

Universität Paderborn/ C - LAB

WS 2002/ 2003

Projektgruppe Paderkicker2

Leitung: Dr. rer nat. Bernd Kleinjohann, Dirk Stichling

Multiagentensysteme

Tobias Bräutigam

9. Semester Diplominformatik

tobiasb@upb.de

INHALTSVERZEICHNIS

1. Einführung
 - 1.1. Definitionen
 - 1.2. Interaktion
 - 1.3. Ziel

2. Kommunikation
 - 2.1. Voraussetzungen
 - 2.2. Sprechakte
 - 2.3. Weitere wichtige Aspekte

3. Planen
 - 3.1. Besonderheiten im Roboterfußball
 - 3.2. Bestandteile der BDI-Architektur
 - 3.3. Der BDI-Entscheidungsprozess

4. Lernen
 - 4.1. Die Verhaltensregeln der Teams
 - 4.2. Die verschiedenen Reinforcement Funktionen
 - 4.2.1. Die Globale Reinforcement Funktion
 - 4.2.2. Die Lokale Reinforcement Funktion
 - 4.2.3. Die R-touch Funktion
 - 4.3. Die Versuchsserie
 - 4.4. Erkenntnisse

5. Zusammenfassung

1. EINFÜHRUNG

Das Gebiet der Multiagentensysteme ist ein sehr umfangreiches Gebiet der Forschung, daher sollen hier nur die Aspekte berücksichtigt werden, welche für den Roboterfußball relevant sind. Zunächst wird durch einige Definitionen verdeutlicht was die wesentlichen Merkmale eines Multiagentensystems sind um dann einige Gründe anzugeben warum es von Vorteil ist mehrere Agenten miteinander arbeiten zu lassen.

Im folgenden Abschnitt sollen die Probleme dargestellt werden, die man beachten muss wenn man mehrere Agenten miteinander kommunizieren lassen will. Aufgabe dieses Abschnitts ist es lediglich Problembewusstsein zu schaffen, da der Bereich der Kommunikation sehr umfangreich ist und sich hauptsächlich auf den Teilbereich der natürlich sprachlichen Kommunikation konzentriert, was vor dem Hintergrund des Roboterfußballs nicht weiter relevant ist.

Im nächsten Abschnitt wird eine Architektur vorgestellt, die sich aufgrund ihrer speziellen Eigenschaften sehr gut für die Planung in Multiagentensystemen verwenden lässt.

Abschließend werden einige Erkenntnisse vorgestellt, die sich aus empirischen Untersuchungen von lernenden Roboterfußballteams ergeben haben. Diese Erkenntnisse lassen Rückschlüsse auf vorteilhafte Lernstrategien in Roboterteams zu.

1.1. DEFINITIONEN

Zunächst mal soll hier die grundlegende Frage geklärt werden: „Was sind überhaupt Multiagentensysteme?“ Vereinfacht gesagt sind dies Systeme interagierender Agenten.

Agenten werden in [RN95] folgendermaßen definiert:

„An Agent is anything, that can be viewed as perceiving its environment through sensors and action upon that environment through effectors”

Das heißt ein Agent nimmt seine Umwelt wahr und basierend auf diesen Informationen.

Abschließend soll auch der Begriff Interaktion eindeutig definiert werden, in [SF86] findet man dazu folgende Begriffsbestimmung:

„Interaktion ist die Wechselseitige Beeinflussung des Handelns mindestens zweier Personen“

1.2. INTERAKTION

Interaktion kann man wiederum in implizite und explizite Interaktion unterteilen. Dabei spricht man von einer impliziten Interaktion wenn das Verhalten von anderen durch die Veränderung der Umgebung beeinflusst wird. Dies geschieht meist unbeabsichtigt, da der agierende Agent in der Regel seine Aktion nicht ausführt um eine bestimmte Reaktion von Anderen zu provozieren. Wenn ein Agent zum Beispiel einen Ball in Richtung Tor schießt, wird er es mit der Absicht tun ein Tor zu erzielen und nicht damit gegnerische Roboter ihre Position ändern, obwohl ein ähnliches Szenario durchaus denkbar wäre, worauf hier aber nicht weiter eingegangen werden soll.

