese wichtige Eigenschaft würde bei einer Reaktion auf die eintretenden Gefahrensignale verloren gehen und rechtfertigt damit den Einsatz des implementierten Algorithmus.

## **Experimente zu gemeinsamer Behandlung von Positionsänderung und visueller Suche**

Die bisher angeführten Ergebnisse haben gezeigt, dass KRIS sehr gut in der Lage ist, Fehler beider beschriebener Klassen innerhalb weniger Sekunden zu erkennen. Allerdings wird bei allen durchgeführten Experimenten jeweils nur eine Fehlerklasse aktiviert und berücksichtigt. Das implementierte Immunsystem wäre in der Praxis von größerem Nutzen, falls es verschiedene Fehlerklassen gleichzeitig behandeln könnte. Dazu werden lediglich die Signale und Antigene wie in Abschnitt 6.4 beschrieben vereinigt. Die folgenden Ergebnisse zeigen die Fähigkeiten des Systems bei der Kombination mehrerer Fehlerklassen. Neben den Experimenten im Simulator werden Versuchsergebnisse am NAO-Roboter vorgestellt, welche den ersten Schritt zum praktischen Einsatz bilden.

Der Aufbau des Simulators bleibt identisch zu den vorangegangenen Versuchen. Ebenso wird die Versuchsdauer wiederum auf zehn Minuten, ergo auf 15.000 abgearbeiteten Antigenen festge-



**Abbildung 7.6:** Verlauf ohne Fehler

legt. Der erste Versuch zeigt die Einschätzungen und Reaktionen von KRIS auf den fehlerfreien Spielbetrieb, der als Referenzversuch dient.

Während des gesamten Experimentverlaufs kann sich der Roboter vollkommen frei bewegen. Ebenso ist er durchgehend in der Lage, die visuelle Suche nach Ball oder Tor erfolgreich durchzuführen, wodurch die Ergebnisse in Abbildung 7.6 zu keiner Zeit einen größeren Anstieg aufweisen. Der MCAV-Koeffizient keines der erlernten Antigene erreicht einen Wert über 15%, was wiederum die Robustheit des Algorithmus unterstreicht. Der Großteil aller Antigene wird mit einem MCAV zwischen 0% und 5% bewertet und die restlichen Verhaltenszustände weisen im Verlauf des Experiments deutlich sinkende MCAVs auf. Lediglich zwei Zustände besitzen einen steigenden Verlauf. Der Zustand “gehe zum Ball und schaue auf und ab” ohne Positionsänderung wird zunächst mit einem MCAV von 5% eingeschätzt und steigt bis zum Ende hin auf einen Wert von nahezu 9%. Noch deutlicher ist der Anstieg beim Antigen “positioniere zum Ball und schaue zum Ball” ohne Positionsänderung zu erkennen, welcher um nahezu 10 Prozentpunkte steigt.

Bei beiden Verhaltenszuständen mit steigendem MCAV-Koeffizient werden im Spielbetrieb häufig sehr kleine Positionsänderungen gefordert, welche der Roboter zwar erfüllt, aber der Beschleunigungssensor nicht erkennen kann. In weiterer Folge führt dies zu einem Fehlerzustand der Klasse Positionsänderung, was auch dementsprechende Signale im System hervorruft. Im fehlerfreien Spielbetrieb besitzt jeder dieser Zustände einen Gegenspieler, welcher durch den-

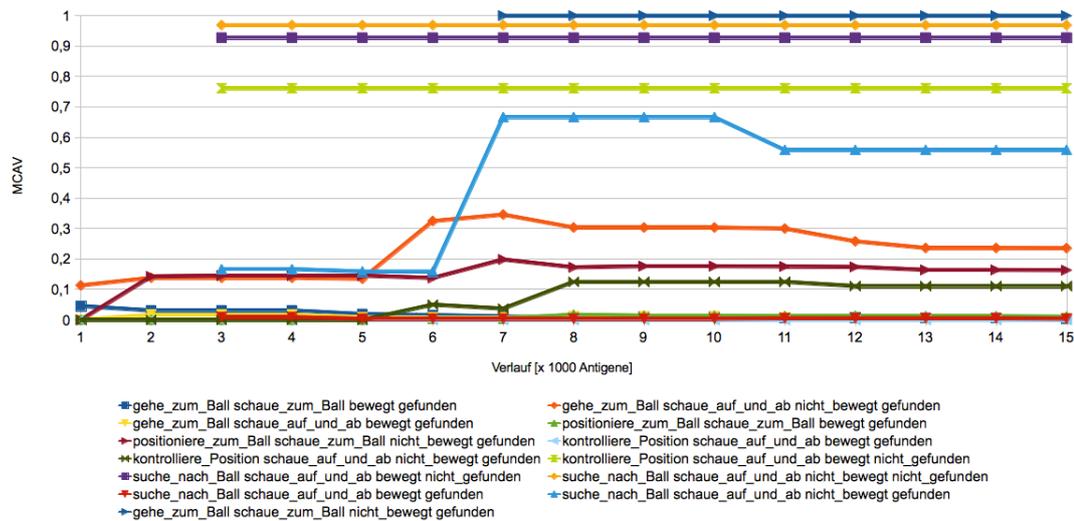
selben KI-Zustand mit Positionsänderung verkörpert wird. So stellt beispielsweise der Zustand “positioniere\_zum\_Ball und schaue\_zum\_Ball” mit erkannter Änderung der Position den Gegenspieler zu “positioniere\_zum\_Ball und schaue\_zum\_Ball” ohne Positionsänderung dar. Im Spiel wechseln sich diese Zustände aufgrund der kleinen Bewegungen des Roboters ständig ab und werden daher vom dendritischen Zell-Algorithmus zusammen bewertet. Dies führt dazu, dass das Verhältnis der Vorkommen stark auf die jeweilige Einschätzung einwirken wird.