Nilsson [NIL98] bezeichnet dies als implizite Kommunikation, diese Begriffsbestimmung wurde aber bewusst nicht übernommen, da in dem oben

beschriebenen Fall eine Form von Interaktion stattfindet jedoch keine Kommunikation. Um bei dem bereits benutzten Beispiel zu bleiben, wenn ein Agent aufs Tor schießt und somit die Gegner dazu bringt ihre Position zu ändern, findet kein Akt der Kommunikation zwischen diesen Agenten statt, allerdings kann man sehr wohl von Interaktion sprechen.

Explizite Interaktion könnte man hingegen mit Kommunikation gleichsetzen. Hier sprechen sich die Agenten ab um ihr Verhalten dann gemäß Absprache zu verändern. Bei dieser Art der Interaktion verursacht ein Agent eine gewollte Verhaltensänderung bei einem oder mehreren anderen Agenten, was ohne Kommunikation nicht zuverlässig möglich ist. Im Folgenden ist wenn von Interaktion die Rede ist explizite Interaktion gemeint.

Nun stellt sich die Frage warum sollen Agenten überhaupt interagieren. Hierfür spricht eine Reihe von Aspekten.

Jeder Agent hat nur ein unvollständiges Weltmodell, er kann immer nur einen Teil seiner Umgebung wahrnehmen, hat also eine eingeschränkte Weltsicht.

Die durch die Sensoren gesammelten Daten werden lokal bei dem Agenten gespeichert, sind also dezentralisiert.

Die Systemkontrolle ist verteilt, denn es gibt in der Regel keine übergeordnete Instanz, welche die Agenten steuert. In der hier betrachteten Domäne des Roboterfußballs ist dies sogar explizit untersagt worden.

Zudem haben Agenten nur beschränkte Fähigkeiten, ein Team kann also durch Spezialisierung der einzelnen Mitglieder einen wesentlich größeren Aufgabenbereich abdecken und gestellte Aufgaben besser lösen, als Vorbild dient hier wie so oft die Natur, wo es zum Beispiel in Ameisenkolonien oder Bienenstöcken Spezialisten für die verschiedenen Aufgaben gibt wie Arbeiter, die Königin, usw..

All diese Nachteile lassen sich durch Interaktion entweder ganz oder zumindest teilweise beheben.

1.3. ZIEL

Das Ziel in einem Multiagentensystem soll sein, das die einzelnen Agenten parallel an der Bearbeitung einer Aufgabe arbeiten, indem sie diese in Teilaufgaben zerlegen und sie so effizienter lösen können.

Die Vorteile die sich hieraus ergeben liegen auf der Hand, zum einen ist es schwerer nur einen „allmächtigen“ Agenten zu entwickeln, der alle Aufgaben lösen kann, als mehrere kleine Agenten mit speziellerem Fähigkeiten die zusammengenommen auch alle Aufgaben lösen können.

Zum anderen können die Aufgaben schneller bearbeitet werden wenn sich mehrere Agenten mit der Lösung befassen.

2. KOMMUNIKATION

2.1. VORRAUSSETZUNGEN

Als Basis für die Interaktion zwischen Agenten dient die Kommunikation, damit zwei Agenten überhaupt miteinander kommunizieren können müssen bestimmte Voraussetzungen geschaffen werden. Damit sich diese Agenten überhaupt verstehen können müssen sie dieselbe Sprache sprechen. Ebenfalls müssen sie eine vergleichbare Vorstellung von ihrer Umgebung haben um die Nachrichten verstehen zu können. Wenn Agent A ein Objekt anspricht muss Agent B wissen um welches Objekt es sich handelt damit er die Nachricht richtig versteht. Was Agent A zu Agent B sagt, und wie dieser die Nachricht interpretiert, hängt also davon ab was

A und B voneinander zu wissen glauben. Nielsson [NIL98] spricht hier von Modellen von anderen Agenten.

2.2. SPRECHAKTE

Bei Russel / Norvig [RN95] und Nielsson [NIL98] werden so genannte Sprechakte eingeführt mit denen die Kommunikation klassifiziert wird. Wobei hier jegliche Art von Kommunikation gemeint ist also nicht nur gesprochene Worte sondern auch zum Beispiel Schrift oder Zeichensprache. Folgende Sprechakte werden unterschieden:

- Information
- Frage
- Antwort
- Befehl
- Bitte
- Versprechen / Angebot
- Bestätigung
-

Diese Liste könnte man ohne weiteres fortführen, jedoch sollten hier die gegebenen Beispiele reichen um sich einen Eindruck verschaffen zu können.

Der Agent steht nun vor dem Problem den richtigen Sprechakt auszuwählen, er muss also entscheiden mit welchem Sprechakt er sein Kommunikationsziel am besten erreichen kann, auch hier soll ein Beispiel aus dem Roboterfußball zur besseren Verständlichkeit dienen. Wenn ein Roboter möchte, dass sein Kommunikationspartner zu einer bestimmten Stelle fährt um dort den Ball abzufangen, hat er mehrere Möglichkeiten. Er könnte dem anderen Agenten den

Befehl erteilen dorthin zu fahren, wobei es allerdings keine Garantie gibt, dass dieser Befehl auch befolgt wird, oder er teilt dem anderen die Position des Balles mit und lässt ihn entscheiden, ob er zum Ball fährt oder nicht. An diesem Beispiel wird deutlich, dass der Agent die Auswahl des Sprechaktes nur treffen kann, wenn er ein gewisses Modell des anderen Agenten hat und somit berechnen kann, welcher Sprechakt den anderen Agenten zur gewünschten Aktion führt.

2.3. WEITERE WICHTIGE ASPEKTE

Es gibt noch einige weitere wichtige Aspekte, die bei der Kommunikation berücksichtigt werden müssen. Kommunikation ist nichtdeterministisch, da man nicht vorhersehen kann, wie die Gegenseite auf die eigenen Mitteilungen reagiert, daher ist sie nicht planbar, man kann immer nur auf die aktuellen Antworten reagieren.

Man muss auch berücksichtigen, dass eine Mitteilung falsch sein könnte, was auch unabsichtlich geschehen kann, denn ein Agent kann durchaus falsche Informationen weitergeben, die er allerdings für richtig hält. Daher sollten Informationen immer mit der eigenen Wissensbasis verglichen werden und nicht immer alles automatisch als wahr in den eigenen Wissensstand übernommen werden. Agenten dürfen also nicht naiv sein und alles „glauben“.

3. PLANEN

3.1 BESONDERHEITEN IM ROBOTERFUSSBALL

In diesem Abschnitt soll eine Architektur vorgestellt werden, welche sich hervorragend für den Bereich des Roboterfußballs einsetzen lässt. Einige der besonderen Begebenheiten, die in diesem speziellen Umfeld herrschen, werden nun kurz vorgestellt. Ein entscheidender Aspekt ist die nichtdeterministische Umgebung,

d.h. es ist nicht berechenbar was als nächstes passiert, ebenso handelt der Spieler selbst nichtdeterministisch, weil Teile seines Planungsprozesses zufällig geschehen, da er seine Planung immer wieder der aktuellen Situation anpassen muss. Daher hängt der Erfolg der eigenen Planung auch von der Umgebung und den anderen Spielern ab.

Ein weiterer Aspekt ist, dass ein Agent mehrere Ziele zur gleichen Zeit haben kann, er kann also zum Beispiel zur selben Zeit versuchen den Ball zu erreichen und dabei den Gegner zu decken.

Wie bereits erwähnt werden die Informationen lokal beim Agenten gespeichert und sind daher bei jedem Spieler unterschiedlich.

Außerdem kann ein Agent beim Planungsprozess nicht alle Möglichkeiten berücksichtigen, da der Aufwand viel zu hoch wäre, daher muss der Agent rationell sein und solche Möglichkeiten, welche nicht realisierbar sind von vorne herein unberücksichtigt lassen, in [BHWMSM99] wird hier von "bounded rationality" gesprochen.

All diese Begebenheiten sprechen für den Einsatz der BDI-Architektur im Umfeld des Robocup.

3.2. DIE BESTANDTEILE DER BDI-ARCHITEKTUR

Um die Bestandteile von BDI zu erklären müssen zunächst einige Begriffe vorgestellt werden. Wenn von einer Welt die Rede ist, dann ist das ein Snapshot von allen Informationen die ein Agent zu einem Zeitpunkt von seiner Umgebung hat. Eine Option ist eine Menge zukünftiger Welten in denen eine bestimmte Bedingung gilt, zum Beispiel alle zukünftigen Welten in denen der Agent im Ballbesitz ist.