Neben der Anzahl der Vorkommen spielen bei der Einschätzung solcher Verhältnisse eine Vielzahl von Faktoren eine tragende Rolle, beispielsweise die Größe der Zellpopulation oder der Schwellenwert der Zellreifung. Die Gewichtung der Eingangssignale trägt außerdem zur Bewertung bei, wobei das Sicherheitssignal um ein Viertel stärker ist als das Gefahrensignal. Somit ergeben die Systemparameter zusammen mit den Nachbarantigenen und deren Eingangssignalen in letzter Instanz das Gefahrenpotenzial für jedes Antigen. Die Systemparameter bleiben über die gesamte Laufzeit konstant, wodurch eine Veränderung des MCAV-Koeffizienten eines Antigens ausschließlich durch die Änderungen der Auftrittsumgebung ausgelöst wird. Im konkreten Fall des Zustandes “positioniere\_zum\_Ball und schaue\_zum\_Ball” ohne Positionsänderung wird mittels Langzeitversuchen erkannt, dass sich der MCAV-Koeffizient auf einen Wert um 16% einpendelt.

Der MCAV-Koeffizient des Zustandes “gehe\_zum\_Ball und schaue\_auf\_und\_ab” ohne Positionsänderung wird zusätzlich durch die Situation nach jedem Schuss gemindert, da die Verhaltenskontrolle zwar diesen Verhaltenszustand einnimmt, aber noch nicht die Aufforderung zur Positionsänderung stellen kann, bedingt durch das Andauern der Schussbewegung. Dies hat das Vorkommen des Antigens ohne die entsprechenden Gefahrensignale zur Folge. Diese Verringerung des MCAV-Koeffizienten wird auch in Langzeitversuchen erkannt, wo sich ein Wert um 12% ergeben hat.

Ein weiterer Versuch soll die Entwicklung des MCAV-Koeffizienten beim Auftreten von Fehlern aufzeigen. Dabei wird die Veränderung zunächst durch die Fehlerklasse der visuellen Suche herbeigerufen, worauf direkt das Festhalten des Roboters folgt und somit ein Fehler bei der Positionsänderung erkannt werden soll. Die einzelnen Auftretungsmuster wurden bereits in vorangegangenen Experimenten untersucht und sind daher ausreichend bekannt. Aus diesem Grund liegt bei diesem Versuch das Augenmerk auf den Auswirkungen der Fehler aufeinander.

Abbildung 7.7 zeigt den Verlauf des Experiments über zehn Minuten anhand der MCAV-Koeffizienten einzelner Antigene, wobei nach etwa zwei Minuten eine erste deutliche Veränderung in den



**Abbildung 7.7:** Verlauf mit beiden Fehlerklassen

Werten zu erkennen ist. Dabei treten gleich mehrere neue Antigene auf, welche teilweise eine sehr hohe Einschätzung erhalten. Diese eindeutigen Hinweise lassen das Auftreten eines Fehlers vermuten, welcher im konkreten Fall durch das Löschen der Farbkalibrierung des Roboters ausgelöst wird. Nach ungefähr 12 Sekunden wird die Kamera wieder aktiviert und der Roboter kann den Spielbetrieb wieder aufnehmen. Weitere zwei Minuten später steigt plötzlich der MCAV-Koeffizient eines Antigens rapide an, worauf ein neues sehr gefährliches Antigen erlernt wird. Dieser Fehler wird durch eine Fixierung des Roboters über eine Dauer von 12 Sekunden ausgelöst. Anschließend werden keine weiteren Fehler hervorgerufen, was durch den weiteren Verlauf der MCAV-Koeffizienten sichtbar wird.

Der Fehler der visuellen Suche nach etwa zwei Minuten der Versuchsdauer lässt KRIS fünf neue Antigene erlernen. Dabei werden einige Antigene als ungefährlich eingestuft, da sie als Folge des vorangegangenen Fehlers im fehlerfreien Verhalten auftreten. Diese beiden Antigene beschreiben die anfängliche Suche nach dem Ball, welche durch die ausgeprägte Toleranzzeit als erfolgreich eingestuft wird. Den restlichen drei Neulingen werden sehr hohe Einschätzungen zugewiesen und diese werden damit als gefährlich eingestuft, wobei zwei Antigene den Verhaltenszustand "suche\_nach\_Ball und schaue\_auf\_und\_ab" ohne Tor oder Ball zu erkennen wiedergeben und sich dabei nur durch erkannte oder nicht erkannte Positionsänderung unterscheiden. Auch bei der Suche nach dem Ball werden vergleichsweise langsame Bewegungen durchgeführt, welche wiederum zu diesem Wechsel zwischen aufgetretener Positionsänderung

und keiner Änderung der Position führen. Dies hat allerdings nicht das hohe Gefahrenpotenzial dieser Zustände zur Folge, welches durch den Ausfall der visuellen Suche derart hoch eingestuft wird. Ebenso ergeht es dem Verhaltenszustand “kontrolliere\_Position und schaue\_auf\_und\_ab” ohne erfolgreiche Suche, welcher einen MCAV-Koeffizient von 76% erhalten hat.

Der Fehler bei der angeforderten Positionsänderung, welcher nach vier Minuten des Versuchsvorgangs auftritt, erzeugt hingegen nur ein neues Antigen, welches allerdings direkt eine Einschätzung von 100% zugewiesen bekommt. Dabei handelt es sich um den Verhaltenszustand “gehe\_zum\_Ball und schaue\_zum\_Ball” ohne Positionsänderung, welcher bereits in vielen vorangegangenen Versuchen als gleichermaßen gefährlich eingestuft wird. Außerdem sind unterschiedlich starke Anstiege beim MCAV-Koeffizient bei weiteren vier Antigenen zu erkennen, welche nach der Definition der Fehlerklasse Positionsänderung angehören. Diese Antigene treten ungeachtet dessen auch im fehlerfreien Systemzustand auf und werden sich deshalb nach kürzerer Zeit einer Wahrscheinlichkeit annähern, welche sich aus Systemparametern und der Auftretsumgebung des Antigens zusammensetzt.

Grundsätzlich können in allen Versuchen drei verschiedene Arten von Antigenen erkannt werden, welche unterschiedliche Eigenschaften aufweisen. Am häufigsten treten die ungefährlichen Antigene auf, welche keiner Definition einer Gefahrenklasse gerecht werden und auch nur im fehlerfreien Systemzustand auftreten. Das Gegenstück dazu bilden die gefährlichen Antigene, welche einer Fehlerklassendefinition entsprechen und ausschließlich im Fehlerfall auftreten. Würden im System nur diese beiden Antigenarten auftreten, könnten die Unterscheidung sowie Immunreaktionen ausschließlich durch die Signale erfolgen.