BDI steht für *Belief-Desire-Intention*, dabei steht *Belief* für das aktuelle Weltmodell des Agenten, seine Wissensbasis. *Desires* sind die aktuell möglichen Optionen, wenn ein Agent in Ballbesitz kommen möchte, dann sind seine *desires* alle zukünftigen Welten in denen dies der Fall ist, zu diesem Zeitpunkt ist das noch unabhängig davon ob das realisierbar ist, d.h. das der Agent diese Welten tatsächlich erreichen kann oder nicht. Eine dieser Welten wird dann zur *intention* also zu dem Zustand den der Agent tatsächlich erreichen will.

Wie der Agent nun genau von *belief* über *desire* zur *intention* kommt soll nun am BDI-Entscheidungsprozess erläutert werden.

3.3 DER BDI- ENTSCHEIDUNGSPROZESS

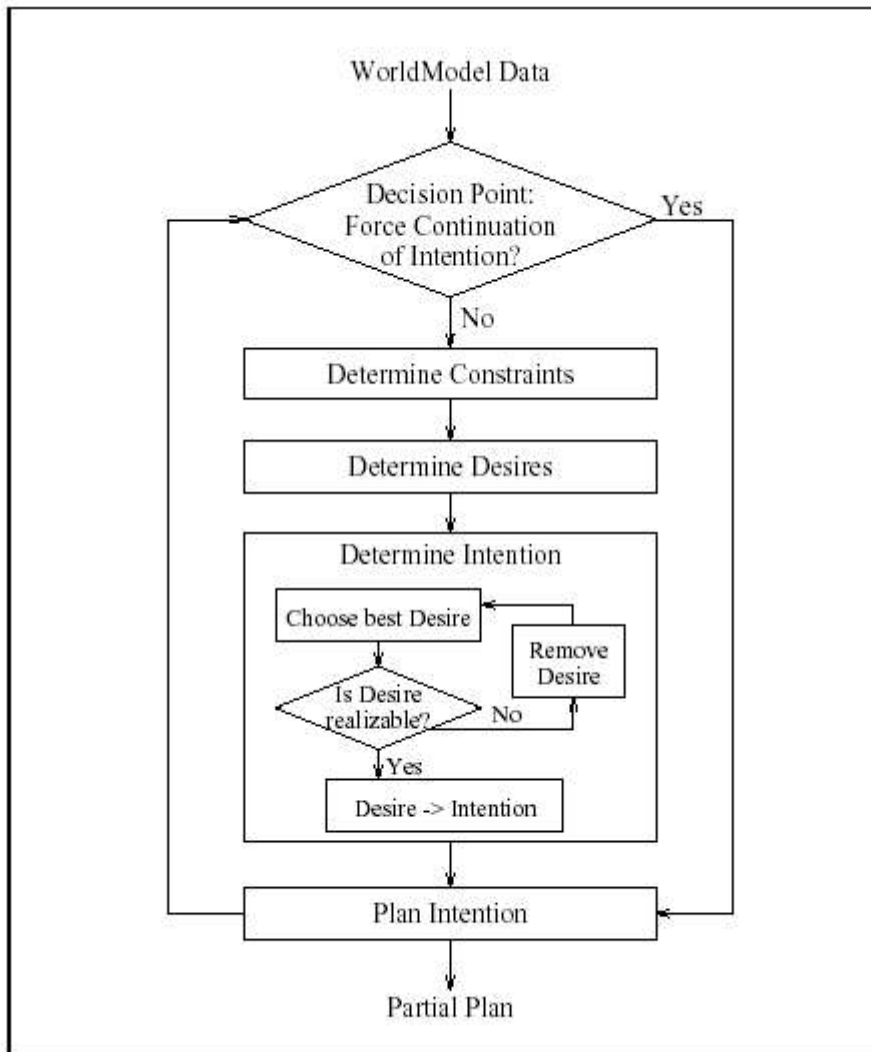


Abb. 1: Der BDI-Entscheidungsprozess

In Abbildung 1 sieht man den Ablauf eines BDI-Entscheidungsprozesses. Wenn ein Agent an einem so genannten Entscheidungspunkt angelangt ist, was in der Regel immer dann passiert wenn neue Sensordaten vorliegen, prüft er anhand der Sensordaten und dem noch aktuellen Plan ob dieser weitergeführt oder verworfen werden soll.

Wenn der Agent entschieden hat den alten Plan nicht mehr weiterzuführen, bestimmt er zunächst einmal die aktuell geltenden Beschränkungen. Danach wird eine

Auswahl zukünftiger Welten in denen vom Agenten gewünschte Bedingungen gelten bestimmt. Dies können eventuell zu viele sein, daher wird der Planungshorizont eingeschränkt, d.h. die Ziele sollten nicht zu langfristig gewählt werden.

Nun folgt der Schritt von den *desires* zur *intention*. In diesem Schritt muss der Agent entscheiden, welche der *desires* den höchsten Nützlichkeitswert hat, für diese Berechnung müssen auch die vorher bestimmten geltenden Beschränkungen berücksichtigt werden. Zu diesem Zeitpunkt hat der Agent also eine Option bestimmt, welche einen für ihn, unter den geltenden Beschränkungen, größt möglichen Nützlichkeitswert hat. Wenn diese Option realisierbar ist wird sie als *intention* ausgewählt.

Der letzte Schritt besteht nun daraus einen Plan zu entwerfen mit dem die *intention* realisiert werden kann. Dies geschieht durch die Aneinanderreihung so genannter Skills [BHWMSM99], wie zum Beispiel dribbeln, schießen oder rennen.

Zum Verdeutlichung sollen nun einige Aspekte des BDI-Entscheidungsprozesses am Beispiel *InterceptBall* gezeigt werden. Die Nützlichkeit dieser Option hängt stark von der Wahrscheinlichkeit ab, ob der Agent tatsächlich in Ballbesitz kommen kann. Wenn dies nicht möglich ist, weil zum Beispiel ein gegnerischer Spieler viel am Ball steht, ist es auch nur wenig nützlich diese Option zu realisieren. Wie man hier schon sieht muss der Agent also die anderen Agenten mit in seine Überlegungen einbeziehen, in diesem speziellen Fall also deren Entfernung zum Ball und die Geschwindigkeit und Laufrichtung des Balles, um für sich eine Entscheidung treffen kann. Bei der Berechnung der Nützlichkeit müssen noch bestimmte Nebenbedingungen berücksichtigt werden, wie zum Beispiel Abseits vermeiden, usw..

Ein wichtiger Aspekt in der BDI-Architektur ist die Beharrlichkeit mit der ein Agent seinen Plan verfolgt, in [BHW99] spricht man von „persistence of planning“. Wie wir schon bei der Vorstellung des Entscheidungsprozesses gesehen haben, startet der Agent diesen jedes Mal neu, wenn er eine neue Sensorinformation erhält. Es kann daher vorkommen, das ein Agent seine *intention* dauernd ändert und so nie eine

dieser Optionen wirklich realisiert, also sein selbst gesetztes Ziel nie erreicht, da er es nie bis zum Ende verfolgt.

Aus diesem Grund muss eine gewisse Stabilität der *intention* gewährleistet sein. Hierbei zeigen sich die folgenden zwei Extreme, zum einen wäre da ein sturer Agent, der seine Ziele nur selten ändert, was zweifelsohne in Umgebungen von Vorteil ist in denen nur wenige Änderungen vorkommen. Zum anderen wäre da ein unsicherer Agent, der seine Ziele sehr häufig ändert und so in sehr dynamischen Umgebungen besser zurecht kommen sollte.