Jedoch existiert eine weitere Art von Antigenen, welche zwar der Definition einer Fehlerklasse entspricht, aber auch im fehlerfreien Systemzustand auftritt. Diese Antigene stellen KRIS und damit den dendritischen Zell-Algorithmus vor eine enorme Herausforderung, da sie nur im tatsächlichen Fehlerfall als Gefahr erkannt werden dürfen. Eine Eigenschaft dieser Antigene ist es, dass sie im Fehlerfall gehäuft auftreten, hingegen im Normalzustand nur vereinzelt erscheinen. Diese sporadischen Vorkommen versetzen den dendritischen Zell-Algorithmus durch seine Einschätzungsverfahren von ganzen Antigenengruppen in die Lage, das geringe Gefahrenpotenzial derartiger Zustände zu erkennen und entsprechend zu bewerten.

## Parameteroptimierung

Die Parameterwerte bei den durchgeführten Versuchen entsprechen jenen aus der Literatur [28], was den Aufwand der Voruntersuchungen einschränkt aber dennoch zum Erfolg führt. Ein Teil der Parameter sind allerdings bedeutungslos, da sie im Verlauf der Versuche nicht zum Tragen kommen. Dabei werden beispielsweise die Anzahl der Signale pro Signalkategorie auf eins und die maximale Anzahl an Antigenen in einer Zelle auf 50 beschränkt. Wogegen andere Parameter wie die Größe der Zellpopulation oder der Migrationsmedian unmittelbar Auswirkungen auf den Ausgang der Versuche erkennen lassen.

Die ersten Voruntersuchungen zeigen, dass der Algorithmus mit den Standardwerten nur zögerlich auf etwaige Veränderungen reagiert, wodurch es in weiterer Folge zu einer eingreifenden Veränderung kommt. Die Verteilung der gesammelten Antigene und die einhergehende Aktualisierung der Zellen erfolgen erst nach einer Zeitspanne von hundert Sekunden, was zwar die benötigte Rechenleistung begrenzt, aber dadurch die Reaktionszeit und Genauigkeit von KRIS vermindert. Die Genauigkeit der Einschätzungen wird in diesem Fall dadurch beeinträchtigt, dass zur Bewertung aller gesammelten Antigene nur die aktuellen Werte der Signale verwendet werden. Aus diesen Gründen wird bei den hier angeführten Experimenten die Aktualisierung der Zellen nach jedem erkannten Antigen erfolgen.

Durch diese Veränderung kann bei jedem auftretenden Antigen der aktuelle Wert der Signale zu dessen Einschätzung herangezogen werden, aufgrund dessen die Genauigkeit des dendritischen Zell-Algorithmus deutlich erhöht wird. Des Weiteren wird die Reaktionszeit von KRIS deutlich verkürzt. Im vorliegenden System wird durchschnittlich alle 0,02 Sekunden ein Antigen erzeugt, was dazu führt, dass eine Zelle im Mittel nach 0,35 Sekunden die Migrationsgrenze erreicht und zur Berechnung der MCAV-Koeffizienten verwendet wird.

Die Populationsgröße wird durch die kurze Reifungszeit zunehmend irrelevant, da nur zehn bis zwanzig Antigene im Reifeprozess einer Population auftreten. Allerdings kann durch eine höhere Populationsanzahl und der unterschiedlichen Migrationsgrenzen der einzelnen Zellen eine feinere Granulierung der zu bewertenden Zeitspannen erfolgen. Somit ist von der Verwendung einer einzelnen Zelle oder einer Populationsgröße über der Anzahl von Antigenvorkommen abzuraten.

Der Migrationsmedian sowie die Migrationstoleranz, welche zusammen für jede Zelle die Migrationsgrenze ergeben, erhalten somit eine tragende Rolle in der Konfiguration von KRIS, da

sie die Lebensdauer der Zellen und damit die Länge der einzuschätzenden Zeitspannen festlegen.

Bei Versuchen mit einem Migrationsmedian von zehn liegt die Reifedauer der einzelnen Zellen durchschnittlich bei einer Zeitspanne von 0,7 Sekunden. Im Laufe der Periode können somit nur 2-3 Antigene auftreten, wodurch die Bewertung von scheinbaren Fehlerzuständen im fehlerfreien Betrieb kaum gemildert werden kann und aus diesem Grund sehr hohe MCAV-Koeffizienten bei derartigen Antigenen auftreten. Andererseits treten tatsächliche Fehlerzustände deutlich hervor und werden zwischen 71 und 100% eingeschätzt. Bei einem Migrationsmedian von 240 tritt hingegen eine starke Verringerung aller Werte auf, da eine Zelle im Mittel 1,3 Sekunden benötigt um die Migrationsgrenze zu erreichen. Aufgrund dessen wird eine größere Menge an Antigenen zur selben Zeit bewertet. Dies führt in weiterer Folge zu einer Verringerung des MCAV-Koeffizienten der scheinbaren Fehlerzustände im fehlerfreien Verlauf. Allerdings wirkt sich dieser Effekt auch auf die tatsächlichen Fehlerzustände aus, welche sich in einem Bereich von 30% und 97% bewegt.

Durch diese beiden Versuche konnte der ideale Migrationsmedian auf einen Bereich zwischen 10 und 240 eingegrenzt werden. Durch weitere Versuche wird nun das Spektrum immer weiter beschnitten, bis schlussendlich ein hinreichend guter Wert ermittelt werden kann. Durch diesen Prozess kann der Migrationsmedian auf einen Bereich um 60 begrenzt werden, dabei zeigen geringfügige Abweichungen keinen erkennbaren Einfluss auf die Ergebnisse. Dieser Wert kann allerdings nur für das vorliegende System als ideal bezeichnet werden und wird sich demzufolge bei einfacheren Systemen verringern und bei komplexeren Systemen schnell erhöhen.

## **7.2 Einsatztest**

Bereits im Vorfeld der Robocup Weltmeisterschaft 2012 werden erste Versuche des implementierten Immunsystems durchgeführt und etwaige Einstellungen der einzelnen Parameter vorgenommen. Dabei werden im Laufe der Vorbereitung des Systems erste Erkenntnisse über den Einsatz im Echtbetrieb gewonnen. Des Weiteren wird eine einfache Reaktion von KRIS implementiert, welche im Fehlerfall durchgeführt wird. Im Rahmen der Open Challenge der Weltmeisterschaft wird KRIS präsentiert.