Diese Beharrlichkeit soll nun wieder am Beispiel *InterceptBall* verdeutlicht werden. Wenn der Agent sich nun entschieden hat den Ball abzufangen kann es vorkommen, das während er zum Ball fährt, diesen aus den Augen verliert. Wenn er nun seinen alten Plan verwerfen würde und sich erst wieder zum Ball ausrichten würde, würde er wertvolle Zeit verlieren. Um dies zu vermeiden wird in [BHW99] ein so genanntes „don't care“ Intervall eingeführt, was dem Zeitraum entspricht in dem der schnellste Agent am Ball sein kann. In dieser Zeit kann sich an der Ausgangslage nicht viel ändern, daher werden währenddessen keine Aktionen ausgeführt, die zur Informationsgewinnung und nicht unmittelbar zum Plan gehören, also würde er sich in dieser Zeit nicht zum Ball ausrichten. Wenn aber dennoch neue Informationen gewonnen werden, zum Beispiel wenn der Ball von alleine wieder ins Blickfeld des Agenten rollt, werden diese Informationen dennoch berücksichtigt.

4. LERNEN

4.1. DIE VERHALTENSREGELN DER TEAMS

In [BAL98] wurde eine experimentelle Untersuchung durchgeführt, in der ein lernendes *experimental team* gegen ein *control team* mit festen Verhaltensmustern gespielt hat. Das control team besteht jeweils aus drei Stürmern und einem Torwart, die sich durch folgende Verhaltensregeln definieren.

Control Team Forward			
perceptual feature	assemblage		
	<i>mtb</i>	<i>gbb</i>	<i>mtb f</i>
<i>not behind_ball</i>	0	1	0
<i>behind_ball</i>	1	0	0

Control Team Goalie			
perceptual feature	assemblage		
	<i>mtb</i>	<i>gbb</i>	<i>mtb f</i>
<i>not behind_ball</i>	0	1	0
<i>behind_ball</i>	0	0	1

Abb. 2 Verhaltensregeln der Spieler im *control team*

In den Verhaltensregeln gibt es für die Spieler nur zwei Zustände, nämlich *behind_ball* und *not_behind_ball*. Der Torwart unterscheidet sich in seinem Verhalten nur, wenn im Zustand *behind_ball*, dann führt er nämlich die Aktion *move_to_backfield* (*mtbf*) aus, d.h. er bewegt sich zum hinteren Drittel des Spielfeldes. Die Stürmer hingegen bewegen sich in diesem Zustand auf den Ball zu (*mtb* = *move_to_ball*). Wenn sich die Spieler im Zustand *not_behind_ball* befinden führen sie, egal ob Stürmer oder Torwart, dieselbe Aktion aus, sie versuchen nämlich hinter den Ball zu kommen (*gbb* = *get_behind_ball*).

4.2. DIE VERSCHIEDENEN REINFORCEMENT FUNKTIONEN

Die Untersuchung bestand nun daraus, das Verhalten des lernenden Teams bei unterschiedlichen Reinforcement Funktionen zu beobachten. Die verschiedenen Funktionen sollen nun kurz vorgestellt werden.

4.2.1. Globale Reinforcement Funktion:

$$R_{global}(t) = \begin{cases} 1 & \text{eigenes Team erzielt Tor zum Zeitpunkt } t-1 \\ -1 & \text{Gegner erzielt Tor zum Zeitpunkt } t-1 \\ 0 & \text{sonst} \end{cases}$$

In diesem Fall werden alle Spieler gleichzeitig belohnt bzw. bestraft. Die Aktion eines einzelnen wirkt sich also auf alle aus. Der Torwart wird belohnt wenn der Stürmer ein Tor schießt und genauso wird auch der Stürmer bestraft wenn der Torwart nicht hält.

4.2.2. Lokale Reinforcement Funktion:

$$R_{local}(t) = \begin{cases} 1 & \text{wenn der Agent beim Tor in } t-1 \text{ die geringste Entfernung vom Ball hat} \\ -1 & \text{wenn der Agent beim gegn. Tor in } t-1 \text{ die ger. Entfernung vom Ball hat} \\ 0 & \text{sonst} \end{cases}$$

In diesem Fall zählt sich ein defensives Spiel, wie zum Beispiel das Spiel eines Torwartes nicht aus, da man niemals belohnt und nur bestraft werden kann. Ein weiteres Problem bei dieser Funktion ist, das der Agent der beim Tor in kürzester Entfernung zum Ball stand nicht unbedingt der Torschütze sein muss.