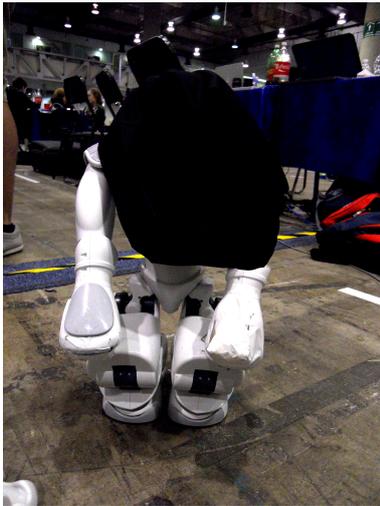
## Fallstudie: Robocup 2012

Die Integration von KRIS in den aktuellen Code der Austrian Kangaroos erfolgt in zwei Schritten. Zunächst wird der neue KRIS-Prozess mit allen seinen Modulen integriert. Anschließend wird die Schnittstelle zur aktuellen Robotersteuerung neu aufgebaut.

Der Aufbau der Verhaltenskontrolle ist in diesem System um Vieles komplexer, so gibt es neben den neuen Rollen wie “Striker”, “Supporter”, “Defender” und “Keeper” komplett neue Verhaltensmuster wie die Positionierung oder das Torwartverhalten. Durch die Anpassung der einzelnen Parameter wird versucht, auch diese weitreichenden Änderungen zu erfassen. Durch die grundlegende Struktur der definierten Fehlerklassen können diese weiterhin zum Einsatz kommen und bedürfen keinerlei Veränderungen. Durch die neue Komplexität der Verhaltenskontrolle kommt es dazu, dass die einzelnen Verhaltenszustände seltener auftreten. Dies führt in weiterer Folge zu einer höheren Anzahl an falsch eingeschätzten Antigenen. Um derartigen Fehleinschätzungen entgegen zu wirken, wird der Migrationsmedian um das Zehnfache erhöht. Die so erhaltenen Ergebnisse zeigen große Übereinstimmungen mit den erhaltenen Werten aus der Simulation.

Die Verhinderung der Positionsänderung gestaltet sich am Roboter sehr schwierig, da die Bewegungssensoren bei nahezu jeder Art des Festhaltens oder Hochhebens eine Veränderung wahrnehmen. Nur vereinzelt konnten Antigene der Positionsänderungsfehlerklasse in den erhaltenen Ergebnissen entdeckt werden, was eine spezifische Auswertung nicht möglich macht.

Bei der Erkennung bekannter Gegenstände ist die Erzeugung des Fehlers durch die Entfernung der Farbkalibrierung sehr viel einfacher. Die ersten Ergebnisse zeigen ein verzögertes Auftreten neuer Antigene, welche dem Muster der Fehlerklasse entsprechen und einen MCAV-Koeffizient von 60% bis 100% aufweisen. Im Vergleich zu den Ergebnissen der Simulation werden die Fehler zwar mit einer geringeren Genauigkeit erkannt, aber dennoch kann eine klare Abgrenzung zu den restlichen Antigenen erkannt werden. Im Spielbetrieb zeigt sich ebenfalls das erwartete Bild. Die Einschätzung der meisten Antigene liegt zwischen 0% und 20%, wodurch diese als ungefährlich eingestuft werden. Eine kleine Gruppe an Antigenen wird allerdings als Fehler mit hundertprozentiger Wahrscheinlichkeit eingestuft. Die Analyse der Zustände ergibt, dass es sich dabei um die Ballsuche mit Teamhilfe handelt. Dabei werden bei Ballverlust die Teamkameraden um die aktuelle Ballposition gefragt und an dieser Stelle wird der Ball gesucht. Allerdings gibt es hierbei häufig Probleme mit der Genauigkeit oder falsch lokalisierten Teammitgliedern,



**Abbildung 7.8:** Roboter mit Behinderung



**Abbildung 7.9:** Visuelle Unterstützung der Präsentation - grün: Anzahl der gesunden Zellen, rot: Anzahl der gefährlichen Zellen, blau: Summe der gereiften Zellen

was zu einer erfolglosen Suche führt. Durch die Erhöhung der Gefahrensignale erkennt KRIS in diesem Verhaltenszustand einen Fehler.

Zur Präsentation von KRIS wird eine Reaktion auf einen Fehler implementiert. Diese bezieht sich auf die Anzahl der neu gereiften Zellen nach jedem Algorithmuszyklus. Dabei wird die Anzahl der harmlosen Zellen mit der Anzahl der gefährlichen Zellen, welche in jedem Berechnungszyklus gereift sind, verglichen. Sollte die Anzahl der gefährlichen Zellen öfters jene der harmlosen Zellen übertreffen, wird die Immunreaktion ausgelöst. In diesem Fall setzt sich der Roboter und unterstreicht akustisch seinen Fehlerzustand. Hervorgerufen wird dieser Zustand durch KRIS, welches durch Interprozesskommunikation in die Lage versetzt wird, direkt in die Verhaltenskontrolle einzugreifen.

Zur Präsentation läuft der Roboter im Kreis und versucht durch unterschiedliche Kopfbewegungen seine Lokalisierung zu verbessern. Im Verlauf der Vorführung wird dem Roboter ein schwarzer Stoffbeutel über den Kopf gestülpt. KRIS soll diesen Ausfall der Kameras erkennen und entsprechend reagieren. Sobald die Behinderung wieder entfernt wird, soll der fehlerfreie Betrieb erkannt werden und der Roboter wieder im Kreis laufen.

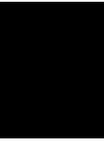
## 7.3 Fazit

Ziel dieser Arbeit war es, einem cyberphysikalischen System mittels künstlichen Immunsystems die Möglichkeit zu geben, sowohl deterministisch als auch nicht deterministisch auftretende Fehler in Echtzeit zu registrieren und dabei von fehlerfreien Zuständen zu unterscheiden. Dazu wurde der dendritische Zell-Algorithmus an die Aufgabe der Fehlererkennung angepasst und in einen Fußball spielenden NAO-Roboter integriert. Dieser erhielt dadurch die Fähigkeit den Ausfall von visuellen sowie motorischen Fähigkeiten zu diagnostizieren.

Jene Experimente mit einzelnen Fehlerklassen weisen besonders herausragende Ergebnisse auf. Im Fall der visuellen Suche konnten 99,8% der aufgetretenen Fehler erkannt werden, wobei kein einziger harmloser Zustand fälschlich als Fehler bewertet wurde. Im Gegensatz dazu wurden innerhalb der Positionsänderungsfehlerklasse vereinzelt Zustände in der Anfangsphase der Versuche zeitweise zu gefährlich eingestuft. Allerdings konnte das System in allen Fällen diese Fehleinschätzung innerhalb weniger Sekunden korrigieren.