4.2.3. R-touch Funktion:

$$R_{touch}(t) = \begin{cases} \gamma^{t_{touch}} & \text{eigenes Team erzielt Tor zum Zeitpunkt } t-1 \\ -\gamma^{t_{touch}} & \text{Gegner erzielt Tor zum Zeitpunkt } t-1 \\ 0 & \text{sonst} \end{cases}$$

t_{touch} ist die Zeit in Millisekunden, seitdem der Agent den Ball zuletzt berührt hat. Somit ist gewährleistet, das der Agent, der das Tor erzielt auch die höchste Belohnung erhält und ebenfalls, das der Agent der den Ball zuletzt berührt hat vor einem gegnerische Tor die höchste Bestrafung erhält. γ ist ein Parameter (aus $[0,1]$), der bestimmt, wie schnell der wert der Belohnung bzw. Bestrafung verfällt nachdem

der Ball berührt wurde. Wie man sehen kann gilt $R_{touch}(t) = R_{global}(t) \times \gamma^{t_{touch}}$. Wenn also $\gamma = 1$ ist verhält sich die R-touch Funktion gleich der R-global Funktion.

4.3. DIE VERSUCHSSERIE

Eine Versuchsserie besteht aus 100 Spielen, die beendet sind sobald 10 Tore gefallen sind. Ein Experiment besteht aus 10 Serien. Die Ergebnisse der Experimente werden bewertet nach dem Torverhältnis, auch Task-Performance genannt, was zeigt wie gut die gestellte Aufgabe bewältigt wurde. Ebenfalls beobachtet wurde die Homo- bzw. Heterogenität der Agenten in dem lernenden Team, d.h. inwiefern sich unterschiedliche Verhaltensweisen herausgebildet haben innerhalb des Teams. Diese Ergebnisse wurden anhand der Unterschiede in den Verhaltensregeln herausgefunden. Als letztes wichtiges Ergebnis wurde das Lernverhalten der Agenten unter den verschiedenen Reinforcement Funktionen beobachtet, rückschlüsse auf diesen Aspekt lassen die Veränderungen in den Verhaltensregeln pro Spiel zu, man spricht hier auch von Konvergenz der Policy [BAL98].

4.4. ERKENNTNISSE

Bei der globalen Reinforcement Funktion hat sich herausgestellt, das das lernende Team im Durchschnitt 6:4 gewonnen hat. Weiterhin hat sich gezeigt, das die einzelnen Spieler des lernenden Teams heterogene Verhaltensweisen an den Tag legen, in der Regel haben sich ein Stürmer, ein Mittelfeldspieler und ein Torwart gebildet wobei der vierte Spieler zwischen dem Verhalten eines Torwartes und eines Mittelfeldspielers gewechselt hat. Die Mittelfeldspieler haben immer die Aktion *get_behind_ball* ausgeführt unabhängig von der Situation und das Verhalten der Stürmer, bzw. des Torwartes entsprach dem des Stürmers bzw. Torwartes im *control team*. Diese Tatsache lässt sich auch sehr gut in Abbildung 3 beobachten,

man sieht das im blauen Team ein Spieler zum Ball fährt während die anderen sich anders verhalten.

Allerdings zeigte sich hier kein gutes Lernverhalten, also keine Konvergenz der Policy. Nach hundert Spielen traten noch immer 0.25 Änderungen der Verhaltensregeln pro Spiel auf.



Abb. 3: Blaues Team = lernendes Team, heterogenes Verhalten

Bei der lokalen Reinforcement Funktion sehen die Ergebnisse ganz anders aus. Hier hat das lernende Team im Durchschnitt 4:6 verloren. Es hat sich auch ein homogenes Teamverhalten herausgebildet, nämlich haben alle Spieler das Verhalten eines Stürmers angenommen. Diese Tatsache ergibt sich daraus das bei der lokalen Reinforcement Funktion nur der Stürmer die Möglichkeit hat belohnt zu werden, daher ist diese Entwicklung nicht weiter verwunderlich. Wie man in Abbildung 4 sehr gut sehen kann bewegen sich alle Agenten zum Ball, zeigen also alle ein homogenes Verhalten.

Allerdings legen die Agenten ein gutes Lernverhalten an den Tag, denn es gab nach 100 Spielen nur noch 0.05 Veränderungen der Policy pro Spiel. Hier kann man durchaus von einer Konvergenz der Policy sprechen.



Abb. 4: Blaues Team = lernendes Team, homogenes Verhalten

Bei der R-touch Funktion wurden die Experimente mit steigendem γ von 0.1 bis 1.0 durchgeführt. Hier hat das lernende Team im Durchschnitt immer gewonnen, die Task Performance war also gut. Es hat sich gezeigt das eben diese Task Performance mit steigendem γ immer besser wurde, sie also höher gewonnen haben. Wie man sich nun denken kann wurden mir $\gamma = 1$ die besten Ergebnisse erzielt und wie schon gesagt wurde gilt $R_{touch}(t) = R_{global}(t)$ für $\gamma = 1$ ist. Daher sieht man das sich eine globale Reinforcement Funktion positiv auf die Task Performance auswirkt.

Ebenfalls entwickelte sich das lernende Team wie schon bei der globalen Reinforcement Funktion heterogen und die Agenten zeigten ein gutes Lernverhalten. Wie man in Abbildung 5 sehen kann nehmen die Änderungen der Policy mit steigender Anzahl der Versuche ab.

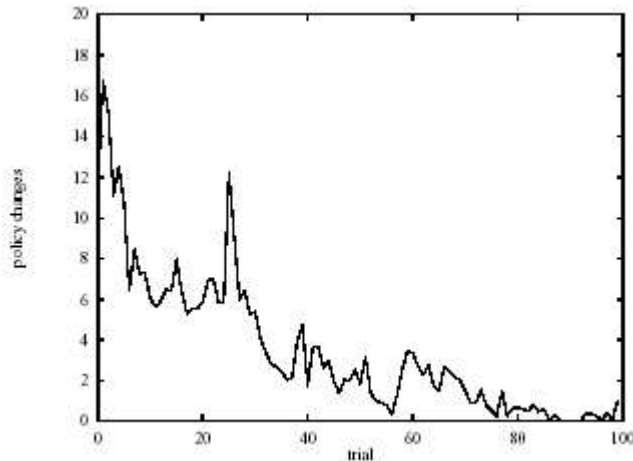


Abb. 5: Policy Änderungen gegen Anzahl der Versuche mit R-touch Funktion ($\gamma = 1$)

5. ZUSAMMENFASSUNG

Es zeigt sich also das Teamverhalten im Roboterfußball durchaus vorteilhaft ist und das vor allem heterogene Teams ein besseres Ergebnis liefern als homogene Teams. Um überhaupt ein Teamverhalten erreichen zu können müssen die Agenten miteinander kommunizieren können. Für die speziellen Eigenschaften des Roboterfußballs eignet sich die BDI-Architektur hervorragend.

Um nun heterogene Teams zu erhalten sollte man eine globale Reinforcement Funktion einsetzen. In [BAL98] wird ebenfalls gesagt das man Heterogenität, durchaus nicht von vorneherein vorsehen sollte sondern diese sich erst durch das Lernen bilden lassen sollte, also Heterogenität als Ergebnis des Lernens.

LITERATURVERZEICHNIS

[RN95] Russel / Norvig: Artificial Intelligence: A modern approach, Kap. 22, Prentice Hall

[NIL98] Nils Nilsson: Artificial Intelligence, Kap. 23/24, Morgan Kaufmann Publishers

[BAL98] T. Balch, Behavioral Diversity in Learning Robot Teams, 1998

[HUB99] Marcus J. Huber: JAM: A BDI-theoretic Mobile Agent Architecture, Proceedings of the third annual conference on Autonomous Agents 1999, Pages: 236 – 243

[BHW99] H. Burkhard, M. Hannebauer, J. Wendler: Belief-Desire-Intention Deliberation in Artificial Soccer, In Proceedings of the IJCAI-99 Third International Workshop on RoboCup, Stockholm, Schweden, Aug. 1999

[BHWMSM99] H. Burkhard, M. Hannebauer, J. Wendler, H. Myritz, G. Sander, T. Meinert: BDI Design Principles and Cooperative Implementation – A Report on Robocup Agents, In Proceedings of the IJCAI-99 Third International Workshop on RoboCup. Stockholm, Schweden, 1999.

[SF86] Sarges, Fricke: Psychologie für die Erwachsenenbildung / Weiterbildung, 1986