Eine kaum messbare Verschlechterung ist während der parallelen Behandlung beider Fehlerklassen zu erkennen. Im Gesamtergebnis konnten auch hierbei 99% der fehlerfreien Zustände als solche erkannt werden. Fehler der Positionsänderung und der visuellen Suchen konnten alle erkannt werden, wobei vereinzelt Zustände erst nach einigen Sekunden tatsächlich als Fehler aufschienen. Besonders bei den Versuchen am echten Roboter konnte dies einige Male beobachtet werden.

Nach diesen vielversprechenden Ergebnissen und der erfolgreichen Präsentation von KRIS beim Robocup ist mittelfristig mit einem praktischen Einsatz derartiger Systeme in diesem Bereich oder auch größeren Projekten zu rechnen.



# Schlussfolgerung und Ausblick

Die Fehlerbehandlung bei der Softwareentwicklung erfordert einen immer größeren Aufwand bedeutet. Diese Beobachtung stellt eine maßgebliche Motivation für diese Arbeit dar. Besonders bei umfangreicheren Systemen, wie Robotern, stellt bereits die Fehlererkennung ein erhebliches Problem dar, welches durch den steigenden Grad der Komplexität häufiger und in geringeren Zeitabständen auftritt.

Biologische Immunsysteme weisen eine Vielzahl attraktiver Eigenschaften auf, welche man auch gerne mit einem Fehlererkennungssystem erreichen würde. Aus diesem Grund wurde in dieser Arbeit ein bekannter Ansatz aus der Biologie gewählt und in ein Robotersystem integriert.

## 8.1 Schlussfolgerung

Ein großer Teil der Erkenntnisse über biologische Immunsysteme beruht auf Hypothesen und Theorien, welche sich in einigen Punkten widersprechen, in anderen wiederum ergänzen. Bei den dendritischen Zellen ist dies der Fall, da sich die klassische Immunologie und die Gefahrentheorie gegenseitig erweitern. Beide Modelle finden sich neben einigen weiteren Abstraktionen aus der Biologie im dendritischen Zell-Algorithmus wieder.

In der Biologie stellen die dendritischen Zellen den Übergang vom angeborenen zum erworbe-

nen Immunsystem dar, indem sie angelockt von Gefahrensignalen die Antigene der Eindringlinge sammeln und innerhalb der Lymphknoten Fresszellen darauf trainieren. Auf den ersten Blick erfüllen die dendritischen Zellobjekte des implementierten Immunsystems dieselbe Aufgabe, wobei im Detail doch einige Unterschiede zu erkennen sind.

Zunächst wird im technischen System das angeborene Immunsystem stark reduziert. So gibt es keine vordefinierten Antigene oder Fresszellen. Nur die Signale des Gewebes werden definiert, welche in der Biologie beispielsweise beim unnatürlichen Tod einer Zelle entstehen. Im Roboter werden diese durch verschiedene Sensoren erzeugt. Des Weiteren wird durch die Abstraktion die Aufgabe der Signale abgeändert. Während in der Biologie die Gefahrensignale die dendritischen Zellen vor allem anlocken sollen, um in dieser Umgebung nach Angreifern zu suchen und erst in zweiter Instanz neben einigen weiteren Faktoren zur Bewertung der gesammelten Antigene verwendet werden, erfolgt die Bewertung der Antigene im künstlichen Immunsystem ausschließlich durch die Signale. Neben den Signalen finden sich auch die Antigene, welche als Proteinketten zur Identifikation von unterschiedlichen Zellen dienen, im entwickelten Abwehrsystem in Form einer Zeichenkette wieder. Hierbei werden allerdings nicht Zellen, sondern Zustände der Verhaltenskontrolle identifiziert, was zu einer globaleren Einschätzung des Gesamtsystems führt.

Auch bei der Abbildung der dendritischen Zellen selbst auf der technischen Ebene lassen sich einige Differenzen zum biologischen Vorbild erkennen. So stellt der unreife Zellzustand zwar in beiden Bereichen eine antigensammelnde Zelle dar, doch unterscheiden sich Tätigkeiten der Zelle in den restlichen Zuständen erheblich. Aus biologischer Sicht nimmt die dendritische Zelle nach der Antigenaufnahme einen semi-reifen Zustand ein und begibt sich zu den Lymphknoten, wo sie durch die Vollendung der Reifung in die Lage versetzt wird, die Antigene zu präsentieren. Bei der technischen Nachbildung der Zellen wird hingegen nach der Antigensammlung direkt der semi-reife oder reife Zustand eingenommen, wobei der semi-reife eine harmlose und der reife eine gefährliche Zelle darstellt.

KRIS weist noch weitere Unterschiede im Detail zum biologischen Abwehrsystem auf, trotzdem lassen sich neben den Namen der Objekte weitere interessante Verbindungen zur Biologie ziehen. Bei der Einschätzung der Antigene erhält das System selbst einen größeren Einfluss, wie es auch in biologischen Organismen zu finden ist. Außerdem entstehen Antigene nicht zufällig, sondern werden vom Robotersystem generiert. Zudem fehlt bei KRIS aufgrund des permanenten Lernens eine klare Trennung zwischen Trainings- und Testphase, die in vielen bisherigen künstlichen Abwehrsystemen üblich war. Im Überblick ergeben sich einige vielversprechende

Ansätze aus den modellierten Beziehungen zwischen künstlichem und biologischem Immunsystem, durch welche diverse vorteilhafte Eigenschaften in ein technisches System eingebracht werden.

Im Rahmen der Evaluierung hat sich gezeigt, dass der dendritische Zell-Algorithmus sehr gut in der Lage ist, einen großen Teil der Fehlererkennung abzuwickeln, was auch die einleitende Forschungsfrage beantwortet. Das System kann klar zwischen Gefahrensituationen und harmlosen Zuständen unterscheiden. Selbst bei schwierigen Sachlagen, wo Antigene zwischen sicheren und gefährlichen Bedingungen wechseln, kann KRIS entsprechend reagieren. Der Ausbau von KRIS in Form einer Immunreaktion hat bewiesen, dass in weiterer Folge selbst die Fehlerbehandlung kein größeres Problem darstellt. Als besondere Bereicherung hat sich hierbei das Einbeziehen verschiedenster Datenquellen, wie Sensoren oder Kameras, zur Bewertung der einzelnen Zustände erwiesen.

Vor allem im Hinblick auf die Verwendung im Roboter und dessen vielen unterschiedlichen Subsystemen, von Motorensteuerung über Bildverarbeitung bis zur künstlichen Intelligenz profitiert KRIS stark von den Eigenschaften des dendritischen Zell-Algorithmus. In jedem einzelnen Bereich des Systems bietet sich eine enorme Menge an Datenquellen an, welche bei den unterschiedlichen Fehlerzuständen zur Erkennung beitragen kann. Im besten Fall können dadurch Ausfälle von gesamten Subsystemen durch andere Systembereiche erkannt und dementsprechend reagiert werden. Diese Feststellung beantwortet die Forschungsfrage nach der parallelen Verarbeitung mehrerer Fehlerklassen, da dies eine Voraussetzung für das Einbeziehen mehrerer Subsysteme darstellt.

In allen Versuchsdurchläufen ist die Genauigkeit der Ergebnisse beeindruckend, was direkt auf den letzten Teil der Forschungsfrage nach der Praktikabilität Bezug nimmt. Es kommt zwar in einigen Fällen zu falschen Einschätzungen, jedoch treten diese Fälle meist nur in zeitlicher Nähe zum Systemstart auf, wo KRIS aufgrund mangelnder Daten äußerst instabil ist. Außerdem können Antigene, welche selten auftreten, recht schnell falsch eingeschätzt werden. Insgesamt gibt es trotzdem in allen Versuchsreihen weniger als 1% an fälschlich positiv eingeschätzten Zuständen. Je nach Konfiguration können die ersten Aussagen über die Antigene innerhalb weniger Sekunden erbracht werden, was im Hinblick auf die Rechenleistung der NAOs und deren Fortbewegungsgeschwindigkeit akzeptabel ist.

Bei den Untersuchungen des biologischen Immunsystems und klassischer künstlicher Immunsysteme können schnell einige Grenzen der negativen und kolonialen Selektion erkannt werden.

Hierbei werden häufig die Faktoren, dass das Immunsystem ein mehrschichtiges und verteiltes System ist, welches auf eine beachtliche Menge an Daten zurückgreift, außer Acht gelassen. Dadurch wird der größte Teil der Krankheitserreger bereits vor dem Einsatz des adaptiven Immunsystems abgewehrt. Diese Faktoren sollten auch in künstlichen Immunsystemen von tragender Bedeutung sein.

In dieser Arbeit wurde daher ein künstliches Immunsystem vorgestellt, welches durch die Verwendung mehrerer Datenquellen einen besseren Überblick über die Systemsituation erhält und dadurch in die Lage versetzt wird, eine bessere Beurteilung der einzelnen Zustände durchzuführen. Neben den zahlreichen Versuchen am Simulator konnten durch die Integration von KRIS in die Software der Austrian Kangaroos erste vielversprechende Daten im realen Einsatz gewonnen werden.

Aus diesem Grund bildet diese Diplomarbeit eine gute Grundlage für weiterführende Forschung im Bereich künstlicher Immunsysteme und vor allem in der Fehlerbehandlung von Robotersystemen. Durch das vorgestellte System konnte der komplette Verlauf von der Datensammlung über die Bewertung bis hin zur Reaktion konstruiert werden. In einigen Punkten besteht allerdings Anpassungs- und Verbesserungsbedarf.

Einen naheliegenden Schritt stellt die Verteilung des künstlichen Immunsystems über den gesamten Roboter dar. Dabei sollen auf den verschiedensten Ebenen und Teilsystemen Signale erfasst werden, wodurch mehr Daten zur Bewertung der Zustände vorhanden wären. Des Weiteren wäre die Überlegung sinnvoll, einige Ansätze der Multi-Agenten-Systeme wegen ihrer besseren Lastverteilung in Betracht zu ziehen. Außerdem ist die Zusammenstellung der Antigenen zu überarbeiten, um neben den Verhaltenszuständen weitere Informationen einbeziehen zu können.

Als sehr interessanten Anwendungsbereich für Immunsysteme im Roboterfußball hat sich im Laufe der Arbeit die Lokalisierung hervorgehoben. Durch den Einfluss verschiedenster Faktoren wird zurzeit auf einem vollkommen symmetrischen Feld versucht, das eigene vom gegnerischen Tor zu unterscheiden. Allerdings treffen die einzelnen Module in diesem Bereich einen Großteil der Entscheidungen eigenständig, was zu einer Vielzahl an Fehlern führt. In diesem Fall könnte das Immunsystem als zentraler Knotenpunkt zwischen den Lokalisierungsmodulen dienen, um fehlerhafte Einzelentscheidungen zu meiden.

Eine weitere Möglichkeit zur Anwendung des implementierten Systems stellt die Farbkalibrie-

rung dar, welche von vielen Teams des Robocup bereits automatisiert wurde. Diese hängt neben der einzelnen Farben von einer Vielzahl von Kameraeinstellungen ab, wodurch sie einen erheblichen Aufwand in der Spielvorbereitung darstellt. Durch Verfahren aus dem Bereich des maschinellen Lernens sowie aufgrund der Überprüfung durch das künstliche Immunsystem könnte eine Automatisierung dieses Vorganges erreicht werden, wodurch das Robocupteam eine enorme Entlastung erfahren würde.

Die Reaktion des Systems bei Auftreten eines Fehlerfalls wurde im vorgestellten System auf eine einfache Art und Weise implementiert. Besonders die Fragen: “Wann soll eine Immunreaktion erfolgen?” und “Wie wird die Reaktionsform ausgewählt?” stellen ein breites Spektrum an Weiterentwicklungsmöglichkeiten dar. Wobei Fähigkeiten wie Systemanalyse und Selbstreparatur eine tragende Rolle spielen können.

Das in dieser Arbeit vorgestellte System bildet die Grundlage für weitere wissenschaftliche Untersuchungen im Bereich künstlicher Immunsysteme in cyberphysikalischen Systemen. Die vielversprechenden Ergebnisse von KRIS und dessen enormes Potenzial lassen mittelfristig einen praktischen Einsatz in diversen Roboteranwendungen erwarten.



# Glossar

**Antigen** ist meist eine Proteinkette, welche einen Eindringling identifiziert und die Bildung von Antikörpern induziert.

**Antikörper** sind Proteine, welche zur Abwehr von Fremdstoffen dienen. Nach dem Erkennen neuer Fremdstoffe werden Antikörper von B-Zellen produziert und gelangen in das Blut.

**B-Zellen** oder B-Lymphozyten stellen das Fundament des spezifischen humoralen Immunsystems dar. Als Folge eines Antigenreizes sind sie in der Lage, Antikörper zu bilden und dadurch eine adaptive Immunreaktion einzuleiten.

**Blackboard** ist ein Architekturmuster in der Softwareentwicklung, welches bei Problemen ohne eindeutige Lösungsstrategie zur Anwendung kommt. Die Zerlegung des Problems und das daraus resultierende Zusammenspiel einzelner Komponenten innerhalb des Blackboards führt zur näherungsweise Lösung des Problems.

**Computational Intelligence** beschreibt ein Teilgebiet der Künstlichen Intelligenz, welches die algorithmische Umsetzung biologisch inspirierter Modelle umfasst.

**Dendritische Zellen** sind antigenpräsentierende Immunzellen, welche als Verbindungsglied zwischen angeborenem und adaptivem Immunsystem agieren.

**Fehlerklasse** beschreibt die Zusammenführung von Fehlern aufgrund ihrer überlappenden Eigenschaften.

**Framework** bezeichnet ein Programmiergerüst innerhalb dessen ein Entwickler seine Anwendung erstellt, wobei die Struktur des Programmes durch die vorhandenen Muster beeinflusst wird.

**Fresszellen** sind Zellen des Immunsystems, welche zur Vernichtung von Krankheitserregern, Fremdkörpern und Zelltrümmern dienen.

**Immunantwort** bezeichnet die Reaktion des Immunsystems auf das Eindringen von Krankheitserregern.

**Immunzellen** fasst alle zellulären Bestandteile des Immunsystems zusammen.

**Lymphknoten** befinden sich an unterschiedlichsten Stellen des Körpers. Die Knoten beschäftigen sich neben der Filterung der Lymphe mit der Erzeugung und Reifung von Immunzellen und bilden daher einen wichtigen Teil des Abwehrsystems.

**Mutation** ist eine Veränderung des Gencodes eines Individuums. Zufällig werden dabei einzelne Gene verändert, was positive aber auch negative Auswirkungen haben kann.

**Pathogen** bedeutet "Krankheit verursachend". Außerdem werden als Pathogene alle Angreifer zusammengefasst, die eine Erkrankung ursächlich bedingen können.

**Rezeptor** bezeichnet Proteinkomplexe an der Hülle einer Zelle, welcher in der Lage sind, unterschiedliche Teilchen zu binden und dadurch Prozesse im Zellinneren auszulösen.

**Robocup** ist eine internationale wissenschaftliche Initiative, welche mit Hilfe von Wettbewerben die Weiterentwicklung intelligenter Roboter fördert.

**T-Zellen** auch T-Lymphozyten haben durch ihre spezifischen Rezeptoren die Möglichkeit, Angreifer zu erkennen, zu binden und zu vernichten.

**Zytokine** beschreiben Proteine, welche der Regulierung der Immunantwort dienen.

# Literaturverzeichnis

- [1] AICKELIN, Uwe ; BENTLEY, Peter ; CAYZER, Steve ; KIM, Jungwon ; MCLEOD, Julie: Danger Theory: The Link between AIS and IDS? In: TIMMIS, Jon (Hrsg.) ; BENTLEY, Peter J. (Hrsg.) ; HART, Emma (Hrsg.): *ICARIS* Bd. 2787, Springer, 2003 (Lecture Notes in Computer Science)
- [2] AL-HAMMADI, Yousof ; AICKELIN, Uwe ; GREENSMITH, Julie: DCA for Bot Detection. In: *Computing Research Repository* abs/1001.2195 (2010)
- [3] ALDEBARAN, Robotics: *Aldebaran robotics website*. November 2011. – <http://www.aldebaran-robotics.com> zuletzt besucht am 02.03.2012
- [4] BI, Ran ; TIMMIS, Jon ; TYRRELL, Andy M.: The Diagnostic Dendritic Cell Algorithm for robotic systems. In: *IEEE Congress on Evolutionary Computation*, IEEE, 2010, S. 1–8
- [5] BURNET, Frank M.: *The clonal selection theory of acquired immunity*. Nashville, Vanderbilt University Press,, 1959. – 232 S. – <http://www.biodiversitylibrary.org/item/34425> zuletzt besucht am 28.08.2012
- [6] BURNET, Frank M.: Clonal selection and after. In: *Theoretical Immunology* (1978), S. 63–85
- [7] CASTRO, Leandro N. ; VON ZUBEN, Fernando J.: The Clonal Selection Algorithm with Engineering Applications. In: *GECCO'00, Workshop on Artificial Immune Systems and Their Applications*, 2000, S. 36–37
- [8] COHEN, Irun R.: *Tending Adam's Garden : Evolving the Cognitive Immune Self*. Academic Press, 2004. – ISBN 0121783561

- [9] DE BOER, Rob J. ; PERELSON, Alan ; KEVEREKIDID, Ioannis G.: Immune Network Behaviour - I. From Stationary States to Limit Cycle Oscillations. In: *Bulletin of Mathematical Biology* 55 (1993), S. 745–780
- [10] FARMER, Doyne J. ; PACKARD, Norman H. ; PERELSON, Alan S.: The immune system, adaptation, and machine learning. In: *Physica 22D* 2 (1986), Nr. 1-3, S. 187–204
- [11] FORREST, Stephanie ; PERELSON, Alan S. ; ALLEN, Lawrence ; CHERUKURI, Rajesh: Self-Nonself Discrimination in a Computer. IEEE Computer Society Press, 1994. – Forschungsbericht. – 202–212 S.
- [12] GREENSMITH, Julie: *The Dendritic Cell Algorithm*, School of Computer Science, University of Nottingham, Diss., 2007
- [13] GREENSMITH, Julie ; AICKELIN, Uwe: The Deterministic Dendritic Cell Algorithm. In: *Proceedings of the 7th international conference on Artificial Immune Systems*. Berlin, Heidelberg : Springer-Verlag, 2008 (ICARIS '08). – ISBN 978–3–540–85071–7, 291–302
- [14] GREENSMITH, Julie ; AICKELIN, Uwe ; CAYZER, Steve: Detecting Danger: The Dendritic Cell Algorithm. In: *Robust Intelligent Systems* (2008), S. 89–112
- [15] GREENSMITH, Julie ; TWYXCROSS, Jamie ; AICKELIN, Uwe: Dendritic Cells for Anomaly Detection. In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, 2006*, S. 664–671
- [16] GREENSMITH, Julie ; WHITBROOK, Amanda M. ; AICKELIN, Uwe: Artificial Immune Systems (2010). In: *Handbook of Metaheuristics*. 2010, S. 421–448
- [17] HANG, Xiaoshu ; DAI, Honghua: An Immune Network Approach for Web Document Clustering. In: *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA : IEEE Computer Society, 2004 (WI '04), S. 278–284
- [18] HART, Emma ; TIMMIS, Jon: Application areas of AIS: The past, the present and the future. In: *Appl. Soft Comput.* 8 (2008), January, S. 191–201
- [19] HOFMEYR, Steven A. ; FORREST, Stephanie: Architecture for an artificial immune system. In: *Evolutionary Computation* 8 (2000), Nr. 4, S. 443–473
- [20] HUNT, John ; TIMMIS, Jon ; COOKE, Denise ; NEAL, Mark ; KING, Clive: Jisys: Development of an Artificial Immune System for real world applications. In: *Artificial Immune Systems and their Applications*, Springer-Verlag, 1998, S. 157–186

- [21] JERNE, Niels K.: Towards a network theory of the immune system. In: *Annales d'immunologie* 125C (1974), Januar, Nr. 1-2, S. 373–389
- [22] JI, Zhou ; DASGUPTA, Dipankar: Revisiting negative selection algorithms. In: *Evolutionary Computation* 15 (2007), Nr. 2, S. 223–251
- [23] KRAUTMACHER, Michael ; DILGER, Werner: AIS Based Robot Navigation in a Rescue Scenario. In: NICOSIA, Giuseppe (Hrsg.) ; CUTELLO, Vincenzo (Hrsg.) ; BENTLEY, Peter J. (Hrsg.) ; TIMMIS, Jon (Hrsg.): *ICARIS* Bd. 3239, Springer, 2004 (Lecture Notes in Computer Science), S. 106–118
- [24] LÖTZSCH, Martin ; BACH, Joscha ; BURKHARD, Hans-Dieter ; JÜNGEL, Matthias: Designing Agent Behavior with the Extensible Agent Behavior Specification Language XABSL. In: POLANI, Daniel (Hrsg.) ; BROWNING, Brett (Hrsg.) ; BONARINI, Andrea (Hrsg.): *RoboCup 2003: Robot Soccer World Cup VII* Bd. 3020. Padova, Italy : Springer, 2004 (Lecture Notes in Artificial Intelligence), S. 114–124
- [25] MATZINGER, Polly: Tolerance, danger, and the extended family. In: *Annual review of immunology* 12 (1994), Nr. 1, S. 991–1045. – ISSN 0732–0582
- [26] MOKHTAR, Maizura ; BI, Ran ; TIMMIS, Jon ; TYRRELL, Andy M.: A modified dendritic cell algorithm for on-line error detection in robotic systems. In: *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*. Piscataway, NJ, USA : IEEE Press, 2009 (CEC'09), S. 2055–2062
- [27] OATES, Robert ; GREENSMITH, Julie ; AICKELIN, Uwe ; GARIBALDI, Jonathan ; KENDALL, Graham: The application of a dendritic cell algorithm to a robotic classifier. In: *Proceedings of the 6th international conference on Artificial immune systems*. Berlin, Heidelberg : Springer-Verlag, 2007 (ICARIS'07). – ISBN 3–540–73921–1, 978–3–540–73921–0, 204–215
- [28] OATES, Robert ; MILFORD, Michael ; WYETH, Gordon ; KENDALL, Graham ; GARIBALDI, Jonathan M.: The implementation of a novel, bio-inspired, robotic security system. In: *Proceedings of the 2009 IEEE international conference on Robotics and Automation*. Piscataway, NJ, USA : IEEE Press, 2009 (ICRA'09). – ISBN 978–1–4244–2788–8, 1337–1342
- [29] RÖFER, Thomas ; LAUE, Tim ; MÜLLER, Judith ; FABISCH, Alexander ; FELDPAUSCH, Fynn ; GILLMANN, Katharina ; GRAF, Colin ; HAAS, Thijs J. ; HÄRTL, Alexander ; HU-

MANN, Arne ; HONSEL, Daniel ; KASTNER, Philipp ; KASTNER, Tobias ; KÖNEMANN, Carsten ; MARKOWSKY, Benjamin ; RIEMANN, Ole Jan L. ; WENK, Felix: *B-Human Team Report and Code Release 2011*. 2011. – [http://www.b-human.de/downloads/bhuman11\\_coderelease.pdf](http://www.b-human.de/downloads/bhuman11_coderelease.pdf) zuletzt besucht am 03.02.2012

- [30] SCHLENDER, Hannes ; SCHILLING, Jo: Infektionsforschung - Immunsystem erforschen, Erreger bekämpfen, Menschen schützen / Bundesministerium für Bildung und Forschung (BMBF). Bonn, Berlin, 2009. – Forschungsbericht
- [31] SCHNURR, Maximilian ; GALAMBOS, Peter ; SCHOLZ, Christoph ; DAUER, Marc ; KRUG, Anne ; HARTMANN, Gunther ; EIGLER, Andreas ; ENDRES, Stefan: Dendritische Zellen - Träger tumorgerichteter Immuntherapie. In: *Deutsches Ärzteblatt* 99 (2002), Nr. 37, S. 2408–2416
- [32] SUZUKI, Junichi ; YAMAMOTO, Yoshikazu: Building an Artificial Immune Network for Decentralized Policy Negotiation in a Communication Endsystem: OpenWebServer/iNexus Study. In: *Proc. of the 4 th World Conference on Systemics, Cybernetics and Informatics*, 2000
- [33] TIMMIS, Jon: Artificial immune systems — today and tomorrow. In: *Natural Computing* 6 (2007), März, Nr. 1, S. 1–18
- [34] TWYXCROSS, Jamie ; AICKELIN, Uwe: libtissue - implementing innate immunity. In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, 2006*, S. 499–506